



HPM5E00

32-bit high-performance microcontroller based on RISC-V core Rev 0.3

HPM5E00 Series High-Performance Microcontrollers User Manual

Suitable for the following models of products of Shanghai Xiangji Semiconductor Technology Co., Ltd.:

HPM5E31IGN1、HPM5E31IPB1、HPM5E31IVK1

HPM5E11IGN1、HPM5E11IPB1、HPM5E11IVK1

HPM5E3YIGN1、HPM5E3YIVK1

Contents

1	Product Overview	1
1.1	System Diagram	1
1.2	Feature Summary	3
1.2.1	Kernel and System	3
1.2.2	Internal memory	4
1.2.3	Power Management	4
1.2.4	Clock	5
1.2.5	Reset	5
1.2.6	Boot Up	5
1.2.7	External memory	6
1.2.8	Algorithm hardware acceleration	6
1.2.9	Electric Control System	6
1.2.10	Timer	6
1.2.11	Real-time Ethernet System	6
1.2.12	Communication Peripherals	7
1.2.13	Analog Peripherals	7
1.2.14	Input and Output	7
1.2.15	Information Security System	8
1.2.16	System debugging	8
1.3	document convention	8
1.3.1	register	8
2	Processor Core	10
2.1	CPU	10
2.2	Dual Core Configuration	10
2.3	Bus and Memory Interface	10
2.4	TRAP	11
2.5	Machine Timer MCHTMR	11
2.6	Hardware Performance Monitor	11
2.7	privileged mode	11
2.8	Physical Memory Attributes	12
2.9	Physical Memory Protection	13
2.10	related documents	13
3	Control status registers for RISC-V processors	14
3.1	Control Status Register CSR Description	14
3.2	Control Status Register CSR Details	20
3.2.1	USTATUS (0x0) (standard read/write)	20
3.2.2	UIE (0x4) (standard read/write)	20
3.2.3	UTVEC (0x5) (standard read/write)	21
3.2.4	USCRATCH (0x40) (standard read/write)	21
3.2.5	UEPC (0x41) (standard read/write)	22
3.2.6	UCAUSE (0x42) (standard read/write)	22

3.2.7	UTVAL (0x43) (standard read/write)	23
3.2.8	UIP (0x44) (standard read/write)	23
3.2.9	SSTATUS (0x100) (standard read/write)	24
3.2.10	SEDELEG (0x102) (standard read/write)	26
3.2.11	SIDELEG (0x103) (standard read/write)	28
3.2.12	SIE (0x104) (standard read/write)	29
3.2.13	STVEC (0x105) (standard read/write)	29
3.2.14	SCOUNTEREN (0x106) (non-standard read/write)	30
3.2.15	SSCRATCH (0x140) (standard read/write)	30
3.2.16	SEPC (0x141) (standard read/write)	31
3.2.17	SCAUSE (0x142) (standard read/write)	31
3.2.18	STVAL (0x143) (standard read/write)	32
3.2.19	SIP (0x144) (standard read/write)	33
3.2.20	SATP (0x180) (standard read/write)	33
3.2.21	MSTATUS (0x300) (standard read/write)	34
3.2.22	MISA (0x301) (standard read/write)	36
3.2.23	MEDELEG (0x302) (standard read/write)	38
3.2.24	MIDELEG (0x303) (standard read/write)	39
3.2.25	MIE (0x304) (standard read/write)	40
3.2.26	MTVEC (0x305) (standard read/write)	42
3.2.27	MCOUNTEREN (0x306) (standard read/write)	42
3.2.28	MCOUNTINHIBIT (0x320) (non-standard read/write)	42
3.2.29	MHPMEVENT3 (0x323) (standard read/write)	43
3.2.30	MHPMEVENT4 (0x324) (standard read/write)	46
3.2.31	MHPMEVENT5 (0x325) (standard read/write)	47
3.2.32	MHPMEVENT6 (0x326) (standard read/write)	47
3.2.33	MSCRATCH (0x340) (standard read/write)	48
3.2.34	MEPC (0x341) (standard read/write)	48
3.2.35	MCAUSE (0x342) (standard read/write)	48
3.2.36	MTVAL (0x343) (standard read/write)	49
3.2.37	MIP (0x344) (standard read/write)	50
3.2.38	PMPCFG0 (0x3A0) (standard read/write)	51
3.2.39	PMPCFG1 (0x3A1) (standard read/write)	52
3.2.40	PMPCFG2 (0x3A2) (standard read/write)	52
3.2.41	PMPCFG3 (0x3A3) (standard read/write)	52
3.2.42	PMPADDR[PMPADDR0] (0x3B0 + 0x1 * n) (standard read/write)	54
3.2.43	TSELECT (0x7A0) (standard read/write)	54
3.2.44	TDATA1 (0x7A1) (standard read/write)	55
3.2.45	MCONTROL (0x7A1) (standard read/write)	55
3.2.46	ICOUNT (0x7A1) (standard read/write)	57
3.2.47	ITRIGGER (0x7A1) (standard read/write)	57
3.2.48	ETRIGGER (0x7A1) (standard read/write)	58
3.2.49	TDATA2 (0x7A2) (standard read/write)	59

3.2.50	TDATA3 (0x7A3) (standard read/write)	59
3.2.51	TEXTRA (0x7A3) (standard read/write)	59
3.2.52	TINFO (0x7A4) (standard read/write)	60
3.2.53	TCONTROL (0x7A5) (standard read/write)	61
3.2.54	MCONTEXT (0x7A8) (standard read/write)	61
3.2.55	SCONTEXT (0x7AA) (standard read/write)	62
3.2.56	DCSR (0x7B0) (debug-mode-only)	62
3.2.57	DPC (0x7B1) (debug-mode-only)	64
3.2.58	DSCRATCH0 (0x7B2) (debug-mode-only)	64
3.2.59	DSCRATCH1 (0x7B3) (debug-mode-only)	65
3.2.60	MILMB (0x7C0) (non-standard read/write)	65
3.2.61	MDLMB (0x7C1) (non-standard read/write)	66
3.2.62	MECC_CODE (0x7C2) (non-standard read/write)	67
3.2.63	MNVEC (0x7C3) (non-standard read/write)	68
3.2.64	MXSTATUS (0x7C4) (non-standard read/write)	68
3.2.65	MPFT_CTL (0x7C5) (non-standard read/write)	69
3.2.66	MHSP_CTL (0x7C6) (non-standard read/write)	70
3.2.67	MSP_BOUND (0x7C7) (non-standard read/write)	71
3.2.68	MSP_BASE (0x7C8) (non-standard read/write)	71
3.2.69	MDCAUSE (0x7C9) (non-standard read/write)	71
3.2.70	MCACHE_CTL (0x7CA) (non-standard read/write)	72
3.2.71	MCCTLBEGINADDR (0x7CB) (non-standard read/write)	74
3.2.72	MCCTLCOMMAND (0x7CC) (non-standard read/write)	75
3.2.73	MCCTLDATA (0x7CD) (non-standard read/write)	76
3.2.74	MCOUNTERWEN (0x7CE) (non-standard read/write)	76
3.2.75	MCOUNTERINTEN (0x7CF) (non-standard read/write)	77
3.2.76	MMISC_CTL (0x7D0) (non-standard read/write)	77
3.2.77	MCOUNTERMASK_M (0x7D1) (non-standard read/write)	78
3.2.78	MCOUNTERMASK_S (0x7D2) (non-standard read/write)	79
3.2.79	MCOUNTERMASK_U (0x7D3) (non-standard read/write)	79
3.2.80	MCOUNTEROVF (0x7D4) (non-standard read/write)	80
3.2.81	MSLIDELEG (0x7D5) (non-standard read/write)	80
3.2.82	MCLK_CTL (0x7DF) (non-standard read/write)	81
3.2.83	DEXC2DBG (0x7E0) (non-standard read/write)	82
3.2.84	DDCAUSE (0x7E1) (non-standard read/write)	84
3.2.85	UITB (0x800) (non-standard read/write)	85
3.2.86	UCODE (0x801) (non-standard read/write)	86
3.2.87	UDCAUSE (0x809) (non-standard read/write)	86
3.2.88	UCCTLBEGINADDR (0x80B) (non-standard read/write)	87
3.2.89	UCCTLCOMMAND (0x80C) (non-standard read/write)	88
3.2.90	SLIE (0x9C4) (non-standard read/write)	89
3.2.91	SLIP (0x9C5) (non-standard read/write)	89
3.2.92	SDCAUSE (0x9C9) (non-standard read/write)	90

3.2.93	SCCTLDATA (0x9CD) (non-standard read/write)	91
3.2.94	SCOUNTERINTEN (0x9CF) (non-standard read/write)	92
3.2.95	SCOUNTERMASK_M (0x9D1) (non-standard read/write)	92
3.2.96	SCOUNTERMASK_S (0x9D2) (non-standard read/write)	93
3.2.97	SCOUNTERMASK_U (0x9D3) (non-standard read/write)	93
3.2.98	SCOUNTEROVF (0x9D4) (non-standard read/write)	93
3.2.99	SCOUNTINHIBIT (0x9E0) (non-standard read/write)	94
3.2.100	SHPMEVENT3 (0x9E3) (non-standard read/write)	94
3.2.101	SHPMEVENT4 (0x9E4) (non-standard read/write)	95
3.2.102	SHPMEVENT5 (0x9E5) (non-standard read/write)	95
3.2.103	SHPMEVENT6 (0x9E6) (non-standard read/write)	95
3.2.104	MCYCLE (0xB00) (standard read/write)	96
3.2.105	MINSTRET (0xB02) (standard read/write)	96
3.2.106	MHPMCOUNTER3 (0xB03) (standard read/write)	97
3.2.107	MHPMCOUNTER4 (0xB04) (standard read/write)	97
3.2.108	MHPMCOUNTER5 (0xB05) (standard read/write)	97
3.2.109	MHPMCOUNTER6 (0xB06) (standard read/write)	98
3.2.110	MCYCLEH (0xB80) (standard read/write)	98
3.2.111	MINSTRETH (0xB82) (standard read/write)	98
3.2.112	MHPMCOUNTER3H (0xB83) (standard read/write)	99
3.2.113	MHPMCOUNTER4H (0xB84) (standard read/write)	99
3.2.114	MHPMCOUNTER5H (0xB85) (standard read/write)	99
3.2.115	MHPMCOUNTER6H (0xB86) (standard read/write)	100
3.2.116	PMACFG0 (0xBC0) (standard read/write)	100
3.2.117	PMACFG1 (0xBC1) (standard read/write)	100
3.2.118	PMACFG2 (0xBC2) (standard read/write)	101
3.2.119	PMACFG3 (0xBC3) (standard read/write)	101
3.2.120	PMAADDR[PMAADDR0] (0xBD0 + 0x1 * n) (standard read/write)	103
3.2.121	CYCLE (0xC00) ()	103
3.2.122	CYCLEH (0xC80) ()	104
3.2.123	MVENDORID (0xF11) (standard read only)	104
3.2.124	MARCHID (0xF12) (standard read only)	104
3.2.125	MIMPID (0xF13) (standard read only)	105
3.2.126	MHARTID (0xF14) (standard read only)	105
4	Exceptions and interrupts for TRAP processor	106
4.1	RISC-V Processor TRAP Overview	106
4.1.1	exception	106
4.1.2	interrupt	107
4.1.3	Platform Interrupt Controller PLIC	107
4.1.4	Software Interrupt Controller PLICSW	107
4.1.5	Machine Timer MCHTMR	108
4.2	TRAP and Processor Privilege Mode	108
4.3	Interrupt nesting	109

4.4	interrupt vector	110
4.5	interrupt response process	113
5	Machine Timer MCHTMR	115
5.1	Feature Summary	115
5.2	Function Description	115
5.2.1	Timing and Timing Function	115
5.2.2	Timer reset	115
5.3	MCHTMR register description	116
5.4	MCHTMR register details	116
5.4.1	MTIME (0x0)	116
5.4.2	MTIMECMP (0x8)	117
6	Platform-level interrupt controller PLIC	118
6.1	Feature Summary	118
6.2	Function Description	118
6.2.1	Vector Interrupt Extension	118
6.2.2	Interrupt Priority Preemption	118
6.3	PLIC Register Description	119
6.4	PLIC register details	122
6.4.1	FEATURE (0x0)	122
6.4.2	PRIORITY (0x4 + 0x4 * n)	123
6.4.3	PENDING (0x1000 + 0x4 * n)	123
6.4.4	TRIGGER (0x1080 + 0x4 * n)	124
6.4.5	NUMBER (0x1100)	124
6.4.6	INFO (0x1104)	124
6.4.7	TARGETINT[INTEN] (0x2000 + 0x80 * n + 0x4 * m)	125
6.4.8	TARGETCONFIG[THRESHOLD] (0x200000 + 0x1000 * n)	125
6.4.9	TARGETCONFIG[CLAIM] (0x200004 + 0x1000 * n)	126
6.4.10	TARGETCONFIG[PPS] (0x200400 + 0x1000 * n)	126
7	Platform-level Software Interrupt Controller PLICSW	127
7.1	Feature Summary	127
7.2	Function Description	127
7.3	PLICSW Register Description	127
7.3.1	PLICSW register details	127
7.3.2	PENDING (0x1000)	127
7.3.3	INTEN (0x2000)	128
7.3.4	CLAIM (0x200004)	128
8	Power Management	129
8.1	Power Supply System Structure	129
8.2	Power Supply System	129
8.2.1	Switching Power Supply DCDC	131
8.2.2	Power Management Domain Linear Regulator LDOPMC	131
8.2.3	OTP Linear Regulator LDOOTP	132

8.3	IO power supply	132
8.4	Power Supply Pin Notes	132
8.5	Overview of Low Power Consumption	133
8.6	IO related to power management functions	134
9	System Reset	135
9.1	Reset Overview	135
9.2	Global Reset	135
9.3	Reset the system power domain	135
10	Clock System	136
10.1	Clock System Overview	136
10.2	Clock Source	136
10.2.1	32KHz clock source CLK_32K	136
10.2.2	24MHz clock source CLK_24M	136
10.2.3	Phase-locked loop PLL	137
10.3	Function Clock	137
10.4	Module that directly uses the clock source	139
11	Power Management Domain Switch Module PDGO	141
11.1	Main features	141
11.2	CLK_32K Clock Source Management	141
11.3	Global reset pin RESETN control	141
11.4	Power Management Domain Switch Control	141
11.4.1	Entering shutdown mode	141
11.4.2	Exit of shutdown mode	141
11.4.2.1	Software Countdown Single or Periodic Wake-up	142
11.4.2.2	pin WAKEUP wake-up	142
11.4.2.3	Reset Pin RESETN Wake-up	142
11.5	General purpose register	142
11.6	PDGO register description	142
11.7	PDGO register details	144
11.7.1	DGO_TURNOFF (0x0)	144
11.7.2	DGO_RC32K_CFG (0x4)	144
11.7.3	DGO_CTRL (0x8)	145
11.7.4	PIN_CTRL (0x10)	145
11.7.5	PIN_STS (0x14)	146
11.7.6	WAKEUP (0x18)	147
11.7.7	WKUP_CYC (0x1C)	147
11.7.8	WKUP_EN (0x20)	147
11.7.9	WKUP_STS (0x24)	148
11.7.10	IRQ0_EN (0x60)	149
11.7.11	IRQ0_STS (0x64)	150
11.7.12	IRQ1_EN (0x68)	151
11.7.13	IRQ1_STS (0x6C)	152

11.7.14	GPR (0x80 + 0x4 * n)	153
11.7.15	IOFILTER (0x100 + 0x4 * n)	153
11.7.16	PCAP[CTRL] (0x200 + 0x18 * n)	154
11.7.17	PCAP[CP_CFG] (0x204 + 0x18 * n)	156
11.7.18	PCAP[COUNTER] (0x210 + 0x18 * n)	156
11.7.19	PCAP[DUMP] (0x214 + 0x18 * n)	157
11.7.20	PCNT[CNT_CFG] (0x280 + 0x30 * n)	157
11.7.21	PCNT[CALIB_CFG] (0x284 + 0x30 * n)	158
11.7.22	PCNT[RESOLUTION] (0x288 + 0x30 * n)	160
11.7.23	PCNT[HOMING_CFG] (0x28C + 0x30 * n)	160
11.7.24	PCNT[ANALYSE_CFG] (0x290 + 0x30 * n)	163
11.7.25	PCNT[DIRECTION] (0x294 + 0x30 * n)	164
11.7.26	PCNT[COMP0] (0x298 + 0x30 * n)	164
11.7.27	PCNT[COMP1] (0x29C + 0x30 * n)	165
11.7.28	PCNT[CNT] (0x2A0 + 0x30 * n)	165
11.7.29	PCNT[TACHO] (0x2A4 + 0x30 * n)	166
11.7.30	PCNT[DUMP] (0x2A8 + 0x30 * n)	167
11.7.31	PCNT[STS] (0x2AC + 0x30 * n)	167
12	Power Management Domain Configuration Module PCFG	169
12.1	Feature Summary	169
12.2	Function Description	169
12.2.1	Linear Regulator LDOOTP Configuration	169
12.2.2	Switching Power Supply DCDC Configuration	169
12.2.3	Clock gating for Power Management Domain Module	170
12.2.4	System power domain wake-up	170
12.2.5	Debugging Control	170
12.2.6	Power Management Domain Reset Control Module PPOR	171
12.3	PCFG register description	171
12.4	PCFG register details	172
12.4.1	BANDGAP (0x0)	172
12.4.2	LDO1P1 (0x4)	172
12.4.3	LDO2P5 (0x8)	173
12.4.4	DCDC_MODE (0x10)	173
12.4.5	DCDC_LPMODE (0x14)	174
12.4.6	DCDC_PROT (0x18)	175
12.4.7	DCDC_CURRENT (0x1C)	176
12.4.8	DCDC_ADVMODE (0x20)	176
12.4.9	DCDC_ADVPARAM (0x24)	177
12.4.10	DCDC_MISC (0x28)	178
12.4.11	DCDC_DEBUG (0x2C)	178
12.4.12	DCDC_START_TIME (0x30)	179
12.4.13	DCDC_RESUME_TIME (0x34)	179
12.4.14	POWER_TRAP (0x40)	179

12.4.15	WAKE_CAUSE (0x44)	180
12.4.16	WAKE_MASK (0x48)	181
12.4.17	SCG_CTRL (0x4C)	182
12.4.18	RC24M (0x60)	182
12.4.19	RC24M_TRACK (0x64)	183
12.4.20	TRACK_TARGET (0x68)	183
12.4.21	STATUS (0x6C)	183
12.5	PPOR register description	184
12.6	PPOR register details	184
12.6.1	RESET_FLAG (0x0)	184
12.6.2	RESET_STATUS (0x4)	185
12.6.3	RESET_HOLD (0x8)	186
12.6.4	RESET_ENABLE (0xC)	187
12.6.5	RESET_TYPE (0x10)	188
12.6.6	SOFTWARE_RESET (0x1C)	189
13	System Control Module (SYSCTL)	190
13.1	Feature Summary	190
13.2	Functional Clock Configuration	190
13.3	CPU Boot Management	190
13.3.1	Boot Management	190
13.4	System Power Domain Resource Management	190
13.4.1	Resource Nodes	190
13.4.2	Chain Structure of Resource Nodes	191
13.4.3	Automatic Shutdown Mechanism of Resource Nodes	192
13.4.4	Resource Node Keep-ON	193
13.4.5	Connection between Functional Modules and Processors	193
13.4.6	Usage of Resource Nodes	196
13.5	Low Power Management	196
13.5.1	Processor Low Power Modes and Wake-up	196
13.5.2	State Retention of Resource Nodes in Low Power Mode	197
13.6	Clock Measurement Module	198
13.7	SYSCTL Register Description	200
13.8	Detailed Register Information	211
13.8.1	RESOURCE (0x0 + 0x4 * n)	211
13.8.2	GROUP0[VALUE] (0x800 + 0x10 * n)	211
13.8.3	GROUP0[SET] (0x804 + 0x10 * n)	212
13.8.4	GROUP0[CLEAR] (0x808 + 0x10 * n)	212
13.8.5	GROUP0[TOGGLE] (0x80C + 0x10 * n)	212
13.8.6	GROUP1[VALUE] (0x840 + 0x10 * n)	213
13.8.7	GROUP1[SET] (0x844 + 0x10 * n)	213
13.8.8	GROUP1[CLEAR] (0x848 + 0x10 * n)	214
13.8.9	GROUP1[TOGGLE] (0x84C + 0x10 * n)	214
13.8.10	AFFILIATE[VALUE] (0x900 + 0x10 * n)	214

13.8.11	AFFILIATE[SET] (0x904 + 0x10 * n)	215
13.8.12	AFFILIATE[CLEAR] (0x908 + 0x10 * n)	215
13.8.13	AFFILIATE[TOGGLE] (0x90C + 0x10 * n)	216
13.8.14	RETENTION[VALUE] (0x920 + 0x10 * n)	216
13.8.15	RETENTION[SET] (0x924 + 0x10 * n)	217
13.8.16	RETENTION[CLEAR] (0x928 + 0x10 * n)	217
13.8.17	RETENTION[TOGGLE] (0x92C + 0x10 * n)	218
13.8.18	POWER[STATUS] (0x1000 + 0x10 * n)	218
13.8.19	POWER[LF_WAIT] (0x1004 + 0x10 * n)	219
13.8.20	POWER[OFF_WAIT] (0x100C + 0x10 * n)	220
13.8.21	RESET[CONTROL] (0x1400 + 0x10 * n)	220
13.8.22	RESET[CONFIG] (0x1404 + 0x10 * n)	221
13.8.23	RESET[COUNTER] (0x140C + 0x10 * n)	222
13.8.24	CLOCK (0x1800 + 0x4 * n)	222
13.8.25	ADCCLK (0x1C00 + 0x4 * n)	223
13.8.26	GLOBAL00 (0x2000)	224
13.8.27	MONITOR[CONTROL] (0x2400 + 0x20 * n)	224
13.8.28	MONITOR[CURRENT] (0x2404 + 0x20 * n)	225
13.8.29	MONITOR[LOW_LIMIT] (0x2408 + 0x20 * n)	225
13.8.30	MONITOR[HIGH_LIMIT] (0x240C + 0x20 * n)	226
13.8.31	CPU[LP] (0x2800 + 0x400 * n)	226
13.8.32	CPU[LOCK] (0x2804 + 0x400 * n)	227
13.8.33	CPU[GPR] (0x2808 + 0x400 * n + 0x4 * m)	228
13.8.34	CPU[WAKEUP_STATUS] (0x2840 + 0x400 * n + 0x4 * m)	228
13.8.35	CPU[WAKEUP_ENABLE] (0x2880 + 0x400 * n + 0x4 * m)	228
14	PLL Controller PLLCTL	230
14.1	Feature Summary	230
14.2	Function Description	230
14.2.1	XTAL oscillator	230
14.2.2	PLL reference clock settings	230
14.2.3	PLL operating frequency configuration	230
14.2.4	PLL spread spectrum mode	231
14.2.5	PLL output divider	231
14.3	PLLCTL register	231
14.3.1	register description	231
14.3.2	XTAL (0x0)	233
14.3.3	PLL[MFI] (0x80 + 0x80 * n)	234
14.3.4	PLL[MFN] (0x84 + 0x80 * n)	235
14.3.5	PLL[MFD] (0x88 + 0x80 * n)	235
14.3.6	PLL[SS_STEP] (0x8C + 0x80 * n)	235
14.3.7	PLL[SS_STOP] (0x90 + 0x80 * n)	236
14.3.8	PLL[CONFIG] (0x94 + 0x80 * n)	236
14.3.9	PLL[LOCKTIME] (0x98 + 0x80 * n)	237

14.3.10	PLL[STEPTIME] (0x9C + 0x80 * n)	237
14.3.11	PLL[ADVANCED] (0xA0 + 0x80 * n)	237
14.3.12	PLL[DIV] (0xC0 + 0x80 * n + 0x4 * m)	238
15	Low Power Management	240
15.1	Running Mode RUN	240
15.2	Waiting Mode WAIT	240
15.3	Stop Mode STOP	240
15.4	Sleep Mode STANDBY	241
15.5	Shutdown mode SHUTDOWN	242
16	System Memory Mapping	243
16.1	System Memory Map	243
17	OTP Mapping Table	246
18	BootROM	247
18.1	BootROM Overview	247
18.2	Boot Flow	248
18.2.1	Boot Flow Diagram	248
18.2.2	XPI NOR Boot	248
18.2.2.1	Supported Serial NOR Devices	248
18.2.2.2	FLASH Connection Methods	249
18.2.2.3	XPI NOR Boot Flow	250
18.2.2.4	XPI NOR Boot Image Layout	250
18.2.2.5	Execute-In-Place with Decryption	251
18.2.3	In-System Programming (In-System-Programming)	251
18.2.3.1	Command Protocol	252
18.2.4	Serial Port Communication	268
18.2.4.1	Payload Packet	268
18.2.4.2	Acknowledgment Packet (ACK Packet)	269
18.2.5	USB-HID Communication	269
18.2.5.1	USB-HID Descriptor Information	269
18.2.5.2	USB Endpoint Information	270
18.2.5.3	USB-HID Communication Protocol	270
18.3	Boot Image	271
18.3.1	Firmware Container Header	271
18.3.2	Firmware Info Table	273
18.3.3	Device Configuration Info Block	274
18.3.3.1	XPI NOR Configuration Info	275
18.3.3.2	Wake-up Verification Info	280
18.3.3.3	OTP Unlock Info	280
18.3.4	Signature Block	281
18.3.4.1	Signature Block Header	281
18.3.5	Root Key Table (SRK Table)	282
18.3.5.1	Root Key Table Header	282

18.3.5.2	Root Key Table Item	282
18.3.6	Signature Info	283
18.3.7	Certificate	283
18.3.8	Command Container	284
18.3.8.1	Command Container Header	284
18.3.8.2	Commands Supported by Command Container	284
18.3.8.3	Encrypted Command Container	286
18.4	Secure Boot	287
18.4.1	Secure Boot Introduction	287
18.4.2	Secure Boot Flow	287
18.4.3	Firmware Container Signature Verification	288
18.4.4	Firmware Signature Verification	288
18.4.5	Public Key Revocation	289
18.5	Low Power Wake-up	289
18.5.1	Fast Jump	289
18.5.2	Strong Verification Jump	290
18.5.3	Disable Low Power Wake-up	290
18.6	Debug Interface Permission Management	290
18.7	ROM API	290
18.7.1	Introduction	291
18.7.2	run_bootloader API	291
18.7.2.1	Example:	292
18.7.3	OTP API	292
18.7.3.1	init	292
18.7.3.2	read_from_shadow	293
18.7.3.3	read_from_ip	293
18.7.3.4	program	293
18.7.3.5	reload	293
18.7.3.6	lock_otp	293
18.7.3.7	set_configurable_region	293
18.7.3.8	write_shadow_register	293
18.7.3.9	Example	294
18.7.4	XPI Low-level Driver API	294
18.7.4.1	get_default_config	294
18.7.4.2	get_default_device_config	295
18.7.4.3	init	295
18.7.4.4	config_ahb_buffer	295
18.7.4.5	config_device	295
18.7.4.6	update_instr_table	296
18.7.4.7	transfer_blocking	296
18.7.4.8	software_reset	296
18.7.4.9	is_idle	296
18.7.4.10	update_dllcr	296

18.7.5	XPI NOR API	297
18.7.5.1	get_config	298
18.7.5.2	init	298
18.7.5.3	enable_write	298
18.7.5.4	get_status	298
18.7.5.5	wait_busy	299
18.7.5.6	erase	299
18.7.5.7	erase_chip	299
18.7.5.8	erase_sector	299
18.7.5.9	erase_block	299
18.7.5.10	program	299
18.7.5.11	read	300
18.7.5.12	page_program_nonblocking	300
18.7.5.13	erase_sector_nonblocking	300
18.7.5.14	erase_block_nonblocking	300
18.7.5.15	erase_chip_nonblocking	300
18.7.6	XPI RAM API	301
18.7.6.1	get_config	301
18.7.6.2	init	301
18.7.7	Secure Boot API	301
18.8	SoC Specific Information	302
18.8.1	OTP Mapping Table	302
18.8.2	BootROM Supported Peripheral Information	305
18.8.2.1	XPI Information	305
18.8.2.2	UART Pins	306
18.8.2.3	USB Pins	306
18.8.2.4	Boot Mode (BOOT_MODE) Pins	306
18.8.3	BootROM Pre-occupied Register Information	306
18.8.4	General Register Configuration Supported Register Range	307
18.8.5	BootROM Memory Mapping Table	307
18.8.6	BootROM Lifecycle	307
19	Pin Configuration and Function PINMUX	309
19.1	IO function allocation	309
19.2	System Power Out-Field Pin Allocation	352
19.3	Special Function Pins	394
19.4	IO reset status	394
20	Input and Output Module	395
20.1	IO Controller	395
20.2	GPIO Controller	395
20.3	GPIO Management GPIOM	396
21	IO Controller IOC	397
21.1	Feature Summary	397

21.2	Function Description	397
21.2.1	IO basic configuration	397
21.2.2	IO peripheral function configuration	397
21.3	IOC register	398
21.3.1	register description	398
21.3.2	PAD[FUNC_CTL] (0x0 + 0x8 * n)	407
21.3.3	PAD[PAD_CTL] (0x4 + 0x8 * n)	408
22	GPIO Controller	411
22.1	Feature Summary	411
22.2	Function Description	411
22.2.1	GPIO Control	411
22.2.2	GPIO interrupt	411
22.3	GPIO register list	412
22.4	GPIO register description	422
22.4.1	DI[VALUE] (0x0 + 0x10 * n)	422
22.4.2	DO[VALUE] (0x100 + 0x10 * n)	422
22.4.3	DO[SET] (0x104 + 0x10 * n)	423
22.4.4	DO[CLEAR] (0x108 + 0x10 * n)	423
22.4.5	DO[TOGGLE] (0x10C + 0x10 * n)	423
22.4.6	OE[VALUE] (0x200 + 0x10 * n)	424
22.4.7	OE[SET] (0x204 + 0x10 * n)	424
22.4.8	OE[CLEAR] (0x208 + 0x10 * n)	425
22.4.9	OE[TOGGLE] (0x20C + 0x10 * n)	425
22.4.10	IF[VALUE] (0x300 + 0x10 * n)	425
22.4.11	IE[VALUE] (0x400 + 0x10 * n)	426
22.4.12	IE[SET] (0x404 + 0x10 * n)	426
22.4.13	IE[CLEAR] (0x408 + 0x10 * n)	427
22.4.14	IE[TOGGLE] (0x40C + 0x10 * n)	427
22.4.15	PL[VALUE] (0x500 + 0x10 * n)	427
22.4.16	PL[SET] (0x504 + 0x10 * n)	428
22.4.17	PL[CLEAR] (0x508 + 0x10 * n)	428
22.4.18	PL[TOGGLE] (0x50C + 0x10 * n)	429
22.4.19	TP[VALUE] (0x600 + 0x10 * n)	429
22.4.20	TP[SET] (0x604 + 0x10 * n)	429
22.4.21	TP[CLEAR] (0x608 + 0x10 * n)	430
22.4.22	TP[TOGGLE] (0x60C + 0x10 * n)	430
22.4.23	AS[VALUE] (0x700 + 0x10 * n)	431
22.4.24	AS[SET] (0x704 + 0x10 * n)	431
22.4.25	AS[CLEAR] (0x708 + 0x10 * n)	432
22.4.26	AS[TOGGLE] (0x70C + 0x10 * n)	432
22.4.27	PD[VALUE] (0x800 + 0x10 * n)	432
22.4.28	PD[SET] (0x804 + 0x10 * n)	433
22.4.29	PD[CLEAR] (0x808 + 0x10 * n)	433

22.4.30	PD[TOGGLE] (0x80C + 0x10 * n)	434
23	GPIO Manager GPIOM	435
23.1	Feature Summary	435
23.2	Function Description	435
23.2.1	GPIO allocation	435
23.2.2	Access Control	435
23.3	GPIOM register list	436
23.4	GPIOM register description	444
23.4.1	ASSIGN[PIN] (0x0 + 0x80 * n + 0x4 * m)	444
24	EUI	445
24.1	Feature Summary	445
24.2	EUI Block Diagram	445
24.3	Functional Description	445
24.3.1	EUI Port Interface Description	447
24.3.2	Output Mode and Display	448
24.3.3	Drive Timing Parameters	448
24.3.4	Receiving Serial Data	449
24.3.5	Interrupt Function Description	449
24.3.6	Register Configuration Guide	449
24.4	EUI Registers	450
24.4.1	Register Overview	450
24.4.2	Register Details	451
24.4.3	CTRL (0x0)	451
24.4.4	DIV_CFG (0x4)	452
24.4.5	PAD_CFG (0x8)	452
24.4.6	POL_CFG (0xC)	453
24.4.7	OUT_CFG (0x10)	453
24.4.8	IN_CFG (0x14)	454
24.4.9	IRQ_STS (0x20)	454
24.4.10	IRQ_EN (0x24)	455
24.4.11	DATA_IN0 (0x40)	455
24.4.12	DATA_IN1 (0x44)	456
24.4.13	DATA_IN2 (0x48)	456
24.4.14	DATA_IN3 (0x4C)	456
24.4.15	DATA_IN4 (0x50)	457
24.4.16	DATA_IN5 (0x54)	457
24.4.17	DATA_IN6 (0x58)	457
24.4.18	DATA_IN7 (0x5C)	458
24.4.19	DAB_CTRL (0x80)	458
24.4.20	DCD_CTRL (0x90)	459
24.4.21	DEF_CTRL (0xA0)	459
24.4.22	DGH_CTRL (0xB0)	459

24.4.23	DA_00 (0x100)	460
24.4.24	DA_01 (0x104)	460
24.4.25	DB_00 (0x140)	460
24.4.26	DB_01 (0x144)	461
24.4.27	DC_00 (0x180)	461
24.4.28	DC_01 (0x184)	461
24.4.29	DD_00 (0x1C0)	462
24.4.30	DD_01 (0x1C4)	462
24.4.31	DE_00 (0x200)	462
24.4.32	DE_01 (0x204)	463
24.4.33	DF_00 (0x240)	463
24.4.34	DF_01 (0x244)	463
24.4.35	DG_00 (0x280)	464
24.4.36	DG_01 (0x284)	464
24.4.37	DH_00 (0x2C0)	464
24.4.38	DH_01 (0x2C4)	465
25	Logic Observation Module LOBS	466
25.1	Function Description	466
25.2	Feature Summary	466
25.3	Configuration Notes	467
25.4	LOBS register description	470
25.4.1	CTRL (0x0)	470
25.4.2	IRQ_EN (0x4)	470
25.4.3	IRQ_STS (0x8)	471
25.4.4	STREAMCTRL (0xC)	471
25.4.5	PTACTION (0x10)	472
25.4.6	STARTADDR (0x14)	472
25.4.7	ENDADDR (0x18)	473
25.4.8	CTSR (0x20)	473
25.4.9	CCVR (0x24)	474
25.4.10	CAVR (0x28)	474
25.4.11	FIFOSTATE (0x30)	475
25.4.12	FINALADDR (0x34)	475
25.4.13	TRIGADDR1 (0x40)	475
25.4.14	TRIGADDR2 (0x44)	476
25.4.15	TRIGADDR3 (0x48)	476
25.4.16	TRIGADDR4 (0x4C)	476
25.4.17	GRPSELA (0x50)	477
25.4.18	GRPENA (0x54)	477
25.4.19	SIGSELA1 (0x58)	477
25.4.20	SIGSELA2 (0x60)	478
25.4.21	SIGENA (0x98)	478
25.4.22	STATE[SIGSEL] (0x100 + 0x100 * n)	479

25.4.23	STATE[TRIGCTRL] (0x104 + 0x100 * n)	479
25.4.24	STATE[NEXTSTATE] (0x108 + 0x100 * n)	480
25.4.25	STATE[ACTION] (0x10C + 0x100 * n)	481
25.4.26	STATE[COUNTCOMP] (0x120 + 0x100 * n)	481
25.4.27	STATE[EXTMASK] (0x130 + 0x100 * n)	482
25.4.28	STATE[EXTCOMP] (0x134 + 0x100 * n)	482
25.4.29	STATE[SIGMASK] (0x140 + 0x100 * n)	482
25.4.30	STATE[COMPEN] (0x144 + 0x100 * n)	483
25.4.31	STATE[SIGCOMP0] (0x180 + 0x100 * n)	483
25.4.32	STATE[SIGCOMP1] (0x184 + 0x100 * n)	483
25.4.33	STATE[SIGCOMP2] (0x188 + 0x100 * n)	484
25.4.34	STATE[SIGCOMP3] (0x18C + 0x100 * n)	484
25.4.35	LAR (0xFB0)	484
26	Memory Overview	486
26.1	Internal SRAM	486
26.1.1	Local Memory	486
26.1.2	Universal Memory	486
26.2	General Purpose Register	486
26.2.1	PMIC_GPR00 (0x0)	488
26.2.2	PMIC_GPR01 (0x4)	488
26.2.3	PMIC_GPR02 (0x8)	488
26.2.4	PMIC_GPR03 (0xC)	489
26.2.5	PMIC_GPR04 (0x10)	489
26.2.6	PMIC_GPR05 (0x14)	489
26.2.7	PMIC_GPR06 (0x18)	489
26.2.8	PMIC_GPR07 (0x1C)	490
26.2.9	PMIC_GPR08 (0x20)	490
26.2.10	PMIC_GPR09 (0x24)	490
26.2.11	PMIC_GPR10 (0x28)	491
26.2.12	PMIC_GPR11 (0x2C)	491
26.2.13	PMIC_GPR12 (0x30)	491
26.2.14	PMIC_GPR13 (0x34)	492
26.2.15	PMIC_GPR14 (0x38)	492
26.2.16	PMIC_GPR15 (0x3C)	492
26.3	Programmable parallel port PPI	494
26.4	Serial Bus Controller XPI	494
26.5	Read Only Memory ROM	494
26.6	One-time Programmable Memory OTP	494
27	OTP and OTP Controller	495
27.1	Feature Summary	495
27.1.1	OTP Controller Feature Summary	495
27.2	OTP Controller Function Description	496

27.2.1	OTP Shadow Registers	496
27.3	OTP Fuse Registers	497
27.4	OTP Operation Engine	497
27.5	OTP Read and Write Protection	498
27.6	OTP Controller Registers	501
27.7	OTP Controller Register Details	508
27.7.1	SHADOW (0x0 + 0x4 * n)	508
27.7.2	SHADOW_LOCK (0x200 + 0x4 * n)	509
27.7.3	FUSE (0x400 + 0x4 * n)	509
27.7.4	FUSE_LOCK (0x600 + 0x4 * n)	509
27.7.5	UNLOCK (0x800)	510
27.7.6	DATA (0x804)	510
27.7.7	ADDR (0x808)	511
27.7.8	CMD (0x80C)	511
27.7.9	LOAD_REQ (0xA00)	511
27.7.10	LOAD_COMP (0xA04)	512
27.7.11	REGION (0xA20 + 0x4 * n)	512
27.7.12	INT_FLAG (0xC00)	513
27.7.13	INT_EN (0xC04)	513
28	Expanded Serial Peripheral Interface XPI	515
28.1	Feature Summary	515
28.2	Function Description	515
28.2.1	External device connection configuration	515
28.2.2	XPI address mapping	516
28.2.3	External Device Address Translation	516
28.2.4	instruction list	516
28.2.5	Bus addressed access	517
28.2.6	register instruction access	518
28.2.7	XPI Time Characteristics	518
28.2.8	interrupt	518
28.3	XPI Configuration and Use	519
29	Programmable parallel port PPI	520
29.1	Overview	520
29.2	PPI Architecture Diagram	520
29.3	External Interface Pin Description	520
29.4	Feature Summary	520
29.5	Function Description	521
29.5.1	PPI_AD[31:0] Address data allocation	521
29.5.2	Read and write cycle settings	521
29.5.2.1	Read Cycle	522
29.5.2.2	Write Cycle	523
29.5.2.3	chip bus read and write bits are wider than peripheral data bits wide	523

29.5.2.4	The read and write bit width of the chip bus is smaller than the peripheral data bit width	524
29.5.3	Clock Output	524
29.5.4	Address mapping conversion relationship	525
29.5.5	interrupt	526
29.6	Programmable Parallel Port Register	526
29.6.1	register description	526
29.6.2	GLB_CFG (0x0)	531
29.6.3	PAD_CFG (0x4)	532
29.6.4	CLKPIN_CFG (0x14)	532
29.6.5	TM_CFG (0x18)	533
29.6.6	IRQ_STS (0x20)	533
29.6.7	IRQ_EN (0x24)	534
29.6.8	CS[CFG0] (0x40 + 0x20 * n)	534
29.6.9	CS[CFG1] (0x44 + 0x20 * n)	535
29.6.10	CS[CFG2] (0x48 + 0x20 * n)	535
29.6.11	CS[CFG3] (0x4C + 0x20 * n)	536
29.6.12	CS[CFG4] (0x50 + 0x20 * n)	536
29.6.13	CMD[CMD_CFG] (0x400 + 0x10 * n)	537
29.6.14	CMD[AD_CFG] (0x404 + 0x10 * n)	537
29.6.15	CMD[CTRL_CFG] (0x408 + 0x10 * n)	538
30	Algorithm Acceleration Overview	539
30.1	Transcendental Function Accelerator (TFA)	539
30.2	Cyclic Redundancy Check (CRC)	539
31	Cyclic Redundancy Detection CRC	540
31.1	Feature Summary	540
31.2	Function Description	540
31.3	CRC algorithm configuration	540
31.4	CRC register	541
31.5	CRC register details	542
31.5.1	CHN[PRE_SET] (0x0 + 0x40 * n)	542
31.5.2	CHN[CLR] (0x4 + 0x40 * n)	543
31.5.3	CHN[POLY] (0x8 + 0x40 * n)	544
31.5.4	CHN[INIT_DATA] (0xC + 0x40 * n)	544
31.5.5	CHN[XOROUT] (0x10 + 0x40 * n)	544
31.5.6	CHN[MISC_SETTING] (0x14 + 0x40 * n)	545
31.5.7	CHN[DATA] (0x18 + 0x40 * n)	545
31.5.8	CHN[RESULT] (0x1C + 0x40 * n)	545
32	Overview of DMA and Mailbox	547
32.1	DMA Controller	547
32.2	DMA Request Router	547
32.3	Mail Box MBX	549

33 DMA Controller	550
33.1 Overview	550
33.2 DMA Architecture Block Diagram	550
33.3 Pin Description	550
33.4 Functional Description	551
33.4.1 DMA Hardware Handshake	551
33.4.2 DMA Chained Transfer	551
33.4.3 Infinite Loop	551
33.4.4 Fixed Burst	551
33.4.5 Byte Swapping	552
33.4.6 Data Order	552
33.5 DMA Registers	553
33.6 DMA Register Details	563
33.6.1 IDMISC (0x4)	563
33.6.2 DMACFG (0x10)	563
33.6.3 DMACTRL (0x14)	565
33.6.4 CHABORT (0x18)	565
33.6.5 INTHALFSTS (0x24)	566
33.6.6 INTTCSTS (0x28)	566
33.6.7 INTABORTSTS (0x2C)	566
33.6.8 INTERRSTS (0x30)	567
33.6.9 CHEN (0x34)	567
33.6.10 CHCTRL[CTRL] (0x40 + 0x20 * n)	568
33.6.11 CHCTRL[TRANSIZE] (0x44 + 0x20 * n)	571
33.6.12 CHCTRL[SRCADDR] (0x48 + 0x20 * n)	571
33.6.13 CHCTRL[CHANREQCTRL] (0x4C + 0x20 * n)	572
33.6.14 CHCTRL[DSTADDR] (0x50 + 0x20 * n)	572
33.6.15 CHCTRL[SWAPTABLE] (0x54 + 0x20 * n)	572
33.6.16 CHCTRL[LLPOINTER] (0x58 + 0x20 * n)	573
33.7 DMA Configuration and Usage Instructions	573
34 DMA Request Router DMAMUX	575
34.1 Feature Summary	575
34.2 Function Description	575
34.3 DMAMUX Register	575
34.4 DMAMUX register details	577
34.4.1 MUXCFG (0x0 + 0x4 * n)	577
35 Communication Mailbox MBX	578
35.1 Feature Summary	578
35.2 Function Description	578
35.2.1 MBX register access interface	578
35.2.2 FIFO Management	578
35.2.3 interrupt	578

35.3 MBX register	579
35.4 MBX register details	579
35.4.1 CR (0x0)	579
35.4.2 SR (0x4)	580
35.4.3 TXREG (0x8)	582
35.4.4 RXREG (0xC)	583
35.4.5 TXWRD (0x10 + 0x4 * n)	583
35.4.6 RXWRD (0x20 + 0x4 * n)	583
36 Electric Control System Overview	585
36.1 PWM timer PWMV2	585
36.2 Quadrature encoder interface QEIV2	586
36.3 Quadrature encoder output QEOV2	587
36.4 Programmable Logic Unit PLB	587
36.5 $\Sigma\Delta$ signal receiving unit SDM	588
36.6 Synchronization timer SYNT	588
36.7 Interconnect Manager TRGM	589
36.7.1 trigger manager TRIGMUX	589
36.7.2 DMA manager DMAMUX	597
36.7.3 Digital filter FILTER	597
36.7.4 Location Transponder POS_MATRIX	599
36.7.5 ADC Data Transponder ADC_MATRIX	599
36.7.6 DAC data transponder DAC MATRIX	600
37 Pulse Width Modulation Module PWMv2	601
37.1 Feature Summary	601
37.2 Block Diagram	601
37.3 pin description	602
37.4 Function Description	602
37.4.1 shadow register	602
37.4.2 DAC Input	602
37.4.3 Time Baseline	603
37.4.4 Counter calculation unit	603
37.4.5 PWM Generation	604
37.4.6 Burst	605
37.4.7 Complementary PWM and dead zone	607
37.4.8 comparison value	608
37.4.9 General Purpose Computing Unit	609
37.4.10 Forced output	611
37.4.11 Fault Protection	612
37.4.12 Input Capture	614
37.4.13 Trigger input	614
37.4.14 Trigger output	614
37.4.15 interrupt	614

37.4.16 DMA Request	616
37.5 PWMv2 register	616
37.6 PWM register details	622
37.6.1 WORK_CTRL0 (0x0)	622
37.6.2 UNLOCK (0x4)	622
37.6.3 SHADOW_VAL (0x8 + 0x4 * n)	623
37.6.4 FORCE_MODE (0x78)	623
37.6.5 WORK_CTRL1 (0x7C)	624
37.6.6 PWM[CFG0] (0x100 + 0x10 * n)	624
37.6.7 PWM[CFG1] (0x104 + 0x10 * n)	625
37.6.8 PWM[DEAD_AREA] (0x108 + 0x10 * n)	627
37.6.9 PWM[CFG3] (0x10C + 0x10 * n)	627
37.6.10 TRIGGER_CFG (0x180 + 0x4 * n)	628
37.6.11 GLB_CTRL (0x1F0)	628
37.6.12 GLB_CTRL2 (0x1F4)	629
37.6.13 GLB_CTRL3 (0x1F8)	629
37.6.14 CNT_RELOAD_WORK (0x200 + 0x4 * n)	630
37.6.15 CMP_VAL_WORK (0x210 + 0x4 * n)	630
37.6.16 FORCE_WORK (0x27C)	631
37.6.17 CNT_VAL (0x2A0 + 0x4 * n)	631
37.6.18 DAC_VALUE_SV (0x2B0 + 0x4 * n)	631
37.6.19 CAPTURE_POS (0x300 + 0x4 * n)	632
37.6.20 CAPTURE_NEG (0x380 + 0x4 * n)	632
37.6.21 IRQ_STS (0x400)	633
37.6.22 IRQ_EN (0x404)	633
37.6.23 IRQ_STS_CMP (0x410)	634
37.6.24 IRQ_STS_RELOAD (0x414)	634
37.6.25 IRQ_STS_CAP_POS (0x418)	634
37.6.26 IRQ_STS_CAP_NEG (0x41C)	635
37.6.27 IRQ_STS_FAULT (0x420)	635
37.6.28 IRQ_STS_BURSTEND (0x424)	636
37.6.29 IRQ_EN_CMP (0x430)	636
37.6.30 IRQ_EN_RELOAD (0x434)	637
37.6.31 IRQ_EN_CAP_POS (0x438)	637
37.6.32 IRQ_EN_CAP_NEG (0x43C)	637
37.6.33 IRQ_EN_FAULT (0x440)	638
37.6.34 IRQ_EN_BURSTEND (0x444)	638
37.6.35 DMA_EN (0x480)	639
37.6.36 CNT[CFG0] (0x500 + 0x10 * n)	639
37.6.37 CNT[CFG1] (0x504 + 0x10 * n)	640
37.6.38 CNT[CFG2] (0x508 + 0x10 * n)	641
37.6.39 CNT[CFG3] (0x50C + 0x10 * n)	642
37.6.40 CNT_GLBCFG (0x540)	642

37.6.41	CAL[CFG0] (0x600 + 0x10 * n)	643
37.6.42	CAL[CFG1] (0x604 + 0x10 * n)	643
37.6.43	CMP[CFG] (0x800 + 0x10 * n)	644
38	Internet Manager TRGM	646
38.1	Feature Summary	646
38.2	Function Description	646
38.2.1	Manager Signal Input Output Checkbox	646
38.2.2	Output Configuration	646
38.2.3	Digital Filter	647
38.2.4	DMA Request Management	647
38.3	TRGM register list	647
38.4	TRGM register description	657
38.4.1	FILTCFG (0x0 + 0x4 * n)	657
38.4.2	DMACFG (0x400 + 0x4 * n)	658
38.4.3	GCR (0x500)	658
38.4.4	ADC_MATRIX_SEL0 (0x540)	658
38.4.5	DAC_MATRIX_SEL0 (0x580)	659
38.4.6	DAC_MATRIX_SEL1 (0x584)	659
38.4.7	DAC_MATRIX_SEL2 (0x588)	660
38.4.8	POS_MATRIX_SEL0 (0x5C0)	660
38.4.9	TRGM_IN (0x600 + 0x4 * n)	661
38.4.10	TRGM_OUT (0x680 + 0x4 * n)	661
38.4.11	PWM_DELAY_CFG (0x800)	662
38.4.12	PWM_CALIB_CFG (0x804)	662
38.4.13	TRGOCFG (0x1000 + 0x4 * n)	662
39	Sync Timer SYNT	664
39.1	Feature Summary	664
39.2	Function Description	664
39.2.1	Output synchronization events	664
39.2.2	Output Timestamp	665
39.2.3	Outputs DCCS_CLK to DCDC	665
39.2.3.1	Fixed ADC operating frequency mode	665
39.2.3.2	Fixed DCCS_CLK frequency mode	666
39.3	SYNT register list	666
39.4	SYNT register details	667
39.4.1	GCR (0x0)	667
39.4.2	RLD (0x4)	668
39.4.3	TIMESTAMP_NEW (0x8)	668
39.4.4	CNT (0xC)	669
39.4.5	TIMESTAMP_SAV (0x10)	669
39.4.6	TIMESTAMP_CUR (0x14)	669
39.4.7	CMP (0x20 + 0x4 * n)	670

40 Quadrature Encoder Interface QEIV2	671
40.1 Feature Summary	671
40.2 Block Diagram	671
40.3 Pin Description	672
40.4 Functional Description	672
40.4.1 QEIV2 Decoding Modes	672
40.4.1.1 Single-Phase Digital Mode	672
40.4.1.2 ABZ Mode	673
40.4.1.3 PD Mode	675
40.4.1.4 UD Mode	675
40.4.1.5 UVW Mode	676
40.4.1.6 Quadrature Sine-Cosine Wave Mode	677
40.4.1.7 Single-Phase Sine Wave Mode	678
40.4.2 Position Information Output	678
40.4.3 Position Matching	679
40.4.4 H Phase and F Phase Functions	679
40.4.5 Simple Speed Measurement	680
40.4.5.1 Based on Pulse Count	680
40.4.5.2 Based on Duration	680
40.4.6 Watchdog Counter	681
40.4.7 Counters Reading and Snapshot Function	681
40.4.8 Counter Reset	683
40.4.9 Counter Pause	683
40.4.10 PAD Filter Function of the QEI Module	683
40.4.11 Interrupts, DMA, and Trigger Output	683
40.5 QEI Register Configuration Guide	684
40.6 QEIV2 Register List	685
40.7 QEIV2 Register Details	688
40.7.1 CR (0x0)	688
40.7.2 PHCFG (0x4)	689
40.7.3 WDGCFG (0x8)	690
40.7.4 PHIDX (0xC)	690
40.7.5 TRGOEN (0x10)	691
40.7.6 READEN (0x14)	691
40.7.7 ZCMP (0x18)	692
40.7.8 PHCMP (0x1C)	692
40.7.9 SPDCMP (0x20)	693
40.7.10 DMAEN (0x24)	693
40.7.11 SR (0x28)	694
40.7.12 IRQEN (0x2C)	694
40.7.13 COUNT[Z] (0x30 + 0x10 * n)	695
40.7.14 COUNT[PH] (0x34 + 0x10 * n)	695
40.7.15 COUNT[SPD] (0x38 + 0x10 * n)	696

40.7.16 COUNT[TMR] (0x3C + 0x10 * n)	696
40.7.17 ZCMP2 (0x80)	697
40.7.18 PHCMP2 (0x84)	697
40.7.19 SPDCMP2 (0x88)	697
40.7.20 MATCH_CFG (0x8C)	698
40.7.21 FILT_CFG (0x90 + 0x4 * n)	699
40.7.22 QEI_CFG (0x100)	699
40.7.23 PULSE0_NUM (0x110)	701
40.7.24 PULSE1_NUM (0x114)	701
40.7.25 CYCLE0_CNT (0x118)	701
40.7.26 CYCLE0PULSE_CNT (0x11C)	702
40.7.27 CYCLE1_CNT (0x120)	702
40.7.28 CYCLE1PULSE_CNT (0x124)	703
40.7.29 CYCLE0_SNAP0 (0x128)	703
40.7.30 CYCLE0_SNAP1 (0x12C)	703
40.7.31 CYCLE1_SNAP0 (0x130)	704
40.7.32 CYCLE1_SNAP1 (0x134)	704
40.7.33 CYCLE0_NUM (0x140)	705
40.7.34 CYCLE1_NUM (0x144)	705
40.7.35 PULSE0_CNT (0x148)	705
40.7.36 PULSE0CYCLE_CNT (0x14C)	706
40.7.37 PULSE1_CNT (0x150)	706
40.7.38 PULSE1CYCLE_CNT (0x154)	707
40.7.39 PULSE0_SNAP0 (0x158)	707
40.7.40 PULSE0CYCLE_SNAP0 (0x15C)	707
40.7.41 PULSE0_SNAP1 (0x160)	708
40.7.42 PULSE0CYCLE_SNAP1 (0x164)	708
40.7.43 PULSE1_SNAP0 (0x168)	709
40.7.44 PULSE1CYCLE_SNAP0 (0x16C)	709
40.7.45 PULSE1_SNAP1 (0x170)	709
40.7.46 PULSE1CYCLE_SNAP1 (0x174)	710
40.7.47 TIMESTAMP (0x1E0)	710
40.7.48 ADC_THRESHOLD (0x1F0)	711
40.7.49 ADCX_CFG0 (0x200)	711
40.7.50 ADCX_CFG1 (0x204)	712
40.7.51 ADCX_CFG2 (0x208)	712
40.7.52 ADCX_VAL_SW (0x20C)	712
40.7.53 ADCY_CFG0 (0x210)	713
40.7.54 ADCY_CFG1 (0x214)	713
40.7.55 ADCY_CFG2 (0x218)	714
40.7.56 ADCY_VAL_SW (0x21C)	714
40.7.57 CAL_CFG (0x220)	714
40.7.58 ADC_INJECT_CTRL (0x224)	715

40.7.59	PHASE_PARAM (0x230)	715
40.7.60	POS_ADJ (0x234)	716
40.7.61	POS_THRESHOLD (0x238)	716
40.7.62	UVW_POS (0x240 + 0x4 * n)	716
40.7.63	UVW_POS_CFG (0x258 + 0x4 * n)	717
40.7.64	PHASE_CNT (0x280)	717
40.7.65	PHASE_UPDATE (0x284)	718
40.7.66	POSITION (0x288)	718
40.7.67	POSITION_UPDATE (0x28C)	718
40.7.68	ANGLE (0x290)	719
40.7.69	POS_TIMEOUT (0x294)	719
40.7.70	CALC_STATE (0x2A0)	719
40.7.71	TOGI_CFG0 (0x2C0)	720
40.7.72	TOGI_CFG1 (0x2C4)	720
40.7.73	SINP_ACC (0x2D0)	721
40.7.74	SINN_ACC (0x2D4)	721
40.7.75	COSP_ACC (0x2D8)	722
40.7.76	COSN_ACC (0x2DC)	722
41	Quadrature Encoder Output QEO	723
41.1	Feature Summary	723
41.2	Block Diagram	723
41.3	Functional Description	724
41.3.1	Position Preprocessing	724
41.3.2	Sine Wave Output	724
41.3.2.1	VD/VQ Control for Sine Waves	725
41.3.2.2	Saddle Wave Mode in Sine Wave Output	725
41.3.2.3	Sine Wave Scaling and Limiting	725
41.3.2.4	Sine Wave Dead Zone Compensation	725
41.3.3	ABZ Output	725
41.3.3.1	A/B Two-Phase Quadrature Output with Internal Position	725
41.3.3.2	High-Speed Pulse & Direction Output	726
41.3.3.3	Up & Down Output	726
41.3.3.4	A/B Two-Phase Quadrature Output Without Internal Position	726
41.3.3.5	Watchdog for Internal Position Manager	727
41.3.4	PWM FORCE Output	727
41.4	QEO Register List	727
41.5	QEO Register Details	730
41.5.1	WAVE[MODE] (0x0)	730
41.5.2	WAVE[RESOLUTION] (0x4)	732
41.5.3	WAVE[PHASE_SHIFT] (0x8 + 0x4 * m)	733
41.5.4	WAVE[VD_INJECT] (0x14)	733
41.5.5	WAVE[VQ_INJECT] (0x20)	733
41.5.6	WAVE[VD_VQ_LOAD] (0x2C)	734

41.5.7 WAVE[AMPLITUDE] (0x30 + 0x4 * m)	734
41.5.8 WAVE[MID_POINT] (0x3C + 0x4 * m)	734
41.5.9 WAVE[LIMIT0][MIN_LEVEL0] (0x48 + 0x8 * m)	735
41.5.10 WAVE[LIMIT0][MAX_LEVEL0] (0x4C + 0x8 * m)	735
41.5.11 WAVE[LIMIT1][MIN_LEVEL1] (0x60 + 0x8 * m)	736
41.5.12 WAVE[LIMIT1][MAX_LEVEL1] (0x64 + 0x8 * m)	736
41.5.13 WAVE[DEADZONE_SHIFT] (0x78 + 0x4 * m)	736
41.5.14 WAVE[PWM_CYCLE] (0x84)	737
41.5.15 WAVE[FILTER] (0x88)	737
41.5.16 ABZ[MODE] (0x100)	737
41.5.17 ABZ[RESOLUTION] (0x104)	739
41.5.18 ABZ[PHASE_SHIFT] (0x108 + 0x4 * m)	739
41.5.19 ABZ[LINE_WIDTH] (0x114)	740
41.5.20 ABZ[WDOG_WIDTH] (0x118)	740
41.5.21 ABZ[POSTION_SYNC] (0x11C)	740
41.5.22 ABZ[OVERALL_OFFSET] (0x120)	741
41.5.23 ABZ[Z_START] (0x124)	741
41.5.24 ABZ[Z_END] (0x128)	741
41.5.25 ABZ[Z_OFFSET] (0x12C)	742
41.5.26 ABZ[Z_PULSE_WIDTH] (0x130)	742
41.5.27 ABZ[FILTER] (0x134)	743
41.5.28 PWM[MODE] (0x140)	743
41.5.29 PWM[RESOLUTION] (0x144)	744
41.5.30 PWM[PHASE_SHIFT] (0x148 + 0x4 * m)	744
41.5.31 PWM[PHASE_TABLE] (0x158 + 0x4 * m)	745
41.5.32 PWM[FILTER] (0x1B8)	745
41.5.33 POSTION_SOFTWARE (0x1F8)	745
41.5.34 POSTION_SEL (0x1FC)	746
41.5.35 STATUS (0x200)	746
41.5.36 DEBUG0 (0x204)	747
41.5.37 DEBUG1 (0x208)	747
41.5.38 DEBUG2 (0x20C)	748
41.5.39 DEBUG3 (0x210)	748
41.5.40 DEBUG4 (0x214)	748
41.5.41 DEBUG5 (0x218)	749
42 Programmable Logic Block (PLB)	750
42.1 Feature Summary	750
42.2 Block Diagram	750
42.3 Functional Description	751
42.3.1 TYPE A Functional Description	751
42.3.2 TYPE B Functional Description	752
42.4 PLB Register List	753
42.5 PLB Register Details	757

42.5.1	TYPE_A[LOOKUP_TABLE] (0x0 + 0x20 * n + 0x4 * m)	757
42.5.2	TYPE_A[SW_INJECT] (0x10 + 0x20 * n)	757
42.5.3	TYPE_B[LUT] (0x400 + 0x80 * n + 0x4 * m)	757
42.5.4	TYPE_B[COMP] (0x408 + 0x80 * n + 0x4 * m)	758
42.5.5	TYPE_B[MODE] (0x418 + 0x80 * n)	758
42.5.6	TYPE_B[SW_INJECT] (0x41C + 0x80 * n)	759
42.5.7	TYPE_B[LUT_CMP] (0x420 + 0x80 * n + 0x4 * m)	759
43	Sigma-Delta Signal Reception Unit SDM	760
43.1	Overview	760
43.2	Features	760
43.3	Detailed Description	761
43.3.1	Manchester Mode	762
43.3.2	Sinc Filters	763
43.3.2.1	Sinc Filter Output Data Rate and Delay	763
43.3.3	First Filter Unit (Data Path)	763
43.3.3.1	32-bit or 16-bit Output Control	764
43.3.3.2	Data FIFO	765
43.3.3.3	Synchronization Event	766
43.3.3.4	Mask Control	766
43.3.3.5	Timestamp Output	767
43.3.4	Second Filter Unit (Amplitude Path)	767
43.3.4.1	Upper Threshold Comparator	768
43.3.4.2	Zero-Crossing Comparator	768
43.3.4.3	Lower Threshold Comparator	768
43.4	SDM Registers	768
43.4.1	Register Description	768
43.4.2	Register Details	770
43.4.3	CTRL (0x0)	770
43.4.4	INT_EN (0x4)	771
43.4.5	STATUS (0x8)	772
43.4.6	CH[SDFIFOCTRL] (0x10 + 0x40 * n)	772
43.4.7	CH[SDCTRLP] (0x14 + 0x40 * n)	773
43.4.8	CH[SDCTRLE] (0x18 + 0x40 * n)	774
43.4.9	CH[SDST] (0x1C + 0x40 * n)	775
43.4.10	CH[SDATA] (0x20 + 0x40 * n)	775
43.4.11	CH[SDFIFO] (0x24 + 0x40 * n)	776
43.4.12	CH[SCAMP] (0x28 + 0x40 * n)	776
43.4.13	CH[SCHTL] (0x2C + 0x40 * n)	776
43.4.14	CH[SCHTLZ] (0x30 + 0x40 * n)	777
43.4.15	CH[SCLLT] (0x34 + 0x40 * n)	777
43.4.16	CH[SCCTRL] (0x38 + 0x40 * n)	778
43.4.17	CH[SCST] (0x3C + 0x40 * n)	778
43.5	SDM Configuration and Usage Guide	779

43.5.1	Interrupt Handling	779
44	Timer Overview	780
44.1	General-Purpose Timers GPTMR, PTMR	780
44.2	Watchdog Timers WDG, PWDG	780
45	Timer TMR	781
45.1	Feature Summary	781
45.2	Function Description	781
45.2.1	Timer Time Base	781
45.2.2	output comparison	782
45.2.3	Input Capture	783
45.2.4	Single Mode	783
45.2.5	burst mode	784
45.2.6	External counting mode	784
45.2.7	Simple Encoder Mode	784
45.2.8	Interrupts and DMA	784
45.2.9	debug mode support	785
45.3	Timer Register	785
45.3.1	register description	785
45.3.2	CHANNEL[CR] (0x0 + 0x40 * n)	787
45.3.3	CHANNEL[CMR] (0x4 + 0x40 * n + 0x4 * m)	789
45.3.4	CHANNEL[RLD] (0xC + 0x40 * n)	789
45.3.5	CHANNEL[CNTR] (0x10 + 0x40 * n)	789
45.3.6	CHANNEL[BURST_CFG] (0x14 + 0x40 * n)	790
45.3.7	CHANNEL[BURST_COUNT] (0x18 + 0x40 * n)	790
45.3.8	CHANNEL[CAPPOS] (0x20 + 0x40 * n)	790
45.3.9	CHANNEL[CAPNEG] (0x24 + 0x40 * n)	791
45.3.10	CHANNEL[CAPPRD] (0x28 + 0x40 * n)	791
45.3.11	CHANNEL[CAPDLY] (0x2C + 0x40 * n)	791
45.3.12	CHANNEL[CNT] (0x30 + 0x40 * n)	792
45.3.13	SR (0x200)	792
45.3.14	IRQEN (0x204)	793
45.3.15	GCR (0x208)	794
46	Watchdog EWDG	795
46.1	Feature Summary	795
46.2	EWDG Architecture Chart	795
46.3	Function Description	795
46.3.1	Feature Overview	795
46.3.2	Clock selection	796
46.3.3	register configuration unlocking and parity requirements	796
46.3.4	Unlock dog feeding operation	797
46.3.4.1	Dog feeding protection enable	797
46.3.4.2	Unlock dog feeding operation	797

46.3.5 Window Mode	798
46.3.6 Working status in Debug mode	798
46.3.7 Operating status in low power mode	798
46.4 EWDG register list	798
46.5 EWDG register description	799
46.5.1 CTRL0 (0x0)	799
46.5.2 CTRL1 (0x4)	801
46.5.3 OT_INT_VAL (0x8)	802
46.5.4 OT_RST_VAL (0xC)	802
46.5.5 WDT_REFRESH_REG (0x10)	802
46.5.6 WDT_STATUS (0x14)	803
46.5.7 CFG_PROT (0x18)	804
46.5.8 REF_PROT (0x1C)	804
46.5.9 WDT_EN (0x20)	804
46.5.10 REF_TIME (0x24)	805
46.6 Watchdog configuration instructions	805
46.6.1 Configuration Reference: Window working mode, dog feeding protection enabled	805
46.6.2 Configuration Reference: Register Configuration Locking and Unlocking	806
47 Communication Peripherals Overview	807
47.1 Universal Asynchronous Receiver Transmitter (UART, PUART)	807
47.2 Serial Peripheral Interface (SPI)	807
47.3 Inter-Integrated Circuit (I2C)	807
47.4 Controller Area Network (MCAN)	807
47.5 Precision Time Protocol Controller (PTPC)	807
47.6 Ethernet Controller (ENET)	808
47.7 Universal Serial Bus (USB)	808
48 Universal Asynchronous Receiver Transmitter (UART)	809
48.1 Feature Summary	809
48.2 Functional Description	809
48.2.1 UART Transmission	809
48.2.2 UART Reception	810
48.2.3 Idle Detection	810
48.2.4 Baud Rate Control	810
48.2.5 Modem Controller	811
48.2.6 Loopback Mode	811
48.2.7 Interrupts	811
48.2.8 DMA	811
48.3 UART Register List	812
48.4 UART Register Details	813
48.4.1 IDLE_CFG (0x4)	813
48.4.2 ADDR_CFG (0x8)	813
48.4.3 IIR2 (0xC)	814

48.4.4	CFG (0x10)	815
48.4.5	OSCR (0x14)	816
48.4.6	FCRR (0x18)	816
48.4.7	MOTO_CFG (0x1C)	818
48.4.8	RBR (0x20)	819
48.4.9	THR (0x20)	819
48.4.10	DLL (0x20)	819
48.4.11	IER (0x24)	820
48.4.12	DLM (0x24)	821
48.4.13	IIR (0x28)	821
48.4.14	FCR (0x28)	822
48.4.15	LCR (0x2C)	823
48.4.16	MCR (0x30)	824
48.4.17	LSR (0x34)	824
48.4.18	MSR (0x38)	826
48.4.19	GPR (0x3C)	826
49	Serial Peripheral Interface (SPI) Bus	828
49.1	Module Functions	828
49.2	Functional Description	828
49.2.1	Master Mode	829
49.2.2	Slave Mode	829
49.2.3	DSPI Mode	830
49.2.4	QSPI Mode	830
49.2.5	Initiating Transmission and Command Values	830
49.2.6	FIFO Thresholds, Interrupts, and DMA	832
49.3	Brief SPI Configuration Instructions	832
49.3.1	Sending Command–Address–Data Fields Using DMA in Master Mode	832
49.3.2	Pure Data Send Without DMA in Master Mode	833
49.4	SPI Register List	833
49.5	SPI Register Description	834
49.5.1	WR_TRANS_CNT (0x4)	834
49.5.2	RD_TRANS_CNT (0x8)	834
49.5.3	TRANSFMT (0x10)	835
49.5.4	DIRECTIO (0x14)	836
49.5.5	TRANSCTRL (0x20)	837
49.5.6	CMD (0x24)	839
49.5.7	ADDR (0x28)	839
49.5.8	DATA (0x2C)	840
49.5.9	CTRL (0x30)	841
49.5.10	STATUS (0x34)	842
49.5.11	INTREN (0x38)	843
49.5.12	INTRST (0x3C)	844
49.5.13	TIMING (0x40)	845

49.5.14	SLVST (0x60)	845
49.5.15	SLVDATAcnt (0x64)	846
49.5.16	SLVDATAWCNT (0x68)	846
49.5.17	SLVDATArcnt (0x6C)	847
49.5.18	CONFIG (0x7C)	847
50	Integrated Circuit Bus I2C	849
50.1	Feature Summary	849
50.2	Function Description	849
50.2.1	Main functions	849
50.2.2	Timing Configuration	850
50.2.3	Master Mode	851
50.2.4	Slave mode	852
50.2.5	interrupt	853
50.2.6	DMA	853
50.3	I2C register	853
50.3.1	register description	853
50.3.2	register details	853
50.3.3	CFG (0x10)	854
50.3.4	INTEN (0x14)	854
50.3.5	STATUS (0x18)	855
50.3.6	ADDR (0x1C)	856
50.3.7	DATA (0x20)	857
50.3.8	CTRL (0x24)	857
50.3.9	CMD (0x28)	859
50.3.10	SETUP (0x2C)	860
50.3.11	TPM (0x30)	861
51	Controller Area Network MCAN	863
51.1	Overview	863
51.2	CAN Architecture Diagram	863
51.3	Pin Description	865
51.4	Functional Description	865
51.4.1	CAN General Description	865
51.4.2	Bit Time	866
51.4.3	Normal Operating Mode	867
51.4.4	CAN FD Operating Mode	867
51.4.5	Silent Mode	867
51.4.6	Standby Mode	867
51.4.7	External Loopback Mode	868
51.4.8	Internal Loopback Mode	868
51.4.9	Timestamp Generation	869
51.4.10	Receive Buffer Data Format	869
51.4.11	Transmit Buffer Data Format	871

51.4.12	Transmit Event Data Format	872
51.4.13	Standard Message ID Filter	874
51.4.14	Extended Message ID Filter	875
51.4.15	Receive FIFO	876
51.4.15.1	Receive FIFO Blocking Mode	876
51.4.15.2	Receive FIFO Overwrite Mode	876
51.4.16	Interrupt	877
51.4.17	Message Memory Schematic	877
51.5	CAN Configuration and Usage Instructions	877
51.5.1	Software Initialization	877
51.5.2	Receive Processing	878
51.5.3	Transmit Processing	878
51.5.3.1	Transmit Buffer	879
51.5.3.2	Transmit FIFO	879
51.5.3.3	Transmit Queue	879
51.5.3.4	Transmit Cancellation	879
51.6	MCAN Registers	880
51.6.1	MCAN Register Description	880
51.6.2	Register Detailed Information	882
51.6.3	ENDN (0x4)	882
51.6.4	DBTP (0xC)	882
51.6.5	TEST (0x10)	883
51.6.6	RWD (0x14)	884
51.6.7	CCCR (0x18)	885
51.6.8	NBTP (0x1C)	887
51.6.9	TSCC (0x20)	888
51.6.10	TSCV (0x24)	888
51.6.11	TOCC (0x28)	889
51.6.12	TOCV (0x2C)	890
51.6.13	ECR (0x40)	890
51.6.14	PSR (0x44)	891
51.6.15	TDCR (0x48)	895
51.6.16	IR (0x50)	895
51.6.17	IE (0x54)	899
51.6.18	ILS (0x58)	900
51.6.19	ILE (0x5C)	901
51.6.20	GFC (0x80)	901
51.6.21	SIDFC (0x84)	902
51.6.22	XIDFC (0x88)	903
51.6.23	XIDAM (0x90)	903
51.6.24	HPMS (0x94)	903
51.6.25	NDAT1 (0x98)	904
51.6.26	NDAT2 (0x9C)	905

51.6.27	RXF0C (0xA0)	905
51.6.28	RXF0S (0xA4)	906
51.6.29	RXF0A (0xA8)	907
51.6.30	RXBC (0xAC)	907
51.6.31	RXF1C (0xB0)	908
51.6.32	RXF1S (0xB4)	908
51.6.33	RXF1A (0xB8)	909
51.6.34	RXESC (0xBC)	910
51.6.35	TXBC (0xC0)	911
51.6.36	TXFQS (0xC4)	912
51.6.37	TXESC (0xC8)	913
51.6.38	TXBRP (0xCC)	913
51.6.39	TXBAR (0xD0)	914
51.6.40	TXBCR (0xD4)	915
51.6.41	TXBTO (0xD8)	916
51.6.42	TXBCF (0xDC)	916
51.6.43	TXBTIE (0xE0)	917
51.6.44	TXBCIE (0xE4)	917
51.6.45	TXEFC (0xF0)	918
51.6.46	TXEFS (0xF4)	918
51.6.47	TXEFA (0xF8)	919
51.6.48	TS_SEL (0x200 + 0x4 * n)	919
51.6.49	CREL (0x240)	920
51.6.50	TSCFG (0x244)	921
51.6.51	TSS1 (0x248)	921
51.6.52	TSS2 (0x24C)	922
51.6.53	ATB (0x250)	922
51.6.54	ATBH (0x254)	923
51.6.55	GLB_CTL (0x400)	923
51.6.56	GLB_STATUS (0x404)	924
52	Precise Time Protocol Module PTPC	925
52.1	Feature Summary	925
52.2	Function Description	925
52.2.1	Timestamp Module	925
52.2.2	Timestamp Capture and Compare	926
52.2.3	Timestamp Output Port	927
52.3	PTPC register list	927
52.4	PTPC register description	928
52.4.1	PTPC[CTRL0] (0x0 + 0x1000 * n)	928
52.4.2	PTPC[CTRL1] (0x4 + 0x1000 * n)	929
52.4.3	PTPC[TIMEH] (0x8 + 0x1000 * n)	929
52.4.4	PTPC[TIMEL] (0xC + 0x1000 * n)	929
52.4.5	PTPC[TS_UPDTH] (0x10 + 0x1000 * n)	930

52.4.6	PTPC[TS_UPDTL] (0x14 + 0x1000 * n)	930
52.4.7	PTPC[ADDEND] (0x18 + 0x1000 * n)	931
52.4.8	PTPC[TARH] (0x1C + 0x1000 * n)	931
52.4.9	PTPC[TARL] (0x20 + 0x1000 * n)	931
52.4.10	PTPC[PPS_CTRL] (0x2C + 0x1000 * n)	932
52.4.11	PTPC[CAPT_SNAPH] (0x30 + 0x1000 * n)	932
52.4.12	PTPC[CAPT_SNAPL] (0x34 + 0x1000 * n)	933
52.4.13	TIME_SEL (0x2000)	933
52.4.14	INT_STS (0x2004)	934
52.4.15	INT_EN (0x2008)	934
52.4.16	PTPC_CAN_TS_SEL (0x3000)	935
53	1-Wire Protocol Controller OWR	936
53.1	Feature Summary	936
53.2	External Interface Pin Description	936
53.3	Functional Description	936
53.3.1	Clock Division	936
53.3.2	Initialization	936
53.3.3	OWR Read/Write Timing	937
53.3.3.1	Write Timing	937
53.3.3.2	Read Timing	937
53.3.4	Reading/Writing Multiple Bytes via DMA	937
53.3.5	SRA (Search ROM Accelerate)	938
53.3.6	Interrupts	939
53.4	OWR Registers	939
53.4.1	Register Description	939
53.4.2	Register Details	939
53.4.3	CTRL (0x0)	939
53.4.4	DIV_CFG (0x4)	940
53.4.5	RESET (0x8)	940
53.4.6	CMD (0xC)	941
53.4.7	DATA (0x10)	941
53.4.8	RX_CNT (0x54)	942
53.4.9	GLB_CFG (0x58)	942
53.4.10	DMA_EN (0x60)	942
53.4.11	IRQ_STS (0x70)	943
53.4.12	IRQ_EN (0x74)	943
54	Universal Serial Bus USB	945
54.1	Function Introduction	945
54.2	pin description	945
54.3	Workflow	945
54.3.1	usbphy initialization	945
54.3.2	Configuration working mode	945

54.3.3	Host initialization process	946
54.3.4	Device initialization process	946
54.4	data structure	946
54.4.1	Host Data Structure	946
54.4.2	Device Data Structure	951
54.5	USB register list	955
54.6	USB register description	957
54.6.1	GPTIMER0LD (0x80)	957
54.6.2	GPTIMER0CTRL (0x84)	958
54.6.3	GPTIMER1LD (0x88)	958
54.6.4	GPTIMER1CTRL (0x8C)	959
54.6.5	SBUSCFG (0x90)	960
54.6.6	USBCMD (0x140)	961
54.6.7	USBSTS (0x144)	966
54.6.8	USBINTR (0x148)	971
54.6.9	FRINDEX (0x14C)	973
54.6.10	DEVICEADDR (0x154)	974
54.6.11	PERIODICLISTBASE (0x154)	975
54.6.12	ASYNCLISTADDR (0x158)	976
54.6.13	ENDPTLISTADDR (0x158)	976
54.6.14	BURSTSIZE (0x160)	977
54.6.15	TXFILLTUNING (0x164)	977
54.6.16	ENDPTNAK (0x178)	979
54.6.17	ENDPTNAKEN (0x17C)	980
54.6.18	PORTSC1 (0x184)	980
54.6.19	OTGSC (0x1A4)	989
54.6.20	USBMODE (0x1A8)	991
54.6.21	ENDPTSETUPSTAT (0x1AC)	993
54.6.22	ENDPTPRIME (0x1B0)	994
54.6.23	ENDPTFLUSH (0x1B4)	995
54.6.24	ENDPTSTAT (0x1B8)	996
54.6.25	ENDPTCOMPLETE (0x1BC)	997
54.6.26	ENDPTCTRL (0x1C0 + 0x4 * n)	998
54.6.27	OTG_CTRL0 (0x200)	1000
54.6.28	PHY_CTRL0 (0x210)	1001
54.6.29	PHY_CTRL1 (0x214)	1002
54.6.30	TOP_STATUS (0x220)	1003
54.6.31	PHY_STATUS (0x224)	1003
55	Ethernet Controller ENET	1005
55.1	Feature Summary	1005
55.2	ENET Module Structure	1005
55.3	MAC Data Packet Description	1005
55.3.1	Ethernet ii Ethernet Frame Structure	1005

55.3.1.1	Frame Format	1005
55.3.1.2	Description of Each Field	1006
55.3.2	IEEE 802.3 Type	1007
55.3.2.1	Frame Format	1007
55.3.2.2	Description of Each Field	1007
55.4	ENET Module Port Signal Description	1007
55.4.1	MII Interface	1007
55.4.2	RMII Interface	1008
55.4.3	RGMII Interface	1009
55.4.4	MDIO Interface Introduction	1009
55.5	ENET Functional Description	1010
55.5.1	Built-in DMA Controller Features	1010
55.5.1.1	Burst Access	1010
55.5.1.2	Error Access	1011
55.5.2	DMA Controller Receive Function	1011
55.5.2.1	DMA Receive Error	1011
55.5.2.2	DMA Receive CRC Handling	1012
55.5.2.3	DMA Receive Process Suspend	1012
55.5.3	DMA Controller Transmit Function	1012
55.5.3.1	DMA Transmit Frame Processing	1013
55.5.4	MAC Control Transmit Function	1013
55.5.4.1	TX_FIFO Flush	1014
55.5.4.2	Collision Transmission	1015
55.5.4.3	Transmit Frame Processing: CRC and PAD Generation	1015
55.5.4.4	Source MAC Address Processing	1015
55.5.4.5	VLAN Insert, Replace, and Delete Processing	1015
55.5.5	MAC Control Receive Function	1016
55.5.5.1	Receive Frame Processing	1016
55.5.6	MAC Controller Frame Filtering	1016
55.5.6.1	Destination MAC Address Exact Filtering	1016
55.5.6.2	DMAC Address Filtering	1017
55.5.6.3	Filtering of TAG Identifier in Ethernet Packets with VLAN Flag	1017
55.5.7	PTP Sub-Controller Functional Description	1017
55.5.7.1	Reference Clock Source	1018
55.5.7.2	PTP Controller Message Types	1018
55.5.7.3	Receive Characteristics	1019
55.5.7.4	Transmit Characteristics	1019
55.5.7.5	System Time Calibration	1019
55.5.8	ENET Interrupt Description	1021
55.5.9	ENET Configuration Steps Description	1021
55.5.9.1	MII Interface Transmission	1021
55.5.9.2	MII Interface Reception	1022
55.5.10	Descriptor Details	1022

55.5.10.1 Enhanced Transmit Descriptor	1022
55.5.10.2 Enhanced Receive Descriptor	1026
55.6 ENET Register List	1031
55.7 ENET Register Description	1036
55.7.1 MACCFG (0x0)	1036
55.7.2 MACFF (0x4)	1043
55.7.3 HASH_H (0x8)	1046
55.7.4 HASH_L (0xC)	1047
55.7.5 GMII_ADDR (0x10)	1047
55.7.6 GMII_DATA (0x14)	1049
55.7.7 FLOWCTRL (0x18)	1050
55.7.8 VLAN_TAG (0x1C)	1052
55.7.9 RWKFRMFILT (0x28)	1054
55.7.10 PMT_CSR (0x2C)	1055
55.7.11 LPI_CSR (0x30)	1056
55.7.12 LPI_TCR (0x34)	1058
55.7.13 INTR_STATUS (0x38)	1059
55.7.14 INTR_MASK (0x3C)	1061
55.7.15 MAC_ADDR_0_HIGH (0x40)	1062
55.7.16 MAC_ADDR_0_LOW (0x44)	1062
55.7.17 MAC_ADDR[HIGH] (0x48 + 0x8 * n)	1063
55.7.18 MAC_ADDR[LOW] (0x4C + 0x8 * n)	1064
55.7.19 XMII_CSR (0xD8)	1064
55.7.20 WDOG_WTO (0xDC)	1065
55.7.21 MMC_CNTRL (0x100)	1066
55.7.22 MMC_INTR_RX (0x104)	1068
55.7.23 MMC_INTR_TX (0x108)	1070
55.7.24 MMC_INTR_MASK_RX (0x10C)	1073
55.7.25 MMC_INTR_MASK_TX (0x110)	1076
55.7.26 TX64OCTETS_GB (0x124)	1079
55.7.27 TX65TO127OCTETS_GB (0x128)	1080
55.7.28 TX128TO255OCTETS_GB (0x12C)	1080
55.7.29 TX256TO511OCTETS_GB (0x130)	1080
55.7.30 TX512TO1023OCTETS_GB (0x134)	1081
55.7.31 TX1024TOMAXOCTETS_GB (0x138)	1081
55.7.32 RXFRAMECOUNT_GB (0x180)	1082
55.7.33 MMC_IPC_INTR_MASK_RX (0x200)	1082
55.7.34 MMC_IPC_INTR_RX (0x208)	1082
55.7.35 RXIPV4_GD_FMS (0x210)	1083
55.7.36 L3_L4_CFG[L3_L4_CTRL] (0x400 + 0x20 * n)	1083
55.7.37 L3_L4_CFG[L4_ADDR] (0x404 + 0x20 * n)	1086
55.7.38 L3_L4_CFG[L3_ADDR_0] (0x410 + 0x20 * n)	1086
55.7.39 L3_L4_CFG[L3_ADDR_1] (0x414 + 0x20 * n)	1087

55.7.40 L3_L4_CFG[L3_ADDR_2] (0x418 + 0x20 * n)	1088
55.7.41 L3_L4_CFG[L3_ADDR_3] (0x41C + 0x20 * n)	1088
55.7.42 VLAN_TAG_INC_RPL (0x584)	1089
55.7.43 VLAN_HASH (0x588)	1090
55.7.44 TS_CTRL (0x700)	1091
55.7.45 SUB_SEC_INCR (0x704)	1094
55.7.46 SYST_SEC (0x708)	1095
55.7.47 SYST_NSEC (0x70C)	1095
55.7.48 SYST_SEC_UPD (0x710)	1096
55.7.49 SYST_NSEC_UPD (0x714)	1096
55.7.50 TS_ADDEND (0x718)	1097
55.7.51 TGTMM_SEC (0x71C)	1097
55.7.52 TGTMM_NSEC (0x720)	1098
55.7.53 SYSTM_H_SEC (0x724)	1099
55.7.54 TS_STATUS (0x728)	1099
55.7.55 PPS_CTRL (0x72C)	1101
55.7.56 AUX_TS_NSEC (0x730)	1104
55.7.57 AUX_TS_SEC (0x734)	1104
55.7.58 PPS0_INTERVAL (0x760)	1104
55.7.59 PPS0_WIDTH (0x764)	1105
55.7.60 PPS[TGTMM_SEC] (0x780 + 0x20 * n)	1106
55.7.61 PPS[TGTMM_NSEC] (0x784 + 0x20 * n)	1106
55.7.62 PPS[INTERVAL] (0x788 + 0x20 * n)	1107
55.7.63 PPS[WIDTH] (0x78C + 0x20 * n)	1108
55.7.64 DMA_BUS_MODE (0x1000)	1108
55.7.65 DMA_TX_POLL_DEMAND (0x1004)	1113
55.7.66 DMA_RX_POLL_DEMAND (0x1008)	1114
55.7.67 DMA_RX_DESC_LIST_ADDR (0x100C)	1114
55.7.68 DMA_TX_DESC_LIST_ADDR (0x1010)	1114
55.7.69 DMA_STATUS (0x1014)	1115
55.7.70 DMA_OP_MODE (0x1018)	1122
55.7.71 DMA_INTR_EN (0x101C)	1129
55.7.72 DMA_MISS_OVF_CNT (0x1020)	1132
55.7.73 DMA_RX_INTR_WDOG (0x1024)	1132
55.7.74 DMA_AXI_MODE (0x1028)	1133
55.7.75 DMA_BUS_STATUS (0x102C)	1135
55.7.76 DMA_CURR_HOST_TX_DESC (0x1048)	1136
55.7.77 DMA_CURR_HOST_RX_DESC (0x104C)	1136
55.7.78 DMA_CURR_HOST_TX_BUF (0x1050)	1137
55.7.79 DMA_CURR_HOST_RX_BUF (0x1054)	1137
55.7.80 CTRL0 (0x3000)	1138
55.7.81 CTRL2 (0x3008)	1138

56 Industrial Network Overview	.1140
56.1 EtherCat Slave Controller ESC	.1140
57 EtherCAT slave controller ESC	.1141
57.1 Overview	.1141
57.2 Function Description	.1141
57.2.1 EtherCAT Physical Layer Interface	.1141
57.2.2 EEPROM	.1141
57.2.3 Clock	.1141
57.2.4 Reset	.1142
57.2.5 LED and LINK status	.1143
57.2.6 IO Multipurpose Selector CTR MUX	.1144
57.2.7 SYNC and LATCH	.1144
57.2.8 Interrupts and DMA	.1145
57.2.9 GPIO	.1146
57.3 ESC register	.1146
57.4 ESC register details	.1154
57.4.1 TYPE (0x0)	.1154
57.4.2 REVISION (0x1)	.1155
57.4.3 BUILD (0x2)	.1155
57.4.4 FMMU_NUM (0x4)	.1155
57.4.5 SYNCM_NUM (0x5)	.1156
57.4.6 RAM_SIZE (0x6)	.1156
57.4.7 PORT_DESC (0x7)	.1156
57.4.8 FEATURE (0x8)	.1157
57.4.9 STATION_ADDR (0x10)	.1158
57.4.10 STATION_ALS (0x12)	.1158
57.4.11 REG_WEN (0x20)	.1159
57.4.12 REG_WP (0x21)	.1159
57.4.13 ESC_WEN (0x30)	.1160
57.4.14 ESC_WP (0x31)	.1160
57.4.15 ESC_RST_ECOT (0x40)	.1161
57.4.16 ESC_RST_PDI (0x41)	.1161
57.4.17 ESC_DL_CTRL (0x100)	.1162
57.4.18 PHYSICAL_RW_OFFSET (0x108)	.1164
57.4.19 ESC_DL_STAT (0x110)	.1165
57.4.20 AL_CTRL (0x120)	.1167
57.4.21 AL_STAT (0x130)	.1167
57.4.22 AL_STAT_CODE (0x134)	.1168
57.4.23 RUN_LED_OVRD (0x138)	.1168
57.4.24 ERR_LED_OVRD (0x139)	.1169
57.4.25 PDI_CTRL (0x140)	.1170
57.4.26 ESC_CFG (0x141)	.1170
57.4.27 PDI_INFO (0x14E)	.1171

57.4.28 PDI_CFG (0x150)	1172
57.4.29 PDI_SL_CFG (0x151)	1173
57.4.30 PDI_EXT_CFG (0x152)	1174
57.4.31 ECAT_EVT_MSK (0x200)	1174
57.4.32 PDI_AL_EVT_MSK (0x204)	1175
57.4.33 ECAT_EVT_REQ (0x210)	1175
57.4.34 AL_EVT_REQ (0x220)	1176
57.4.35 RX_ERR_CNT (0x300 + 0x2 * n)	1178
57.4.36 FWD_RX_ERR_CNT (0x308 + 0x1 * n)	1178
57.4.37 ECAT_PU_ERR_CNT (0x30C)	1179
57.4.38 PDI_ERR_CNT (0x30D)	1179
57.4.39 LOST_LINK_CNT (0x310 + 0x1 * n)	1179
57.4.40 WDG_DIV (0x400)	1180
57.4.41 WDG_TIME_PDI (0x410)	1180
57.4.42 WDG_TIME_PDAT (0x420)	1181
57.4.43 WDG_STAT_PDAT (0x440)	1181
57.4.44 WDG_CNT_PDAT (0x442)	1182
57.4.45 WDG_CNT_PDI (0x443)	1182
57.4.46 EEPROM_CFG (0x500)	1182
57.4.47 EEPROM_PDI_ACC_STAT (0x501)	1183
57.4.48 EEPROM_CTRL_STAT (0x502)	1183
57.4.49 EEPROM_ADDR (0x504)	1185
57.4.50 EEPROM_DATA (0x508)	1185
57.4.51 MII_MNG_CS (0x510)	1186
57.4.52 PHY_ADDR (0x512)	1187
57.4.53 PHY_REG_ADDR (0x513)	1188
57.4.54 PHY_DATA (0x514)	1189
57.4.55 MIIM_ECAT_ACC_STAT (0x516)	1189
57.4.56 MIIM_PDI_ACC_STAT (0x517)	1189
57.4.57 PHY_STAT (0x518 + 0x1 * n)	1190
57.4.58 FMMU[LOGIC_START_ADDR] (0x600 + 0x10 * n)	1191
57.4.59 FMMU[LENGTH] (0x604 + 0x10 * n)	1191
57.4.60 FMMU[LOGIC_START_BIT] (0x606 + 0x10 * n)	1191
57.4.61 FMMU[LOGIC_STOP_BIT] (0x607 + 0x10 * n)	1192
57.4.62 FMMU[PHYSICAL_START_ADDR] (0x608 + 0x10 * n)	1192
57.4.63 FMMU[PHYSICAL_START_BIT] (0x60A + 0x10 * n)	1193
57.4.64 FMMU[TYPE] (0x60B + 0x10 * n)	1193
57.4.65 FMMU[ACTIVATE] (0x60C + 0x10 * n)	1193
57.4.66 SYNCM[PHYSICAL_START_ADDR] (0x800 + 0x8 * n)	1194
57.4.67 SYNCM[LENGTH] (0x802 + 0x8 * n)	1194
57.4.68 SYNCM[CONTROL] (0x804 + 0x8 * n)	1195
57.4.69 SYNCM[STATUS] (0x805 + 0x8 * n)	1196
57.4.70 SYNCM[ACTIVATE] (0x806 + 0x8 * n)	1197

57.4.71 SYNCM[PDI_CTRL] (0x807 + 0x8 * n)	1197
57.4.72 RCV_TIME (0x900 + 0x4 * n)	1198
57.4.73 SYS_TIME (0x910)	1199
57.4.74 RCVT_ECAT_PU (0x918)	1200
57.4.75 SYS_TIME_OFFSET (0x920)	1200
57.4.76 SYS_TIME_DELAY (0x928)	1201
57.4.77 SYS_TIME_DIFF (0x92C)	1201
57.4.78 SPD_CNT_START (0x930)	1202
57.4.79 SPD_CNT_DIFF (0x932)	1202
57.4.80 SYS_TIME_DIFF_FD (0x934)	1203
57.4.81 SPD_CNT_FD (0x935)	1203
57.4.82 RCV_TIME_LM (0x936)	1204
57.4.83 CYC_UNIT_CTRL (0x980)	1204
57.4.84 SYNC0_ACT (0x981)	1205
57.4.85 PULSE_LEN (0x982)	1206
57.4.86 ACT_STAT (0x984)	1207
57.4.87 SYNC0_STAT (0x98E)	1207
57.4.88 SYNC1_STAT (0x98F)	1208
57.4.89 START_TIME_CO (0x990)	1208
57.4.90 NXT_SYNC1_PULSE (0x998)	1209
57.4.91 SYNC0_CYC_TIME (0x9A0)	1209
57.4.92 SYNC1_CYC_TIME (0x9A4)	1210
57.4.93 LATCH0_CTRL (0x9A8)	1210
57.4.94 LATCH1_CTRL (0x9A9)	1211
57.4.95 LATCH0_STAT (0x9AE)	1211
57.4.96 LATCH1_STAT (0x9AF)	1212
57.4.97 LATCH0_TIME_PE (0x9B0)	1213
57.4.98 LATCH0_TIME_NE (0x9B8)	1213
57.4.99 LATCH1_TIME_PE (0x9C0)	1214
57.4.100 LATCH1_TIME_NE (0x9C8)	1214
57.4.101 ECAT_BUF_CET (0x9F0)	1215
57.4.102 PDI_BUF_SET (0x9F8)	1215
57.4.103 PDI_BUF_CET (0x9FC)	1216
57.4.104 PID (0xE00)	1216
57.4.105 VID (0xE08)	1217
57.4.106 DIO_OUT_DATA (0xF00)	1217
57.4.107 GPO (0xF10)	1217
57.4.108 GPI (0xF18)	1218
57.4.109 USER_RAM_BYTE0 (0xF80)	1218
57.4.110 USER_RAM_BYTE1 (0xF81)	1219
57.4.111 USER_RAM_BYTE2 (0xF82)	1219
57.4.112 USER_RAM_BYTE3 (0xF83)	1220
57.4.113 USER_RAM_BYTE4 (0xF84)	1220

57.4.114	USER_RAM_BYTE5 (0xF85)	1221
57.4.115	USER_RAM_BYTE6 (0xF86)	1221
57.4.116	USER_RAM_BYTE7 (0xF87)	1221
57.4.117	USER_RAM_BYTE8 (0xF88)	1222
57.4.118	USER_RAM_BYTE9 (0xF89)	1222
57.4.119	USER_RAM_BYTE10 (0xF8A)	1223
57.4.120	USER_RAM_BYTE11 (0xF8B)	1223
57.4.121	USER_RAM_BYTE14 (0xF8E)	1223
57.4.122	USER_RAM_BYTE15 (0xF8F)	1224
57.4.123	USER_RAM_BYTE19 (0xF93)	1224
57.4.124	PDRAM (0x1000)	1225
57.4.125	PDRAM_ALS (0x10000)	1225
57.4.126	GPR_CFG0 (0x1F000)	1225
57.4.127	GPR_CFG1 (0x1F004)	1226
57.4.128	GPR_CFG2 (0x1F008)	1227
57.4.129	PHY_CFG0 (0x1F010)	1227
57.4.130	PHY_CFG1 (0x1F014)	1228
57.4.131	GPIO_CTRL (0x1F020)	1229
57.4.132	GPI_TRIG_CFG (0x1F024)	1230
57.4.133	GPO_TRIG_CFG (0x1F028)	1230
57.4.134	GPI_OVERRIDE0 (0x1F030)	1231
57.4.135	GPI_OVERRIDE1 (0x1F034)	1231
57.4.136	GPO_REG0 (0x1F038)	1232
57.4.137	GPO_REG1 (0x1F03C)	1232
57.4.138	GPI_REG0 (0x1F040)	1232
57.4.139	GPI_REG1 (0x1F044)	1233
57.4.140	GPR_STATUS (0x1F060)	1233
57.4.141	IO_CFG (0x1F080 + 0x4 * n)	1234
58	Ethernet Physical Layer ENET_PHY	1235
58.1	Overview	1235
58.2	Pinout	1235
58.2.1	External Pin	1235
58.2.2	internal pins	1236
58.3	register description	1237
58.3.1	BMCR (0x0)	1238
58.3.2	BMSR (0x1)	1238
58.3.3	FMLR (0x28)	1239
59	Analog Peripherals Overview and Product-Specific Information	1240
59.1	16-Bit Analog-to-Digital Converter ADC16	1240
59.1.1	ADC Clock	1240
59.1.2	ADC Input Channel Assignment	1240
59.1.3	ADC Conversion Trigger Signal Connections	1240

59.1.4	ADC Data Forwarding Connections	1240
59.1.5	ADC Differential Mode	1241
59.2	Analog Comparator Controller ACMP	1241
59.3	Temperature Sensor	1241
60	16-bit analog-to-digital converter ADC16	1242
60.1	Feature Summary	1242
60.2	Function Description	1242
60.2.1	ADC basic configuration	1243
60.2.1.1	ADC clock	1243
60.2.1.2	ADC initialization	1244
60.2.1.3	ADC sampling and conversion configuration	1244
60.2.1.4	AD conversion result calculation	1245
60.2.2	ADC working mode	1245
60.2.2.1	Read conversion mode	1245
60.2.2.2	periodic conversion mode	1246
60.2.2.3	Sequence Conversion Mode	1246
60.2.2.4	Preemption conversion mode	1248
60.2.3	ADC differential mode	1253
60.2.4	Direct output of AD conversion results	1254
60.2.5	ADC interrupt	1254
60.3	ADC16 register list	1255
60.4	ADC16 register description	1259
60.4.1	CONFIG (0x0 + 0x4 * n)	1259
60.4.2	TRG_DMA_ADDR (0x30)	1260
60.4.3	TRG_SW_STA (0x34)	1260
60.4.4	BUS_RESULT (0x400 + 0x4 * n)	1261
60.4.5	BUF_CFG0 (0x500)	1261
60.4.6	SEQ_CFG0 (0x800)	1262
60.4.7	SEQ_DMA_ADDR (0x804)	1262
60.4.8	SEQ_WR_ADDR (0x808)	1263
60.4.9	SEQ_DMA_CFG (0x80C)	1263
60.4.10	SEQ_QUE (0x810 + 0x4 * n)	1264
60.4.11	SEQ_HIGH_CFG (0x850)	1264
60.4.12	PRD_CFG[PRD_CFG] (0xC00 + 0x10 * n)	1265
60.4.13	PRD_CFG[PRD_THSHD_CFG] (0xC04 + 0x10 * n)	1265
60.4.14	PRD_CFG[PRD_RESULT] (0xC08 + 0x10 * n)	1265
60.4.15	SAMPLE_CFG (0x1000 + 0x4 * n)	1266
60.4.16	CONV_CFG0 (0x1100)	1266
60.4.17	CONV_CFG1 (0x1104)	1267
60.4.18	ADC_CFG0 (0x1108)	1268
60.4.19	INT_STS (0x1110)	1269
60.4.20	INT_EN (0x1114)	1269
60.4.21	TRIGMUX_EN (0x1118)	1270

60.4.22	TRG_CMPT_FLAG (0x111C)	1271
60.4.23	ANA_CTRL0 (0x1200)	1271
60.4.24	ANA_STATUS (0x1210)	1272
60.4.25	ADC16_PARAMS (0x1400 + 0x2 * n)	1272
60.4.26	ADC16_CONFIG0 (0x1444)	1272
60.4.27	ADC16_CONFIG1 (0x1460)	1273
61	Analog Comparison Controller ACMP	1274
61.1	Feature Summary	1274
61.2	Function Description	1274
61.2.1	Analog comparator basic configuration	1275
61.2.2	Processing of comparison results	1275
61.2.2.1	Comparison result polarity control	1275
61.2.2.2	window comparison mode	1275
61.2.2.3	Digital filtering	1276
61.2.2.4	Interrupt, DMA and comparison result output	1276
61.2.3	DAC configuration	1276
61.2.4	Comparator operating mode	1277
61.3	ACMP register list	1277
61.4	ACMP register description	1277
61.4.1	CHANNEL[CFG] (0x0 + 0x20 * n)	1277
61.4.2	CHANNEL[DACCFG] (0x4 + 0x20 * n)	1279
61.4.3	CHANNEL[SR] (0x10 + 0x20 * n)	1279
61.4.4	CHANNEL[IRQEN] (0x14 + 0x20 * n)	1279
61.4.5	CHANNEL[DMAEN] (0x18 + 0x20 * n)	1280
62	Temperature Sensor TSNS	1281
62.1	Feature Summary	1281
62.2	Function Description	1281
62.2.1	Temperature acquisition mode	1281
62.2.2	Temperature comparison function	1281
62.3	TSNS register	1282
62.4	TSNS register details	1282
62.4.1	T (0x0)	1282
62.4.2	TMAX (0x4)	1283
62.4.3	TMIN (0x8)	1283
62.4.4	AGE (0xC)	1283
62.4.5	STATUS (0x10)	1284
62.4.6	CONFIG (0x14)	1284
62.4.7	VALIDITY (0x18)	1285
62.4.8	FLAG (0x1C)	1286
62.4.9	UPPER_LIM_IRQ (0x20)	1286
62.4.10	LOWER_LIM_IRQ (0x24)	1286
62.4.11	UPPER_LIM_RST (0x28)	1287

62.4.12	LOWER_LIM_RST (0x2C)	1287
62.4.13	ASYNC (0x30)	1287
62.4.14	ADVAN (0x38)	1288
63	Information Security Module Overview	1289
63.1	Online decryption engine EXIP	1289
63.2	One-time programmable storage OTP	1289
63.3	BOOT ROM	1289
64	Execution In Place Encryption Engine (EXIP)	1291
64.1	Feature Summary	1291
64.2	Functional Description	1291
64.2.1	EXIP Segments, Keys, and Counters	1291
64.2.2	EXIP Key Wrapping and Unwrapping	1292
64.3	Appendix	1294
64.3.1	RFC3394 Overview	1294
65	System Debugging Overview	1296
66	Debugging Transport Module DTM	1297
66.1	DTM command	1297
66.2	DTM command description	1297
66.2.1	USERCODE (0x0)	1297
66.2.2	IDCODE (0x1)	1297
66.2.3	CLAMP (0x8)	1297
66.2.4	EXTEST (0x9)	1298
66.2.5	SAMPLE/PRELOAD (0xA)	1298
66.2.6	HIGHZ (0xB)	1298
66.2.7	AUTHEN (0xF)	1298
66.2.8	DTMCS (0x10)	1298
66.2.9	DMI (0x11)	1299
66.2.10	BYPASS (0x1F)	1300
67	Debugging Module DM	1301
67.1	System memory mapping	1301
67.2	DMI memory mapping	1301
67.3	DM register description	1302
67.3.1	DATA (0x4 + 0x1 * n)	1302
67.3.2	DMCONTROL (0x10)	1302
67.3.3	DMSTATUS (0x11)	1303
67.3.4	HARTINFO (0x12)	1304
67.3.5	HALTSUM (0x13)	1305
67.3.6	HAWINDOWSEL (0x14)	1305
67.3.7	HAWINDOW (0x15)	1306
67.3.8	ABSTRACTCS (0x16)	1306
67.3.9	COMMAND (0x17)	1307
67.3.10	ABSTRACTAUTO (0x18)	1307

67.3.11 PROGBUF (0x20 + 0x1 * n) 1308

67.3.12 SBCS (0x38) 1308

67.3.13 SBADDRESS (0x39 + 0x1 * n) 1310

67.3.14 SBDATA (0x3C + 0x1 * n) 1310

68 Version Information 1311

69 Disclaimer 1312

1 Product Overview

1.1 System Diagram

The system block diagram of this product is [Figure 1](#).

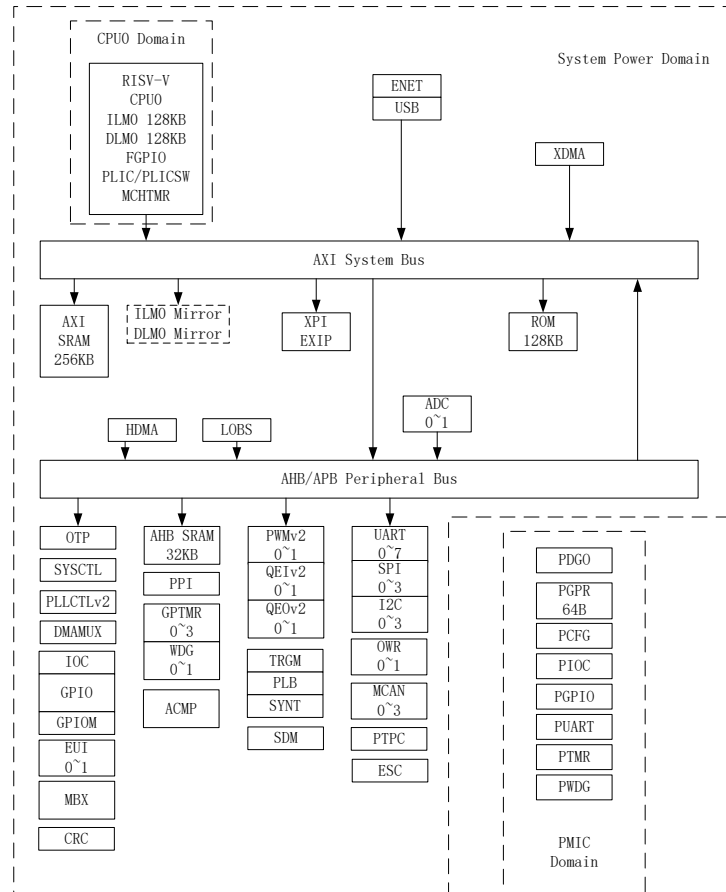


Figure 1: System Architecture Diagram

[Table 1](#) summary [Figure 1](#) all description of abbreviation.

Abbreviation	Description
CPU0 Power Domain	Subsystem that contains RISC-V CPU0, its local memory and private peripherals
HART	Hardware Thread, the RISC-V specification defines a framework that can include a complete RISC-V architecture , and the unit that can independently execute instructions is HART. In this manual, HART is equivalent to RISC-V core.
ILM	Instruction Local Memory
DLM	Data Local Memory
FGPI0	Fast General Purpose Input Output
ENET	Ethernet Controller (Ethernet)
USB	Universal Serial Bus

Abbreviation	Description
XDMA	AXI System Bus DMA Controller (AXI DMA)
HDMA	AHB Peripheral Bus DMA Controller (AHB DMA)
AXI SRAM	AXI Bus SRAM
AHB SRAM	AHB bus SRAM
XPI	Serial Bus Controller
PPI	Programmable Parallel Bus
EXIP	Encrypted Execution-In-Place
ADC	Analog-to-Digital Converter
SYSCTL	System Control Module
PLLCTL	PLL Controller (PLL Controller)
ACMP	Analog Comparator Controller
MBX	Mailbox
DMAMUX	DMA Request Router
TFA	Transcendental Function Accelerator
CRC	Cyclic Redundancy Check
IOC	IO Controller (Input Output Controller)
PIOC	Power Management Domain IO Controller
GPIO	General Purpose Input Output Controller
PGPIO	Power Management Domain GPIO Controller
GPIOM	GPIO Manager
EUI	Extended User Interface
LOBS	Logic Observer
OTP	One Time Program
PWMV2	PWM Timer (Pulse Width Modulation)
QEIV2	Quadrature Encoder Input
QEOV2	Quadrature Encoder Output
PLB	Programmable Logic Block
SDM	$\Sigma\Delta$ Signal Receiver Unit ($\Sigma\Delta$ Modulator)
TRGM	Trigger Manager
SYNT	Sync Timer
GPTMR	General Purpose Timer
PTMR	Universal Timer in Power Management Domain
WDG	Watchdog
PWDG	Power Management Domain Watchdog
ESC	EtherCAT Slave Controller
UART	Universal Asynchronous Receiver and Transmitter
PUART	Universal Asynchronous Transceiver in Power Management Domain
SPI	Serial Peripheral Interface
I2C	Inter-Integrated Circuit Bus
OWR	Single Bus Protocol Controller (1-Wire Controller)

Abbreviation	Description
CAN	Controller local area network (Control Area Network)
PTPC	Precise Time Protocol
RNG	Random Number Generator
KEYM	Key Manager
PGPR	General purpose register for power management domain
PCFG	Power Management Domain Configuration Module
System Power Domain	In this manual, the system power domain specifically refers to the logic and memory circuits powered by VDD_SOC
Power Management Domain	In this manual, power management domain refers specifically to the logic and storage circuits powered by VPMC

Table 1: Summary of Peripheral Abbreviations

1.2 Feature Summary

This chapter describes the main features of this product.

1.2.1 Kernel and System

32-bit RISC-V processor with the following processor characteristics:

- RV32-IMAFDCBP instruction set
 - integer instruction set
 - Multiplication Instruction Set
 - Atomic instruction set
 - Single precision floating point instruction set
 - Double precision floating point instruction set
 - compressed instruction set
 - -bit operation extended instruction set
 - DSP unit, supports SIMD and DSP instructions, is compatible with the RV32-P extended instruction set
- performance up to 5.6 CoreMark / MHz
- Privileged mode supports Machine mode, Supervisor mode and User mode
- supports 16 Physical Memory Protection PMP areas
- supports 32KB L1 instruction cache and 32KB L1 data cache
- supports 128 KB instruction local memory ILM and 128 KB data local memory DLM

processor is equipped with a platform interrupt controller PLIC to manage external interrupts for RISC-V

- supports multiple interrupt sources
- supports 8-level programmable interrupt priority
- Interrupt nesting extension and interrupt vector extension

processor core is equipped with a software interrupt controller PLICSW to manage RISC-V software inter-

rupts

- Generating RISC-V software interrupt

processor core is equipped with a machine timer MCHTMR, which manages RISC-V timer interrupts

- Generating RISC-V timer interrupt

DMA controller:

- XDMA, which supports 32 channels, is used for high-bandwidth data movement between memories, It can also be used for data transfer between peripheral registers and memory, or between peripheral registers.
- HDMA, which supports 32 channels, is used for low-latency data transfers between external registers and memory, Can also be used for data transfer between memories
- supports DMA request routing and allocation to any DMA controller

includes a mailbox MBX that supports communication between different processes in the processor:

- supports independent information transceiver interfaces
- supports interrupt generation

1.2.2 Internal memory

Internal memory includes:

- 544 KB on-chip SRAM
 - ILM0, instruction local memory for RISC-V CPU0, 128KB
 - DLM0, data local memory for RISC-V CPU0, 128KB
 - AXI SRAM0, 256KB, high-speed on-chip SRAM
 - AHB SRAM, 32KB, for low-latency access to HDMA
- General purpose register
 - Power management domain general registers PGPR0 and PGPRP1, each with a capacity of 64 bytes, can save data when the system power domain is powered down
- Internal read-only memory ROM, with a capacity of 128KB, the ROM stores the product's boot code, Flash loader and some peripheral drivers
- One-time programmable memory OTP, 4096 bits, can be used to store some of the chip's factory information, user keys, security configuration, boot configuration and other data

1.2.3 Power Management

This product integrates a complete power management system:

- Multiple on-chip power supplies
 - DCDC voltage converter, providing 0.9 ~ 1.3V output, supplying power to circuits in the system power domain, can adjust the DCDC output to support dynamic voltage frequency adjustment DVFS
 - LDOPMC, a typical 1.1V output linear regulator that powers circuits in the power management domain
 - LDOOTP, a linear regulator with a typical output of 2.5V, can supply power to OTP and can only be turned on when programming OTP.

- Run mode and low power mode: wait mode, stop mode, sleep mode, and shutdown mode
- chip integrated power-on reset circuit
- chip integrated low voltage detection circuit

1.2.4 Clock

The clock management system of this product supports multiple clock sources and low-power clock management:

- External clock source:
 - 24MHz on-chip oscillator, OSC24M, supports 24MHz crystal, and also supports external input of 24MHz active clock through pins. The 24MHz external high-speed oscillator is the default clock source for each PLL on the chip
- Internal clock source:
 - Internal RC oscillator, RC24M, frequency 24MHz, allows configuring the internal RC oscillator as a candidate clock source for the PLL
 - Internal 32KHz RC oscillator, RC32 K, serves as a candidate clock source for devices such as RTC
- 3 phase-locked loop PLLs, support fractional division, support spread spectrum
- supports low power management and automatic clock gating

1.2.5 Reset

Global reset can reset the entire chip, including the power management domain and the system power domain. The reset source is:

- RSTN pin reset

Power reset, also known as power management domain reset, can reset power management domain and system power domain. The reset source is:

- Power-on reset of the VPMC pin (VPMC POR)

System power domain reset can reset the system power domain. The reset source is:

- Low voltage reset on the VPMC pin (VPMC POR)
- Debug Reset (DEBUG RST)
- Watchdog Reset (WDOGx RST)
- Software Reset (SW RST)

1.2.6 Boot Up

BootROM is the first program executed after the chip is powered on. It supports the following functions:

- from [Serial NOR FLASH Boot](#)
- UART/USB Boot
- [In-System Programming \(ISP\)](#)
- [Security Boot](#)
- [Low-Power Wakeup](#)
- [Multiple ROM API](#)

1.2.7 External memory

External memory interfaces include:

- One serial bus controller XPI can be connected to various off-chip SPI serial storage devices, or to devices that support serial bus. Each XPI:
 - supports 1/2/4/8-bit data mode and supports 2 CS chip select signals
 - supports SDR and DDR, up to 166MHz
 - Serial NOR Flash supporting Quad-SPI and Octal-SPI
 - supports serial NAND Flash
 - supports HyperBus, HyperRAM and HyperFlash
 - supports Quad/Oct SPI PSRAM
- 1 programmable parallel bus PPI

1.2.8 Algorithm hardware acceleration

algorithm hardware acceleration includes:

- transcends the function accelerator TFA to accelerate operations such as trigonometric functions
- Cyclic Redundancy Check CRC, supporting various CRC algorithms and custom algorithms

1.2.9 Electric Control System

Electric control system includes:

- Two 8-channel PWM modules PWMV2, 16 PWM outputs with modulation accuracy of up to 100ps, supporting complementary PWM outputs, dead zone insertion and fault protection
- 2 orthogonal encoder inputs QEIV2
- 2 orthogonal encoder outputs QEOV2
- 1 $\Sigma\Delta$ signal receiving unit SDM with built-in 4-channel SINC digital filter
- 1 programmable logic module PLB
- 1 interconnection manager TRGM
- Each module supports interaction with modules inside or outside the motor control system through the interconnection manager TRGM
- 1 synchronous timer SYNT for synchronization of the electric control system

1.2.10 Timer

timers include:

- Five groups of 32-bit universal timers, one group (PTMR) is located in the power management domain and supports low-power wake-up. Each group of universal timers includes 4 32-bit counters
- 3 Watchdogs, one of which (PWDG) is located in the power management domain

1.2.11 Real-time Ethernet System

Ethernet system includes:

- ESC EtherCAT Slave Controller
 - supports 3 network ports
 - 8 FMMU(Field Memory Management Unit)

- 8 8 SyncManagers
 - 60KB Process Data RAM
 - 64-bit Distributed Clock
- 1 Ethernet controller ENET
 - supports 10/100/1000 Mbps data transmission
 - supports RGMII/RMII/MII interface
 - supports Ethernet frame timestamps defined by IEEE 1588-2002 and IEEE 1588-2008 standards
 - MDIO main interface for configuring and managing PHY

1.2.12 Communication Peripherals

supports a variety of communication peripherals, including:

- Nine universal asynchronous transceivers UARTs, one of which (PUART) is located in the power management domain, supporting low-power wake-up
- 4 4 serial peripheral interfaces SPI
- 4 integrated circuit buses I2C, supporting standard (100kbps), fast (400kbps) and fast +(1 Mbps)
- 2 single-bus protocol controllers OWR, supporting 1-Wire protocol
- 4 controller local area network CAN, support CAN_FD
 - supports CAN 2.0B standard, 1Mbps
 - supports CAN FD, 8 Mbps
 - Support Timestamp
- 1 precise time protocol module PTPC. PTPC supports 2 groups of timestamp modules, each group contains a 64-bit counter. Connected to the CAN module, the CAN module can read timestamp information from the port at any time
- 1 USB OTG controller, integrated 1 high-speed USB-PHY
 - complies with Universal Serial Bus Specification Rev. 2.0

1.2.13 Analog Peripherals

Analog peripherals include:

- 2 16-bit analog to digital converters ADCs
 - 16-bit successive approximation ADC
 - supports 16 input channels
 - 2M sampling rate, 4M sampling rate (conversion accuracy is set to 12 bits)
 - supports differential input mode
- 1 analog comparator controller ACMP manages 2 analog comparators
 - Working voltage 3.0 ~ 3.6V, support rail-to-rail input
 - Built-in 12-bit DAC

1.2.14 Input and Output

- provides a total of 6 groups of up to 148 GPIO function multiplexing pins in PA ~ PF
- IO supports open-drain control, internal pull-up and down, drive ability adjustment, and built-in Schmitt trigger

- GPIO Controller
 - supports reading the input of any IO or controlling the output of IO
 - supports IO input triggering interrupts
- Fast GPIO Controller FGPIIO, which serves as the processor's private IO fast access interface
- provides a GPIO manager to manage the IO control permissions of each GPIO controller
- Two extended man-machine interfaces EUI, supporting up to 8x16 LED digital tube or dot matrix display output, and supporting up to 8x8 mechanical key array scan input
- has built-in logic observation module LOBS, which can capture and record IO waveforms

1.2.15 Information Security System

information security module includes:

- Online decryption module EXIP:
 - is tightly coupled to the serial bus controller XPI and supports online decryption of external NOR Flash
 - AES-128 CTR mode, zero-wait period decryption
 - supports key unblocking in RFC3394 and protects the data encryption key DEK by the key encryption key KEK
- OTP supports storage and protection;
 - Relevant key for EXIP
 - Key related to secure boot
 - Security debugging related key
 - Product Lifecycle Configuration
- Safe boot mechanism based on BOOT ROM-supports a trusted execution environment

1.2.16 System debugging

system debugging modules include:

- supports JTAG interface
 - supports RISC-V External Debug Support V0.13 specification
 - supports IEEE 1149.1
 - Access RISC-V core registers and CSR, access memory
- Debugging port locking feature
 - Open mode, open debugging functions
 - Lock mode, debugging functions are turned off, and can be unlocked through debugging key
 - shutdown mode, debugging function is off

1.3 document convention

1.3.1 register

The following abbreviations are used in the register description in this manual

abbreviation	Description
ro (read-only)	software can only read this bit.

abbreviation	Description
RC (Read Only, Read Clear)	Software can only read this bit and clear it after reading.
wo (write only)	software can only write this bit.
w1c (write 1 to clear)	software can clear this bit by writing 1. Writing 0 has no effect on the value of this bit.
w1s (write 1 to set 1)	software can set this bit to 1 by writing 1. Writing 0 has no effect on the value of this bit.
wc (write clear)	software can clear this bit by writing any value.
ws (Write Set 1)	The software can set this bit to 1 by writing any value.
rw (read/write)	software can read and write this bit.
rw1c (Read/Write 1 Clear)	Software can read this bit or clear it by writing 1. Writing 0 has no effect on the value of this bit.
rw1s (read/write 1 is set to 1)	software can read this bit or set the bit to 1 by writing 1. Writing 0 has no effect on the value of this bit.
rwc (Read/Write Clear)	Software can read this bit or clear it by writing any value.
rws (read/write set 1)	software can read this bit or set the bit to 1 by writing any value.

Table 2: register description acronym list

2 Processor Core

This product integrates a high-performance RISC-V processor as a CPU and complies with the following RISC-V specifications:

- The RISC-V Instruction Set Manual, Volume I: User-Level ISA, Version 2.2
- The RISC-V Instruction Set Manual Volume II: Privileged Architecture Version 1.11
- The RISC-V Debug Specification, Version 0.13

2.1 CPU

RISC-V32-bit high-performance embedded processor supports the following instruction sets:

- RISC-V, RV32 I: Basic integer instruction set
- RISC-V, M Extension: Multiply and Divide Instruction Set
- RISC-V, A Extension: Atomic Instruction Set
- RISC-V, F Extension: Single Precision Floating-Point Instruction Set
- RISC-V, D Extension: Double Precision Floating-Point Instruction Set
- RISC-V, C Extension: Compressed Instruction Set
- RISC-V, B Extension: Bit Operation Instruction Set

also supports extended instruction sets:

- RISC-V, P Extensions: SIMD and DSP Extended Instruction Set

Other features:

- 8-level sequential pipeline
- Dual Emission Superscalar Processor
- Dynamic branch prediction
- Processor Performance Monitor
- Unaligned memory access

2.2 Dual Core Configuration

Some models of this product integrate a RISC-V processor and are configured as:

- L1 instruction and data cache
 - 32KB L1 I-Cache, 4-way, 128x 64B cache line per way
 - 32KB L1 D-cache, 4-way, 128x 64B cache line per way
- Local memory for instructions and data: 128 KB ILM and 128 KB DLM

2.3 Bus and Memory Interface

The RISC-V processor is configured to support the following bus interfaces:

- bus main interface, which the processor uses to access memory and other resources on the chip.
- bus slave interface, which other bus masters on the chip can access the processor's data and instruction local memory
- instruction local memory interface that the processor uses to access high-speed instruction local memory

- Data local storage interface that the processor uses to access high-speed data local storage

2.4 TRAP

According to the RISC-V specification, processor instruction execution control flow transition caused by exceptions or interrupts is called trap, where exceptions are caused by the processor's own instruction execution, and interrupts are generated by interrupt sources inside and outside the processor. When a trap occurs, the processor will interrupt the current instruction execution flow, turn off the interrupt, save the required core general registers to the stack, and then execute the corresponding trap service program.

For details about RISC-V processor TRAP, please consult [Chapter 4](#).

For all peripherals in this product that support interrupt generation and their interrupt vector table, please refer to [Section 4.4](#).

2.5 Machine Timer MCHTMR

Machine Timer is a 64-bit timer that complies with the RISC-V specification. This timer runs on a fixed clock, provides a fixed time base, and is capable of generating Machine Timer Interrupts. The machine timer MCHTMR contains a 64-bit counter `mtime` and a 64-bit comparator `mtimecmpx`. When the counter `mtime` $\geq$ `mtimecmpx`, an interrupt is generated.

2.6 Hardware Performance Monitor

hardware performance monitor complies with RISC-V specifications and includes:

- `mcycle` / `mcycleh` CSR, a 64-bit counter that counts the number of clock cycles the processor has run since a specific moment.
- `minstret` / `minstreth` CSR, a 64-bit counter that counts the number of instructions completed by the processor since a specific time.
- `hpmcounter3` ~ `hpmcounter3 + 6` CSR, 4 64-bit event counters. By configuring CSR `mhpmevent3` ~ `6`, you can configure event counters to count the number of events since a specific moment:
 - The number of specific instructions executed. Instructions can be integer instructions such as LOAD, STORE, and multiplication, floating point number-related instructions, program flow control instructions such as jump, and return
 - Access to processor local memory ILM and DLM
 - Number of cache-related events: such as instruction or data cache hits, MISS times, etc.
 - Branch prediction module predicts the number of failures, etc.

2.7 privileged mode

This product complies with The RISC-V Instruction Set Manual Volume II: Privileged Architecture Version 1.11 specification.

- supports machine mode, also known as Machine Mode, M-mode, and M-mode
- supports supervisory mode, also known as Supervisor Mode, S-mode, and S-mode
- supports user mode, also known as User Mode, U-mode, and U-mode

supports Physical Memory Protection(PMP). Support 16 regions.

2.8 Physical Memory Attributes

system can be divided into several different areas, some corresponding to memory and some corresponding to peripheral control registers. These areas all have different properties. For example, some areas do not support reading, writing, or code execution, and some areas do not support caching. The RISC-V specification refers to the different access attributes of these different physical address storage intervals as PMA (Physical Memory Attributes).

The RISC-V processor of this product supports static PMA configurations and programmable PMA configurations.

Static PMA configuration supports setting the specified area of memory mapping to Device Region. Memory in Device Region does not support non-cacheable. On this product, the static PMA configuration is as follows:

- 0x30000000 ~ 0x3FFFFFFF, Device Region 0
- 0xF0000000 ~ 0xFFFFFFFF, Device Region 1

programmable PMA supports 16 PMA entries, allowing software to set the memory attributes of 16 PMAs through 4 PMA CFG CSRs and 16 PMA ADDR CSRs. Programmable PMA configurations have higher priority than static PMAs, and among the 16 PMA regions, PMA Regions with lower serial numbers have higher priority.

Users can specify the base address and length of the configuration area through the PMA ADDR CSR.

Users can configure the specified address area as Device Region or Memory Region through PMA CFG CSR.

Memory Region supports configuration as cacheable or non-cacheable.

corresponds to the cacheable Memory Region, and the caching strategy can be further configured. It is summarized as follows:

- Write-Back, Non-Allocate
 - When a read hit, read from cache
 - When a write hit, write to the cache and mark the cache line as modified
 - Read miss, read from memory
 - Write miss, write to memory
- Write-Back, Read-Allocate
 - When a read hit, read from cache
 - When a write hit, write to the cache and mark the cache line as modified
 - Read miss, read from memory, and allocate cache line
 - Write miss, write to memory
- Write-Back, Write-Allocate
 - Read from cache when a read hit
 - When a write hit, write to the cache and mark the cache line as modified
 - Read miss, read from memory
 - Write miss, allocate cache line, mark as modified, write cache
- Write-Back, Read-and-Write-Allocate
 - Read from cache when a read hit
 - When a write hit, write to the cache and mark the cache line as modified
 - Read miss, read from memory, and allocate cache line

- Write miss, allocate cache line, mark as modified, write cache
- Write-Through, Non-Allocate
 - When a read hit, read from cache
 - When a write hit, write to cache and write to memory
 - Read miss, read from memory
 - Write miss, write to memory
- Write-Through, Read-Allocate
 - When a read hit, read from cache
 - When a write hit, write to cache and write to memory
 - Read miss, read from memory, and allocate cache line
 - Write miss, write to memory

2.9 Physical Memory Protection

The RISC-V specification defines that the RISC-V processor can support the PMP (Physical Memory Protection) module. PMP supports read, write, and execute code protection for specified address intervals of memory maps. PMP's read, write, and code execution checks are specific to the privileged mode of the CPU, which allows users to authorize read, write, or code execution rights to specified address ranges to software in Supervisor Mode or User Mode by configuring PMP.

By configuring PMP, users can revoke the read, write, or code execution rights of software executing in Machine Mode for specified address ranges.

The RISC-V processor of this product supports 16 PMP entry entries. These 16 PMP entries can be configured through 4 PMP CFG CSR and 16 PMP ADDR CSR.

users can specify the base address and length of the memory area through PMP's ADDR CSR configuration.

Users can configure the CFG CSR of the PMP to specify that when the CPU is in Supervisor Mode or User Mode, code can be read, written, or executable for the specified address range.

users can lock the configuration of PMP CSR. Once locked, even if the CPU is in Machine Mode, the relevant CSR of the PMP cannot be modified until the next reset. At the same time, once locked, the PMP configuration no longer takes effect only for Supervisor Mode or User Mode, but also for Machine Mode.

2.10 related documents

For information on the RISC-V open source instruction set, please visit the RISC-V home page

<https://riscv.org/>。

Related specifications for the RISC-V instruction set, please visit

<https://riscv.org/technical/specifications/>。

For information on the DSP extension instructions of the RISC-V CPU of this product, please visit

<http://www.andestech.com/en/products-solutions/product-documentation/> and download the "AndeStar V5 DSP ISA Extension Specification".

3 Control status registers for RISC-V processors

This chapter lists the control and status registers of the RISC-V processor core of this product, referred to as CSR for short.

CSR is a private register of the RISC-V CPU. The RISC-V specification defines a series of standard CSRs. Processor manufacturers can also implement CSR support non-standard functions in the CPU.

The main functions of CSR are:

- contains information inherent to the processor: such as processor manufacturer information, architecture information, number of cores, etc.;
- Interrupt-related: such as interrupt switches and interrupt entries;
- interrupt response related: such as interrupt reason, interrupt return address and other information;
- Memory protection related: Set access properties of memory in different address spaces, such as readable, writable, executable, etc.;
- Performance statistics related and debugging interface related

RISC-V instruction set defines a series of instructions to access the CSR, through which the user can operate the CSR register, such as:

`csrr rd, csr //Read CSR` `csrw csr, rs //Write CSR`

Users can query the relevant RISC-V specifications for more information.

3.1 Control Status Register CSR Description

The CSR list of the control status register is as follows. Note the address offset in the list, which represents the coding space address of the CSR of the RISC-V CPU:

Offset	Register Name	Description	Type	Reset Value
0x000	USTATUS	User Status	standard read/write	0x00000000
0x004	UIE	User Interrupt Enable	standard read/write	0x00000000
0x005	UTVEC	User Trap Vector Base Address	standard read/write	0x00000000
0x040	USCRATCH	User Scratch Register	standard read/write	0x00000000
0x041	UEPC	User Exception Program Counter	standard read/write	0x00000000
0x042	UCAUSE	User Cause Register	standard read/write	0x00000000
0x043	UTVAL	User Trap Value	standard read/write	0x00000000
0x044	UIP	User Interrupt Pending	standard read/write	0x00000000
0x100	SSTATUS	Supervisor Status	standard read/write	0x00000000
0x102	SEDELEG	Supervisor Exception Delegation	standard read/write	0x00000000

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3
RISC-V PROCESSORS

3 CONTROL STATUS REGISTERS FOR

Offset	Register Name	Description	Type	Reset Value
0x103	SIDELEG	Supervisor Interrupt Delegation	standard read/write	0x00000000
0x104	SIE	Supervisor Interrupt Enable	standard read/write	0x00000000
0x105	STVEC	Supervisor Trap Vector Base Address	standard read/write	0x00000000
0x106	SCOUNTEREN	Supervisor Counter Enable Register	non-standard read/write	0x00000000
0x140	SSCRATCH	Supervisor Scratch Register	standard read/write	0x00000000
0x141	SEPC	Supervisor Exception Program Counter	standard read/write	0x00000000
0x142	SCAUSE	Supervisor Cause Register	standard read/write	0x00000000
0x143	STVAL	Supervisor Trap Value	standard read/write	0x00000000
0x144	SIP	Supervisor Interrupt Pending	standard read/write	0x00000000
0x180	SATP	Supervisor Address Translation and Protection	standard read/write	0x00000000
0x300	MSTATUS	Machine Status	standard read/write	0x00001800
0x301	MISA	Machine ISA Register	standard read/write	-
0x302	MEDELEG	Machine Exception Delegation	standard read/write	0x00000000
0x303	MIDELEG	Machine Interrupt Delegation	standard read/write	0x00000000
0x304	MIE	Machine Interrupt Enable	standard read/write	0x00000000
0x305	MTVEC	Machine Trap Vector Base Address	standard read/write	0x00000000
0x306	MCOUNTEREN	Machine Counter Enable	standard read/write	0x00000000
0x320	MCOUNTINHIBIT	Machine Counter-Inhibit	non-standard read/write	0x00000000
0x323	MHPMEVENT3	Machine Performance Monitoring Event Selector	standard read/write	0x00000000
0x324	MHPMEVENT4	Machine Performance Monitoring Event Selector	standard read/write	0x00000000
0x325	MHPMEVENT5	Machine Performance Monitoring Event Selector	standard read/write	0x00000000
0x326	MHPMEVENT6	Machine Performance Monitoring Event Selector	standard read/write	0x00000000
0x340	MSCRATCH	Machine Scratch Register	standard read/write	0x00000000
0x341	MEPC	Machine Exception Program Counter	standard read/write	0x00000000
0x342	MCAUSE	Machine Cause Register	standard read/write	0x00000000
0x343	MTVAL	Machine Trap Value	standard read/write	0x00000000
0x344	MIP	Machine Interrupt Pending	standard read/write	0x00000000
0x3A0	PMPCFG0	PMP Configuration Registers	standard read/write	0x00000000

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3
RISC-V PROCESSORS

3 CONTROL STATUS REGISTERS FOR

Offset	Register Name	Description	Type	Reset Value
0x3A1	PMPCFG1	PMP Configuration Registers	standard read/write	0x00000000
0x3A2	PMPCFG2	PMP Configuration Registers	standard read/write	0x00000000
0x3A3	PMPCFG3	PMP Configuration Registers	standard read/write	0x00000000
0x3B0	PMPADDR[PMPADDR0]	PMP Address Register	standard read/write	-
0x3B1	PMPADDR[PMPADDR1]	PMP Address Register	standard read/write	-
0x3B2	PMPADDR[PMPADDR2]	PMP Address Register	standard read/write	-
0x3B3	PMPADDR[PMPADDR3]	PMP Address Register	standard read/write	-
0x3B4	PMPADDR[PMPADDR4]	PMP Address Register	standard read/write	-
0x3B5	PMPADDR[PMPADDR5]	PMP Address Register	standard read/write	-
0x3B6	PMPADDR[PMPADDR6]	PMP Address Register	standard read/write	-
0x3B7	PMPADDR[PMPADDR7]	PMP Address Register	standard read/write	-
0x3B8	PMPADDR[PMPADDR8]	PMP Address Register	standard read/write	-
0x3B9	PMPADDR[PMPADDR9]	PMP Address Register	standard read/write	-
0x3BA	PMPADDR[PMPADDR10]	PMP Address Register	standard read/write	-
0x3BB	PMPADDR[PMPADDR11]	PMP Address Register	standard read/write	-
0x3BC	PMPADDR[PMPADDR12]	PMP Address Register	standard read/write	-
0x3BD	PMPADDR[PMPADDR13]	PMP Address Register	standard read/write	-
0x3BE	PMPADDR[PMPADDR14]	PMP Address Register	standard read/write	-
0x3BF	PMPADDR[PMPADDR15]	PMP Address Register	standard read/write	-
0x7A0	TSELECT	Trigger Select	standard read/write	0x00000000
0x7A1	TDATA1	Trigger Data 1	standard read/write	0x20000000
0x7A1	MCONTROL	Match Control	standard read/write	0x22400000
0x7A1	ICOUNT	Instruction Count	standard read/write	0x20000400
0x7A1	ITRIGGER	Interrupt Trigger	standard read/write	0x20000000
0x7A1	ETRIGGER	Exception Trigger	standard read/write	0x20000000
0x7A2	TDATA2	Trigger Data 2	standard read/write	0x00000000
0x7A3	TDATA3	Trigger Data 3	standard read/write	0x00000000
0x7A3	TEXTRA	Trigger Extra	standard read/write	0x00000000
0x7A4	TINFO	Trigger Info	standard read/write	-
0x7A5	TCONTROL	Trigger Control	standard read/write	0x00000000
0x7A8	MCONTEXT	Machine Context	standard read/write	0x00000000
0x7AA	SCONTEXT	Supervisor Context	standard read/write	0x00000000
0x7B0	DCSR	Debug Control and Status Register	debug-mode-only	0x40000603
0x7B1	DPC	Debug Program Counter	debug-mode-only	0x00000000
0x7B2	DSCRATCH0	Debug Scratch Register 0	debug-mode-only	0x00000000
0x7B3	DSCRATCH1	Debug Scratch Register 1	debug-mode-only	0x00000000
0x7C0	MILMB	Instruction Local Memory Base Register	non-standard read/write	-

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3
RISC-V PROCESSORS

3 CONTROL STATUS REGISTERS FOR

Offset	Register Name	Description	Type	Reset Value
0x7C1	MDLMB	Data Local Memory Base Register	non-standard read/write	-
0x7C2	MECC_CODE	ECC Code Register	non-standard read/write	0x00000001
0x7C3	MNVEC	NMI Vector Base Address Register	non-standard read/write	-
0x7C4	MXSTATUS	Machine Extended Status	non-standard read/write	0x00000000
0x7C5	MPFT_CTL	Performance Throttling Control Register	non-standard read/write	0x00000000
0x7C6	MHSP_CTL	Machine Hardware Stack Protection Control	non-standard read/write	0x00000000
0x7C7	MSP_BOUND	Machine SP Bound Register	non-standard read/write	0xFFFFFFFF
0x7C8	MSP_BASE	Machine SP Base Register	non-standard read/write	0xFFFFFFFF
0x7C9	MDCAUSE	Machine Detailed Trap Cause	non-standard read/write	0x00000000
0x7CA	MCACHE_CTL	Cache Control Register	non-standard read/write	0x00001800
0x7CB	MCCTLBEGINADDR	Machine CCTL Begin Address	non-standard read/write	0x00000000
0x7CC	MCCTLCOMMAND	Machine CCTL Command	non-standard read/write	0x00000000
0x7CD	MCCTLDATA	Machine CCTL Data	non-standard read/write	0x00000000
0x7CE	MCOUNTERWEN	Machine Counter Write Enable	non-standard read/write	0x00000000
0x7CF	MCOUNTERINTEN	Machine Counter Interrupt Enable	non-standard read/write	0x00000000
0x7D0	MMISC_CTL	Machine Miscellaneous Control Register	non-standard read/write	0x00000048
0x7D1	MCOUNTERMASK_M	Machine Counter Mask for Machine Mode	non-standard read/write	0x00000000
0x7D2	MCOUNTERMASK_S	Machine Counter Mask for Supervisor Mode	non-standard read/write	0x00000000
0x7D3	MCOUNTERMASK_U	Machine Counter Mask for User Mode	non-standard read/write	0x00000000
0x7D4	MCOUNTEROVF	Machine Counter Overflow Status	non-standard read/write	0x00000000

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3
RISC-V PROCESSORS

3 CONTROL STATUS REGISTERS FOR

Offset	Register Name	Description	Type	Reset Value
0x7D5	MSLIDELEG	Machine Supervisor Local Interrupt Delegation	non-standard read/write	0x00000000
0x7DF	MCLK_CTL	Clock Control Register	non-standard read/write	0x01FFFC07
0x7E0	DEXC2DBG	Exception Redirection Register	non-standard read/write	0x00000000
0x7E1	DDCAUSE	Debug Detailed Cause	non-standard read/write	0x00000000
0x800	UITB	Instruction Table Base Address Register	non-standard read/write	0x00000000
0x801	UCODE	Code Register	non-standard read/write	0x00000000
0x809	UDCAUSE	User Detailed Trap Cause	non-standard read/write	0x00000000
0x80B	UCCTLBEGINADDR	User CCTL Begin Address	non-standard read/write	0x00000000
0x80C	UCCTLCOMMAND	User CCTL Command	non-standard read/write	0x00000000
0x9C4	SLIE	Supervisor Local Interrupt Enable	non-standard read/write	0x00000000
0x9C5	SLIP	Supervisor Local Interrupt Pending	non-standard read/write	0x00000000
0x9C9	SDCAUSE	Supervisor Detailed Trap Cause	non-standard read/write	0x00000000
0x9CD	SCCTLDATA	Supervisor CCTL Data	non-standard read/write	0x00000000
0x9CF	SCOUNTERINTEN	Supervisor Counter Interrupt Enable	non-standard read/write	0x00000000
0x9D1	SCOUNTERMASK_M	Supervisor Counter Mask for Machine Mode	non-standard read/write	0x00000000
0x9D2	SCOUNTERMASK_S	Supervisor Counter Mask for Supervisor Mode	non-standard read/write	0x00000000
0x9D3	SCOUNTERMASK_U	Supervisor Counter Mask for User Mode	non-standard read/write	0x00000000
0x9D4	SCOUNTEROVF	Supervisor Counter Overflow Status	non-standard read/write	0x00000000
0x9E0	SCOUNTINHIBIT	Supervisor Counter-Inhibit	non-standard read/write	0x00000000
0x9E3	SHPMEVENT3	Supervisor Performance Monitoring Event Selector	non-standard read/write	0x00000000

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3
RISC-V PROCESSORS

3 CONTROL STATUS REGISTERS FOR

Offset	Register Name	Description	Type	Reset Value
0x9E4	SHPMEVENT4	Supervisor Performance Monitoring Event Selector	non-standard read/write	0x00000000
0x9E5	SHPMEVENT5	Supervisor Performance Monitoring Event Selector	non-standard read/write	0x00000000
0x9E6	SHPMEVENT6	Supervisor Performance Monitoring Event Selector	non-standard read/write	0x00000000
0xB00	MCYCLE	Machine Cycle Counter	standard read/write	0x00000000
0xB02	MINSTRET	Machine Instruction-Retired Counter	standard read/write	0x00000000
0xB03	MHPMCOUNTER3	Machine Performance Monitoring Counter	standard read/write	0x00000000
0xB04	MHPMCOUNTER4	Machine Performance Monitoring Counter	standard read/write	0x00000000
0xB05	MHPMCOUNTER5	Machine Performance Monitoring Counter	standard read/write	0x00000000
0xB06	MHPMCOUNTER6	Machine Performance Monitoring Counter	standard read/write	0x00000000
0xB80	MCYCLEH	Machine Cycle Counter	standard read/write	0x00000000
0xB82	MINSTRETH	Machine Instruction-Retired Counter	standard read/write	0x00000000
0xB83	MHPMCOUNTER3H	Machine Performance Monitoring Counter	standard read/write	0x00000000
0xB84	MHPMCOUNTER4H	Machine Performance Monitoring Counter	standard read/write	0x00000000
0xB85	MHPMCOUNTER5H	Machine Performance Monitoring Counter	standard read/write	0x00000000
0xB86	MHPMCOUNTER6H	Machine Performance Monitoring Counter	standard read/write	0x00000000
0xBC0	PMACFG0	PMA Configuration Registers	standard read/write	0x00000000
0xBC1	PMACFG1	PMA Configuration Registers	standard read/write	0x00000000
0xBC2	PMACFG2	PMA Configuration Registers	standard read/write	0x00000000
0xBC3	PMACFG3	PMA Configuration Registers	standard read/write	0x00000000
0xBD0	PMAADDR[PMAADDR0]	PMA Address Register	standard read/write	-
0xBD1	PMAADDR[PMAADDR1]	PMA Address Register	standard read/write	-
0xBD2	PMAADDR[PMAADDR2]	PMA Address Register	standard read/write	-
0xBD3	PMAADDR[PMAADDR3]	PMA Address Register	standard read/write	-
0xBD4	PMAADDR[PMAADDR4]	PMA Address Register	standard read/write	-
0xBD5	PMAADDR[PMAADDR5]	PMA Address Register	standard read/write	-
0xBD6	PMAADDR[PMAADDR6]	PMA Address Register	standard read/write	-
0xBD7	PMAADDR[PMAADDR7]	PMA Address Register	standard read/write	-
0xBD8	PMAADDR[PMAADDR8]	PMA Address Register	standard read/write	-

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3
RISC-V PROCESSORS

3 CONTROL STATUS REGISTERS FOR

Offset	Register Name	Description	Type	Reset Value
0xBD9	PMAADDR[PMAADDR9]	PMA Address Register	standard read/write	-
0xBDA	PMAADDR[PMAADDR10]	PMA Address Register	standard read/write	-
0xBDB	PMAADDR[PMAADDR11]	PMA Address Register	standard read/write	-
0xBDC	PMAADDR[PMAADDR12]	PMA Address Register	standard read/write	-
0xBDD	PMAADDR[PMAADDR13]	PMA Address Register	standard read/write	-
0xBDE	PMAADDR[PMAADDR14]	PMA Address Register	standard read/write	-
0xBDF	PMAADDR[PMAADDR15]	PMA Address Register	standard read/write	-
0xC00	CYCLE	CYCLE Register		0x00000000
0xC80	CYCLEH	CYCLE High 32-bit register		0x00000000
0xF11	MVENDORID	Machine Vendor ID Register	standard read only	0x0000031E
0xF12	MARCHID	Machine Architecture ID Register	standard read only	0x00000045
0xF13	MIMPID	Machine Implementation ID Register	standard read only	-
0xF14	MHARTID	Hart ID Register	standard read only	0x00000000

Table 3: CSR Register List

3.2 Control Status Register CSR Details

control status register CSR is as follows:

3.2.1 USTATUS (0x0) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RSVD																												UPIE	RSVD		UIE	
N/A																												RW	N/A		RW	
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	x	x	x	0	

USTATUS [31:0]

Bits	Name	Description
4	UPIE	UPIE holds the value of the UIE bit prior to a trap.
0	UIE	U mode interrupt enable bit. 0:Disabled 1:Enabled

USTATUS Bits

3.2.2 UIE (0x4) (standard read/write)

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3
RISC-V PROCESSORS

3 CONTROL STATUS REGISTERS FOR

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																							UEIE	RSVD			UTIE	RSVD			USIE
N/A																							RW	N/A			RW	N/A			RW
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	x	x	x	0	x	x	x	0

UIE [31:0]

Bits	Name	Description
8	UEIE	U mode external interrupt enable bit 0:Disabled 1:Enabled
4	UTIE	U mode timer interrupt enable bit. 0:Disabled 1:Enabled
0	USIE	U mode software interrupt enable bit. 0:Disabled 1:Enabled

UIE Bits

3.2.3 UTVEC (0x5) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BASE_31_2																															RSVD
RW																															N/A
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x

UTVEC [31:0]

Bits	Name	Description
31-2	BASE_31_2	Base address for interrupt and exception handlers. See description above for alignment requirements when PLIC is in the vector mode.

UTVEC Bits

3.2.4 USCRATCH (0x40) (standard read/write)

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3
RISC-V PROCESSORS

3 CONTROL STATUS REGISTERS FOR

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USCRATCH																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

USCRATCH [31:0]

Bits	Name	Description
31-0	USCRATCH	Scratch register storage.

USCRATCH Bits

3.2.5 UEPC (0x41) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EPC																															
RW																															RSVD
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x

UEPC [31:0]

Bits	Name	Description
31-1	EPC	Exception program counter.

UEPC Bits

3.2.6 UCAUSE (0x42) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
INTERRUPT	RSVD																					EXCEPTION_CODE															
	N/A																					RW															
0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0						

UCAUSE [31:0]

Bits	Name	Description
31	INTERRUPT	Interrupt.

Bits	Name	Description
9-0	EXCEPTION_CODE	Exception Code. When interrupt is 1: 0:User software interrupt 4:User timer interrupt 8:User external interrupt When interrupt is 0: 0:Instruction address misaligned 1:Instruction access fault 2:Illegal instruction 3:Breakpoint 4:Load address misaligned 5:Load access fault 6:Store/AMO address misaligned 7:Store/AMO access fault 8:Environment call from U-mode 9-11:Reserved 12:Instruction page fault 13:Load page fault 14:Reserved 15:Store/AMO page fault 32: Stack overflow exception 33:Stack underflow exception 40-47:Reserved

UCAUSE Bits

3.2.7 UTVAL (0x43) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UTVAL																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

UTVAL [31:0]

Bits	Name	Description
31-0	UTVAL	Exception-specific information for software trap handling.

UTVAL Bits

3.2.8 UIP (0x44) (standard read/write)

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3
RISC-V PROCESSORS

3 CONTROL STATUS REGISTERS FOR

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																							UEIP	RSVD			UTIP	RSVD			USIP
N/A																							RW	N/A			RW	N/A			RW
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	x	x	x	0	x	x	x	0

UIP [31:0]

Bits	Name	Description
8	UEIP	U mode external interrupt pending bit. 0:Not pending 1:Pending
4	UTIP	U mode timer interrupt pending bit. 0:Not pending 1:Pending
0	USIP	U mode software interrupt pending bit. 0:Not pending 1:Pending

UIP Bits

3.2.9 SSTATUS (0x100) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SD	RSVD											MXR	SUM	RSVD	XS	FS	RSVD					SPP	RSVD		SPIE	UPIE	RSVD		SIE	UIE	
RO	N/A											RW	RW	N/A	RO	RW	N/A					RW	N/A	RW	RW	N/A	RW	RW			
0	x	x	x	x	x	x	x	x	x	x	x	0	0	x	0	0	0	0	x	x	x	x	0	x	x	0	0	x	x	0	0

SSTATUS [31:0]

Bits	Name	Description
31	SD	SD summarizes whether either the FS field or XS field is dirty.
19	MXR	MXR controls whether execute-only pages are readable. It has no effect when page-based virtual memory is not in effect. 0:Execute-only pages are not readable 1:Execute-only pages are readable

Bits	Name	Description
18	SUM	SUM controls whether a S-mode load/store instruction to a user accessible page is allowed or not when page translation is enabled. It is in effect in two scenarios: (a) M-mode with MPRV=1 and MPP=S, and (b) in S-mode. It has no effect when page-based virtual memory is not in effect. A page is user accessible when the U bit of the corresponding PTE entry is 1. 0:Not Allowed 1:Allowed
16-15	XS	XS holds the status of the architectural states (ACE registers) of ACE instructions. The value of this field is zero if ACE extension is not configured. This field is primarily managed by software. The processor hardware assists the state managements in two regards: Illegal instruction exceptions are triggered when XS is Off. XS is updated to the Dirty state with the execution of ACE instructions when XS is not Off. Changing the setting of this field has no effect on the contents of ACE states. In particular, setting XS to Off does not destroy the states, nor does setting XS to Initial clear the contents. The same copy of XS bits are shared by both mstatus and sstatus. Normally the supervisor mode privileged software would use the XS bits to manage deferred context switches of ACE states. Machine mode software should be more conservative in managing context switches using XS bits 0:Off 1:Initial 2:Clean 3:Dirty

Bits	Name	Description
14-13	FS	FS holds the status of the architectural states of the floating-point unit, including the fcsr CSR and f0 -f31 floating-point data registers. The value of this field is zero and read-only if the processor does not have FPU. This field is primarily managed by software. The processor hardware assists the state managements in two regards: Attempts to access fcsr or any f register raise an illegal-instruction exception when FS is Off. Otherwise, FS is updated to the Dirty state by any instruction that updates fcsr or any f register. Changing the setting of this field has no effect on the contents of the floating-point register states. In particular, setting FS to Off does not destroy the states, nor does setting FS to Initial clear the contents. The same copy of FS bits are shared by both mstatus and sstatus. Normally the supervisor mode privileged software would use the FS bits to manage deferred context switches of FPU states. Machine mode software should be more conservative in managing context switches using FS bits. 0:Off 1:Initial 2:Clean 3:Dirty
8	SPP	SPP holds the privilege mode prior to a trap. Encoding is 1 for S-mode and 0 for U-mode.
5	SPIE	SPIE holds the value of the SIE bit prior to a trap.
4	UPIE	UPIE holds the value of the UIE bit prior to a trap.
1	SIE	S mode interrupt enable bit 0 Disabled 1 Enabled
0	UIE	U mode interrupt enable bit. 0 Disabled 1 Enabled

SSTATUS Bits

3.2.10 SEDELEG (0x102) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD															SPF	RSVD	LPF	IPF	RSVD			UEC	SAF	SAM	LAF	LAM	B	II	IAF	IAM	

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3
RISC-V PROCESSORS

3 CONTROL STATUS REGISTERS FOR

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
N/A																RW	N/A	RW	RW	N/A			RW	RW	RW	RW	RW	RW	RW	RW	RW
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	x	0	0	x	x	x	0	0	0	0	0	0	0	0	0

SEDELEG [31:0]

Bits	Name	Description
15	SPF	SPF indicates whether a Store/AMO Page Fault exception will be delegated to U-mode 0:Do not redirect 1:Redirect
13	LPF	LPF indicates whether a Load Page Fault exception will be delegated to U-mode 0:Do not redirect 1:Redirect
12	IPF	IPF indicates whether an Instruction Page Fault exception will be delegated to U-mode 0:Do not redirect 1:Redirect
8	UEC	UEC indicates whether an exception triggered by environment call from U-mode will be delegated to U-mode 0:Do not redirect 1:Redirect
7	SAF	SAF indicates whether a Store/AMO Access Fault exception will be delegated to U-mode 0:Do not redirect 1:Redirect
6	SAM	SAM indicates whether a Store/AMO Address Misaligned exception will be delegated to U-mode 0:Do not redirect 1:Redirect
5	LAF	LAF indicates whether a Load Access Fault exception will be delegated to U-mode 0:Do not redirect 1:Redirect
4	LAM	LAM indicates whether a Load Address Misaligned exception will be delegated to U-mode 0:Do not redirect 1:Redirect

Bits	Name	Description
3	B	B indicates whether an exception triggered by breakpoint will be delegated to U-mode 0:Do not redirect 1:Redirect
2	II	II indicates whether an Illegal Instruction exception will be delegated to U-mode 0:Do not redirect 1:Redirect
1	IAF	IAF indicates whether an Instruction Access Fault exception will be delegated to U-mode 0:Do not redirect 1:Redirect
0	IAM	IAM indicates whether an Instruction Address Misaligned exception will be delegated to U-mode.. 0:Do not redirect 1:Redirect

SEDELEG Bits

3.2.11 SIDELEG (0x103) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																							UEI	RSVD			UTI	RSVD			USI
N/A																							RW	N/A			RW	N/A			RW
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	x	x	x	0	x	x	x	0

SIDELEG [31:0]

Bits	Name	Description
8	UEI	UEI indicates whether an U-mode external interrupt will be delegated to S-mode 0:Do not redirect 1:Redirect
4	UTI	UTI indicates whether an U-mode timer interrupt will be delegated to S-mode 0:Do not redirect 1:Redirect
0	USI	USI indicates whether an U-mode software interrupt will be delegated to S-mode. 0:Do not redirect 1:Redirect

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3
RISC-V PROCESSORS

3 CONTROL STATUS REGISTERS FOR

Bits	Name	Description

SIDELEG Bits

3.2.12 SIE (0x104) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																						SEIE	UEIE	RSVD		STIE	UTIE	RSVD		SSIE	USIE
N/A																						RW	RW	N/A		RW	RW	N/A		RW	RW
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	x	x	0	0	x	x	0	0

SIE [31:0]

Bits	Name	Description
9	SEIE	S mode external interrupt enable bit 0:Disabled 1:Enabled
8	UEIE	U mode external interrupt enable bit 0:Disabled 1:Enabled
5	STIE	S mode timer interrupt enable bit. 0:Disabled 1:Enabled
4	UTIE	U mode timer interrupt enable bit 0:Disabled 1:Enabled
1	SSIE	S mode software interrupt enable bit. 0:Disabled 1:Enabled
0	USIE	U mode software interrupt enable bit. 0:Disabled 1:Enabled

SIE Bits

3.2.13 STVEC (0x105) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BASE_31_2																															RSVD
RW																															N/A

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3
RISC-V PROCESSORS

3 CONTROL STATUS REGISTERS FOR

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x

STVEC [31:0]

Bits	Name	Description
31-2	BASE_31_2	Base address for interrupt and exception handlers. See description above for alignment requirements when PLIC is in the vector mode.

STVEC Bits

3.2.14 SCOUNTEREN (0x106) (non-standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								HPM6	HPM5	HPM4	HPM3	IR	RSVD	CY	
N/A																								RW	RW	RW	RW	RW	N/A	RW	
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	x	0	

SCOUNTEREN [31:0]

Bits	Name	Description
6	HPM6	See register description
5	HPM5	See register description
4	HPM4	See register description
3	HPM3	See register description
2	IR	See register description
0	CY	See register description

SCOUNTEREN Bits

3.2.15 SSCRATCH (0x140) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
SSCRATCH																																
RW																																
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

SSCRATCH [31:0]

Bits	Name	Description
31-0	SSCRATCH	Scratch register storage.

Bits	Name	Description

SSCRATCH Bits

3.2.16 SEPC (0x141) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EPC																															RSVD
RW																															N/A
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x

SEPC [31:0]

Bits	Name	Description
31-1	EPC	Exception program counter.

SEPC Bits

3.2.17 SCAUSE (0x142) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTERRUPT	RSVD																					EXCEPTION_CODE									
RW	N/A																					RW									
0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0

SCAUSE [31:0]

Bits	Name	Description
31	INTERRUPT	Interrupt.

Bits	Name	Description
9-0	EXCEPTION_CODE	Exception Code. When interrupt is 1: 0:User software interrupt 1:Supervisor software interrupt 4:User timer interrupt 5:Supervisor timer interrupt 8:User external interrupt 9:Supervisor external interrupt 256+16:Slave port ECC error interrupt (S-mode) 256+17:Bus write transaction error interrupt (S-mode) 256+18:Performance monitor overflow interrupt(S-mode) When interrupt is 0: 0:Instruction address misaligned 1:Instruction access fault 2:Illegal instruction 3:Breakpoint 4:Load address misaligned 5:Load access fault 6:Store/AMO address misaligned 7:Store/AMO access fault 8:Environment call from U-mode 9:Environment call from S-mode 11:10:Reserved 12:Instruction page fault 13:Load page fault 14:Reserved 15:Store/AMO page fault 32:Stack overflow exception 33:Stack underflow exception 40-47:Reserved

SCAUSE Bits

3.2.18 STVAL (0x143) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STVAL																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

STVAL [31:0]

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3
RISC-V PROCESSORS

3 CONTROL STATUS REGISTERS FOR

Bits	Name	Description
31-0	STVAL	Exception-specific information for software trap handling.

STVAL Bits

3.2.19 SIP (0x144) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																						SEIP	UEIP	RSVD		STIP	UTIP	RSVD		SSIP	USIP
N/A																						RO	RW	N/A		RO	RO	N/A		RW	RW
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	x	x	0	0	x	x	0	0

SIP [31:0]

Bits	Name	Description
9	SEIP	S mode external interrupt pending bit. 0:Not pending 1:Pending
8	UEIP	U mode external interrupt pending bit. 0:Not pending 1:Pending
5	STIP	S mode timer interrupt pending bit. 0:Not pending 1:Pending
4	UTIP	U mode timer interrupt pending bit 0:Not pending 1:Pending
1	SSIP	S mode software interrupt pending bit. 0:Not pending 1:Pending
0	USIP	U mode software interrupt pending bit. 0:Not pending 1:Pending

SIP Bits

3.2.20 SATP (0x180) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODE	ASID										PPN																				
RW	RW										RW																				

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3
RISC-V PROCESSORS

3 CONTROL STATUS REGISTERS FOR

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SATP [31:0]

Bits	Name	Description
31	MODE	MODE holds the page translation mode. When MODE is Bare, virtual addresses are equal to physical addresses in S-mode. When MMU is not supported in the product, this CSR will be hardwired to 0. 0:No page translation 1:Page-based 32-bit virtual addressing
30-22	ASID	ASID holds the address space identifier.
21-0	PPN	PPN holds the physical page number of the root page table.

SATP Bits

3.2.21 MSTATUS (0x300) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SD	RSVD								TSR	TW	TVM	MXR	SUM	MPRV	XS		FS		MPP		RSVD		SPP	MPIE	RSVD	SPIE	UIPE	MIE	RSVD	SIE	UIE
RO	N/A								RW	RW	RW	RW	RW	RW	RO		RW		RW		N/A		RW	RW	N/A	RW	RW	RW	N/A	RW	RW
0	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	1	1	x	x	0	0	x	0	0	0	x	0	0

MSTATUS [31:0]

Bits	Name	Description
31	SD	SD summarizes whether either the FS field or XS field is dirty.
22	TSR	TSR controls whether executing SRET instructions in S-mode will raise illegal instruction exceptions. It is hardwired to 0 when S-mode is not supported. 0: Normal execution 1: Raising exceptions
21	TW	TW controls whether executing WFI instructions in S-mode will raise illegal instruction exceptions. It is hardwired to 0 when S-mode is not supported. 0: Normal execution 1: Raising exceptions

Bits	Name	Description
20	TVM	TVM controls whether performing certain virtual memory operations in S-mode will raise illegal instruction exceptions. The operations include accessing the satp register and executing the SFENCE.VMA instruction. It is hardwired to 0 when S-mode is not supported. 0:Normal execution 1:Raising exceptions
19	MXR	MXR controls whether execute-only pages are readable. It has no effect when page-based virtual memory is not in effect 0:Execute-only pages are not readable 1:Execute-only pages are readable
18	SUM	SUM controls whether a S-mode load/store instruction to a user accessible page is allowed or not when page translation is enabled. It is in effect in two scenarios: (a) M-mode with MPRV=1 and MPP=S, and (b) in S-mode. It has no effect when page-based virtual memory is not in effect. A page is user accessible when the U bit of the corresponding PTE entry is 1. It is hardwired to 0 when S-mode is not supported. 0:Not Allowed 1:Allowed
17	MPRV	When the MPRV bit is set, the memory access privilege for load and store are specified by the MPP field. When U-mode is not available, this field is hardwired to 0.
16-15	XS	XS holds the status of the architectural states (ACE registers) of ACE instructions. The value of this field is zero if ACE extension is not configured. This field is primarily managed by software. The processor hardware assists the state managements in two regards: Illegal instruction exceptions are triggered when XS is Off. XS is updated to the Dirty state with the execution of ACE instructions when XS is not Off. Changing the setting of this field has no effect on the contents of ACE states. In particular, setting XS to Off does not destroy the states, nor does setting XS to Initial clear the contents. 0:Off 1:Initial 2:Clean 3:Dirty

Bits	Name	Description
14-13	FS	FS holds the status of the architectural states of the floating-point unit, including the fcsr CSR and f0 –f31 floating-point data registers. The value of this field is zero and read-only if the processor does not have FPU. This field is primarily managed by software. The processor hardware assists the state managements in two regards: Attempts to access fcsr or any f register raise an illegal-instruction exception when FS is Off. FS is updated to the Dirty state with the execution of any instruction that updates fcsr or any f register when FS is Initial or Clean. Changing the setting of this field has no effect on the contents of the floating-point register states. In particular, setting FS to Off does not destroy the states, nor does setting FS to Initial clear the contents. 0:Off 1:Initial 2:Clean 3:Dirty
12-11	MPP	MPP holds the privilege mode prior to a trap. Encoding for privilege mode is described in Table5. When U-mode is not available, this field is hardwired to 3.
8	SPP	SPP holds the privilege mode prior to a trap. Encoding is 1 for S-mode and 0 for U-mode.
7	MPIE	MPIE holds the value of the MIE bit prior to a trap.
5	SPIE	SPIE holds the value of the SIE bit prior to a trap.
4	UPIE	UPIE holds the value of the UIE bit prior to a trap.
3	MIE	M mode interrupt enable bit. 0: Disabled 1: Enabled
1	SIE	S mode interrupt enable bit. 0: Disabled 1: Enabled
0	UIE	U mode interrupt enable bit. 0: Disabled 1: Enabled

MSTATUS Bits

3.2.22 MISA (0x301) (standard read/write)

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3
RISC-V PROCESSORS

3 CONTROL STATUS REGISTERS FOR

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BASE		RSVD				Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A
RO		N/A				RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
0	1	x	x	x	x	0	0	1	0	0	*	0	*	0	0	0	0	*	1	0	0	0	1	0	0	*	0	*	1	0	*

MISA [31:0]

Bits	Name	Description
31-30	BASE	The general-purpose register width of the native base integer ISA. 0:Reserved 1:32 2:64 3:128
25	Z	Reserved
24	Y	Reserved
23	X	Non-standard extensions present
22	W	Reserved
21	V	Tentatively reserved for Vector extension
20	U	User mode implemented 0:Machine 1:Machine + User / Machine + Supervisor + User
19	T	Tentatively reserved for Transactional Memory extension
18	S	Supervisor mode implemented 0:Machine / Machine + User 1:Machine + Supervisor + User
17	R	Reserved
16	Q	Quad-precision floating-point extension
15	P	Tentatively reserved for Packed-SIMD extension
14	O	Reserved
13	N	User-level interrupts supported 0:no 1:yes
12	M	Integer Multiply/Divide extension
11	L	Tentatively reserved for Decimal Floating-Point extension
10	K	Reserved
9	J	Tentatively reserved for Dynamically Translated Languages extension
8	I	RV32I/64I/128I base ISA
7	H	Reserved
6	G	Additional standard extensions present

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3
RISC-V PROCESSORS

3 CONTROL STATUS REGISTERS FOR

Bits	Name	Description
5	F	Single-precision floating-point extension 0:none 1:double+single precision / single precision
4	E	RV32E base ISA
3	D	Double-precision floating-point extension 0:single precision / none 1:double+single precision
2	C	Compressed extension
1	B	Tentatively reserved for Bit operations extension
0	A	Atomic extension 0:no 1:yes

MISA Bits

3.2.23 MEDELEG (0x302) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																SPF	RSVD	LPF	IPF	RSVD		SEC	UEC	SAF	SAM	LAF	LAM	RSVD	II	IAF	IAM
N/A																RW	N/A	RW	RW	N/A		RW	RW	RW	RW	RW	RW	N/A	RW	RW	RW
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	x	0	0	x	x	0	0	0	0	0	0	x	0	0	0

MEDELEG [31:0]

Bits	Name	Description
15	SPF	SPF indicates whether a Store/AMO Page Fault exception will be delegated to S-mode. 0:Not delegate 1:delegate
13	LPF	LPF indicates whether a Load Page Fault exception will be delegated to S-mode. 0:Not delegate 1:delegate
12	IPF	IPF indicates whether an Instruction Page Fault exception will be delegated to S-mode. 0:Not delegate 1:delegate
9	SEC	SEC indicates whether an exception triggered by environment call from S-mode will be delegated to S-mode. 0:Not delegate 1:delegate

Bits	Name	Description
8	UEC	UEC indicates whether an exception triggered by environment call from U-mode will be delegated to S-mode. 0:Not delegate 1:delegate
7	SAF	SAF indicates whether a Store/AMO Access Fault exception will be delegated to S-mode. 0:Not delegate 1:delegate
6	SAM	SAM indicates whether a Store/AMO Address Misaligned exception will be delegated to S-mode 0:Not delegate 1:delegate
5	LAF	LAF indicates whether a Load Access Fault exception will be delegated to S-mode. 0:Not delegate 1:delegate
4	LAM	LAM indicates whether a Load Address Misaligned exception will be delegated to S-mode. 0:Not delegate 1:delegate
2	II	II indicates whether an Illegal Instruction exception will be delegated to S-mode. 0:Not delegate 1:delegate
1	IAF	IAF indicates whether an Instruction Access Fault exception will be delegated to S-mode. 0:Not delegate 1:delegate
0	IAM	IAM indicates whether an Instruction Address Misaligned exception will be delegated to S-Mode 0:Not delegate 1:delegate

MEDELEG Bits

3.2.24 MIDELEG (0x303) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																						SEI	UEI	RSVD	STI	UTI	RSVD	SSI	USI		
N/A																						RW	RW	N/A	RW	RW	N/A	RW	RW		

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3
RISC-V PROCESSORS

3 CONTROL STATUS REGISTERS FOR

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	x	x	0	0	x	x	0	0

MIDELEG [31:0]

Bits	Name	Description
9	SEI	SEI indicates whether an S-mode external interrupt will be delegated to S-mode. 0:Not delegate 1:delegate
8	UEI	UEI indicates whether an U-mode external interrupt will be delegated to S-mode. 0:Not delegate 1:delegate
5	STI	STI indicates whether an S-mode timer interrupt will be delegated to S-mode. 0:Not delegate 1:delegate
4	UTI	UTI indicates whether an U-mode timer interrupt will be delegated to S-mode. 0:Not delegate 1:delegate
1	SSI	SSI indicates whether an S-mode software interrupt will be delegated to S-mode. 0:Not delegate 1:delegate
0	USI	USI indicates whether an U-mode software interrupt will be delegated to S-mode. 0:Not delegate 1:delegate

MIDELEG Bits

3.2.25 MIE (0x304) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD													PMOVI	BWEI	IMECCI	RSVD				MEIE	RSVD	SEIE	UEIE	MTIE	RSVD	STIE	UTIE	MSIE	RSVD	SSIE	USIE
N/A													RW	RW	RW	N/A				RW	N/A	RW	RW	RW	N/A	RW	RW	RW	N/A	RW	RW
x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	x	x	x	x	0	x	0	0	0	x	0	0	0	x	0	0

MIE [31:0]

Bits	Name	Description
18	PMOVI	Performance monitor overflow local interrupt enable bit 0:Disabled 1:Enabled
17	BWEI	Bus read/write transaction error local interrupt enable bit. The processor may receive bus errors on load/store instructions or cache writebacks. 0:Disabled 1:Enabled
16	IMECCI	Imprecise ECC error local interrupt enable bit. The processor may receive imprecise ECC errors on slave port accesses or cache writebacks. 0:Disabled 1:Enabled
11	MEIE	M mode external interrupt enable bit 0:Disabled 1:Enabled
9	SEIE	S mode external interrupt enable bit 0:Disabled 1:Enabled
8	UEIE	U mode external interrupt enable bit 0:Disabled 1:Enabled
7	MTIE	M mode timer interrupt enable bit. 0:Disabled 1:Enabled
5	STIE	S mode timer interrupt enable bit. 0:Disabled 1:Enabled
4	UTIE	U mode timer interrupt enable bit. 0:Disabled 1:Enabled
3	MSIE	M mode software interrupt enable bit 0:Disabled 1:Enabled
1	SSIE	S mode software interrupt enable bit. 0:Disabled 1:Enabled
0	USIE	U mode software interrupt enable bit. 0:Disabled 1:Enabled

MIE Bits

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3
RISC-V PROCESSORS

3 CONTROL STATUS REGISTERS FOR

3.2.26 MTVEC (0x305) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BASE_31_2																RW														RSVD	
																														N/A	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x

MTVEC [31:0]

Bits	Name	Description
31-2	BASE_31_2	Base address for interrupt and exception handlers. See description above for alignment requirements when PLIC is in the vector mode

MTVEC Bits

3.2.27 MCOUNTEREN (0x306) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								HPM6	HPM5	HPM4	HPM3	IR	TM	CY	
N/A																								RW	RW	RW	RW	RW	RW	RW	
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	

MCOUNTEREN [31:0]

Bits	Name	Description
6	HPM6	See register description
5	HPM5	See register description
4	HPM4	See register description
3	HPM3	See register description
2	IR	See register description
1	TM	See register description
0	CY	See register description

MCOUNTEREN Bits

3.2.28 MCOUNTINHIBIT (0x320) (non-standard read/write)

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3
RISC-V PROCESSORS

3 CONTROL STATUS REGISTERS FOR

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								HPM6	HPM5	HPM4	HPM3	IR	TM	CY	
N/A																								RW	RW	RW	RW	RW	RW	RW	
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	

MCOUNTINHIBIT [31:0]

Bits	Name	Description
6	HPM6	See register description.
5	HPM5	See register description.
4	HPM4	See register description.
3	HPM3	See register description.
2	IR	See register description.
1	TM	See register description.
0	CY	See register description.

MCOUNTINHIBIT Bits

3.2.29 MHPMEVENT3 (0x323) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																							SEL					TYPE			
N/A																							RW					RW			
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	

MHPMEVENT3 [31:0]

Bits	Name	Description
8-4	SEL	Event Selectors table
3-0	TYPE	Event Selectors table

MHPMEVENT3 Bits

TYPE	SEL	Event Name	Description
0	1	Cycle count	Number of elapsed processor clock cycles
0	2	Retired instruction count	Number of retired instructions
0	3	Integer load instruction count	Number of retired load instructions (including LR).
0	4	Integer store instruction count	Number of retired store instructions (including SC).
0	5	Atomic instruction count	Number of retired atomic instructions (not including LR and SC).

TYPE	SEL	Event Name	Description
0	6	System instruction count	Number of retired SYSTEM instructions(instructions with major opcode equal to 0b1110011).
0	7	Integer computational instruction count	Number of retired integer computational instructions.
0	8	Conditional branch instruction count	Number of retired conditional branch instructions.
0	9	Taken conditional branch instruction count	Number of retired conditional branch instructions that are taken.
0	10	JAL instruction count	Number of retired JAL instructions.
0	11	JALR instruction count	Number of retired JALR instructions.This event selector also counts the events monitored by the return instruction count event selector defined in the next row.
0	12	Return instruction count	Number of retired return instructions. Return instructions are JALR instructions with zero immediate offset and the following operands: • (rd != x1/x5) and (rs1 == x1/x5) • rd == x1 and rs1 == x5 • rd == x5 and rs1 == x1
0	13	Control transfer instruction count	Number of retired unconditional jumps (JAL and JALR) and conditional branch instructions.
0	14	EXEC.IT instruction count	Number of retired EXEC.IT instructions.
0	15	Integer multiplication instruction count	Number of retired integer multiplication instructions.
0	16	Integer division instruction count	Number of retired integer division/remainder instructions.
0	17	Floating-point load instruction count	Number of retired floating-point load instructions.
0	18	Floating-point store instruction count	Number of retired floating-point store instructions.
0	19	Floating-point addition instruction count	Number of retired floating-point addition/subtraction instructions.
0	20	Floating-point multiplication instruction count	Number of retired floating-point multiplication instructions.
0	21	Floating-point fused multiply-add instruction count	Number of retired floating-point fused multiply-add/subtraction instructions (FMADD, FMSUB, FNMSUB, FNMADD).
0	22	Floating-point division or square-root instruction count	Number of retired floating-point division/square-root instructions.

TYPE	SEL	Event Name	Description
0	23	Other floating-point instruction count	Number of retired floating-point instructions not counted by the previous floating-point instruction event selectors
0	24	Integer multiplication and add/sub instruction count	Number of retired integer multiplication and add/sub instructions.
0	25	Retired operation count	Number of retired operations. a floating-point multiply-add instruction is counted as 2 operations. other instructions are counted as 1 operation
1	0	ILM access	Number of ILM transfers, including speculative instruction fetch, load/store accesses, ECC repair and slave port accesses.
1	1	DLM access	Number of DLM transfers, including speculative load/store accesses, ECC repair and slave port accesses.
1	2	I-Cache access	Number of completed I-Cache fetch access.
1	3	I-Cache miss	Number of I-Cache fetch miss.
1	4	D-Cache access	Number of completed D-Cache load-and-store access. Misaligned load/store accesses might increase this counter by either one or two, depending on access sizes and alignments. Only misaligned accesses crossing two cache lines are guaranteed to result in increment of two.
1	5	D-Cache miss	The event counts the number of D-Cache load-and-store miss. Misaligned load/store accesses might increase this counter by either zero, one or two, depending on access sizes, alignments and whether the accessed lines are in D-Cache.
1	6	D-Cache load access	Number of completed D-Cache load access. See the D-Cache access count event selector for the handling of misaligned load accesses.
1	7	D-Cache load miss	Number of D-Cache load miss. See the D-Cache miss count event selector for the handling of misaligned load accesses.
1	8	D-Cache store access	Number of completed D-Cache store access. See the D-Cache access count event selector for the handling of misaligned load accesses.
1	9	D-Cache store miss	Number of D-Cache store miss. See the D-Cache miss count event selector for the handling of misaligned load accesses.
1	10	D-Cache writeback	Number of D-Cache writeback.

TYPE	SEL	Event Name	Description
1	11	Cycles waiting for I-Cache fill data	Number of cycles waiting for the return of the critical word of I-Cache misses from the system bus. This event selector does not monitor accesses to I/O regions or accesses to cacheable regions when I-Cache is turned off.
1	12	Cycles waiting for D-Cache fill data	Number of cycles waiting for the return of the critical word of D-Cache misses from the system bus. This event selector does not monitor accesses to I/O regions or accesses to cacheable regions when D-Cache is turned off.
1	13	Uncached fetch data access from bus	Number of accesses of uncached instruction data returning from the system bus. This event selector monitors accesses to I/O regions or accesses to cacheable regions when I-Cache is not configured or off.
1	14	Uncached load data access from bus	Number of accesses of uncached load data returning from the system bus. This event selector monitors accesses to I/O regions or accesses to cacheable regions when D-Cache is not configured or off.
1	15	Cycles waiting for uncached fetch data from bus	Number of cycles waiting for the instruction data to return from the system bus. This event selector monitors accesses to I/O regions or accesses to cacheable regions when I-Cache is not configured or off.
1	16	Cycles waiting for uncached load data from bus	Number of cycles waiting for the load data to return from the system bus. This event selector monitors accesses to I/O regions or accesses to cacheable regions when D-Cache is not configured or off.
1	23	Hardware prefetch bus access	This event counts the bus accesses generated by the hardware data prefetcher.
2	0	Misprediction of conditional branches (direction)	Number of misprediction of committed conditional branches.
2	1	Misprediction of taken conditional branches (direction)	Number of misprediction of committed taken conditional branches.
2	2	Misprediction of targets of Return instructions	Number of misprediction of committed Return instruction.

Table 4: Event Selectors

3.2.30 MHPMEVENT4 (0x324) (standard read/write)

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3
RISC-V PROCESSORS

3 CONTROL STATUS REGISTERS FOR

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																						SEL					TYPE				
N/A																						RW					RW				
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	

MHPMEVENT4 [31:0]

Bits	Name	Description
8-4	SEL	Event Selectors table
3-0	TYPE	Event Selectors table

MHPMEVENT4 Bits

3.2.31 MHPMEVENT5 (0x325) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																							SEL					TYPE			
N/A																							RW					RW			
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	

MHPMEVENT5 [31:0]

Bits	Name	Description
8-4	SEL	Event Selectors table
3-0	TYPE	Event Selectors table

MHPMEVENT5 Bits

3.2.32 MHPMEVENT6 (0x326) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																						SEL					TYPE				
N/A																						RW					RW				
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	

MHPMEVENT6 [31:0]

Bits	Name	Description
8-4	SEL	Event Selectors table
3-0	TYPE	Event Selectors table

MHPMEVENT6 Bits

3.2.33 MSCRATCH (0x340) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSCRATCH																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MSCRATCH [31:0]

Bits	Name	Description
31-0	MSCRATCH	Scratch register storage.

MSCRATCH Bits

3.2.34 MEPC (0x341) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EPC																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x

MEPC [31:0]

Bits	Name	Description
31-1	EPC	Exception program counter.

MEPC Bits

3.2.35 MCAUSE (0x342) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
INTERRUPT	RSVD																				EXCEPTION_CODE															
RW	N/A																				RW															
0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0					

MCAUSE [31:0]

Bits	Name	Description
31	INTERRUPT	Interrupt
11-0	EXCEPTION_CODE	Exception code When interrupt is 1, the value means: 0:User software interrupt 1:Supervisor software interrupt 3:Machine software interrupt 4:User timer interrupt 5:Supervisor timer interrupt 7:Machine timer interrupt 8:User external interrupt 9:Supervisor external interrupt 11:Machine external interrupt 16:Imprecise ECC error interrupt (slave port accesses, D-Cache evictions, and nonblocking load/stores) (M-mode) 17:Bus read/write transaction error interrupt (M-mode) 18:Performance monitor overflow interrupt (M-mode) 256+16:Imprecise ECC error interrupt (slave port accesses, D-Cache evictions, and nonblocking load/stores) (S-mode) 256+17:Bus write transaction error interrupt (S-mode) 256+18:Performance monitor overflow interrupt (S-mode) When interrupt bit is 0, the value means: 0:Instruction address misaligned 1:Instruction access fault 2:Illegal instruction 3:Breakpoint 4:Load address misaligned 5:Load access fault 6:Store/AMO address misaligned 7:Store/AMO access fault 8:Environment call from U-mode 9:Environment call from S-mode 11:Environment call from M-mode 32:Stack overflow exception 33:Stack underflow exception 40-47:Reserved

MCAUSE Bits

3.2.36 MTVAL (0x343) (standard read/write)

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3
RISC-V PROCESSORS

3 CONTROL STATUS REGISTERS FOR

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MTVAL																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MTVAL [31:0]

Bits	Name	Description
31-0	MTVAL	Exception-specific information for software trap handling.

MTVAL Bits

3.2.37 MIP (0x344) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD													PMOVI	BWEI	IMECCI	RSVD				MEIP	RSVD	SEIP	UEIP	MTIP	RSVD	STIP	UTIP	MSIP	RSVD	SSIP	USIP
N/A													RW	RW	RW	N/A				RW	N/A	RW	RW	RW	N/A	RW	RW	RW	N/A	RW	RW
x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	x	x	x	x	0	x	0	0	0	x	0	0	0	x	0	0

MIP [31:0]

Bits	Name	Description
18	PMOVI	Performance monitor overflow local interrupt pending bit. 0:Not pending 1:Pending
17	BWEI	Bus read/write transaction error local interrupt pending bit. The processor may receive bus errors on load/store instructions or cache writebacks. 0:Not pending 1:Pending
16	IMECCI	Imprecise ECC error local interrupt enable bit. The processor may receive imprecise ECC errors on slave port accesses or cache writebacks. 0:Not pending 1:Pending
11	MEIP	M mode external interrupt pending bit. 0:Not pending 1:Pending
9	SEIP	S mode external interrupt pending bit. 0:Not pending 1:Pending

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3
RISC-V PROCESSORS

3 CONTROL STATUS REGISTERS FOR

Bits	Name	Description
8	UEIP	U mode external interrupt pending bit. 0:Not pending 1:Pending
7	MTIP	M mode timer interrupt pending bit. 0:Not pending 1:Pending
5	STIP	S mode timer interrupt pending bit. 0:Not pending 1:Pending
4	UTIP	U mode timer interrupt pending bit 0:Not pending 1:Pending
3	MSIP	M mode software interrupt pending bit. 0:Not pending 1:Pending
1	SSIP	S mode software interrupt pending bit. 0:Not pending 1:Pending
0	USIP	U mode software interrupt pending bit. 0:Not pending 1:Pending

MIP Bits

3.2.38 PMPCFG0 (0x3A0) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMP3CFG								PMP2CFG								PMP1CFG								PMP0CFG							
RW								RW								RW								RW							
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PMPCFG0 [31:0]

Bits	Name	Description
31-24	PMP3CFG	See PMPCFG register
23-16	PMP2CFG	See PMPCFG register
15-8	PMP1CFG	See PMPCFG register
7-0	PMP0CFG	See PMPCFG register

PMPCFG0 Bits

3.2.39 PMPCFG1 (0x3A1) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMP7CFG								PMP6CFG								PMP5CFG								PMP4CFG							
RW								RW								RW								RW							
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

PMPCFG1 [31:0]

Bits	Name	Description
31-24	PMP7CFG	See PMPCFG register
23-16	PMP6CFG	See PMPCFG register
15-8	PMP5CFG	See PMPCFG register
7-0	PMP4CFG	See PMPCFG register

PMPCFG1 Bits

3.2.40 PMPCFG2 (0x3A2) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMP11CFG								PMP10CFG								PMP9CFG								PMP8CFG							
RW								RW								RW								RW							
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

PMPCFG2 [31:0]

Bits	Name	Description
31-24	PMP11CFG	See PMPCFG register
23-16	PMP10CFG	See PMPCFG register
15-8	PMP9CFG	See PMPCFG register
7-0	PMP8CFG	See PMPCFG register

PMPCFG2 Bits

3.2.41 PMPCFG3 (0x3A3) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMP15CFG								PMP14CFG								PMP13CFG								PMP12CFG							
RW								RW								RW								RW							

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3
RISC-V PROCESSORS

3 CONTROL STATUS REGISTERS FOR

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PMPCFG3 [31:0]

Bits	Name	Description
31-24	PMP15CFG	See PMPCFG register
23-16	PMP14CFG	See PMPCFG register
15-8	PMP13CFG	See PMPCFG register
7-0	PMP12CFG	See PMPCFG register

PMPCFG3 Bits

7	6	5	4	3	2	1	0
L	RSVD		A	X	W	R	
W1S*	N/A		RW	RW	RW	RW	
0	x	x	0	0	0	0	0

Table 5: PMPCFG [7:0]

Bits	Name	Description
7	L	Write lock and permission enforcement bit for Machine mode 0: Machine mode writes to PMP entry registers are allowed. R/W/X permissions apply to S and U modes. 1: For PMP entry i, writes to PMPiCFG and PMPADDRi are ignored. Additionally, if PMPiCFG.A is set to TOR, writes to pmpaddri-1 are ignored as well. As for permission enforcement, R/W/X permissions apply to all modes. This bit can only be cleared to 0 with a system reset.
4-3	A	Address matching mode. 0: OFF: Null region. 1: TOR: Top of range. For PMP entry 0, it matches any address $A < \text{pmpaddr0}$. For PMP entry i, it matches any address A such that $\text{pmpaddri} > A \geq \text{pmpaddri-1}$. But the 4-byte range is not supported. 2: Reserved. 3: NAPOT: Naturally aligned power-of-2 region. This mode makes use of the low-order bits of the associated address register to encode the size of the range. The minimal size of NAPOT regions must be 8 bytes.
2	X	Instruction execution control. 0: Instruction execution is not allowed 1: Instruction execution is allowed.

Bits	Name	Description
1	W	Write access control. 0: Write accesses are not allowed. 1: Write accesses are allowed.
0	R	Read access control. 0: Read accesses are not allowed. 1: Read accesses are allowed.

Table 6: PMPCFG Bit Fields

3.2.42 PMPADDR[PMPADDR0] (0x3B0 + 0x1 * n) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMPADDR_31_2																															RSVD
RW																															N/A
*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	x	x

PMPADDR [31:0]

Bits	Name	Description
31-2	PMPADDR_31_2	Register Content : Match Size(Byte) aaaa. . . aaa0 8 aaaa. . . aa01 16 aaaa. . . a011 32 aa01. . . 1111 2 ^{XLEN} a011. . . 1111 2 ^{XLEN+1} 0111. . . 1111 2 ^{XLEN+2} 1111. . . 1111 2 ^{XLEN+3*1}

PMPADDR Bits

3.2.43 TSELECT (0x7A0) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRIGGER_INDEX																RW															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TSELECT [31:0]

Bits	Name	Description
31-0	TRIGGER_INDEX	This register determines which trigger is accessible through other trigger registers.

TSELECT Bits

3.2.44 TDATA1 (0x7A1) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TYPE				DMODE	DATA																										
RW				RW	RW																										
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

TDATA1 [31:0]

Bits	Name	Description
31-28	TYPE	Indicates the trigger type. 0:The selected trigger is invalid. 2:The selected trigger is an address/data match trigger. 3:The selected trigger is an instruction count trigger 4:The selected trigger is an interrupt trigger. 5:The selected trigger is an exception trigger.
27	DMODE	Setting this field to indicate the trigger is used by Debug Mode. 0:Both Debug-mode and M-mode can write the currently selected trigger registers. 1:Only Debug Mode can write the currently selected trigger registers. Writes from M-mode is ignored.
26-0	DATA	Trigger-specific data

TDATA1 Bits

3.2.45 MCONTROL (0x7A1) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
TYPE				DMODE	MASKMAX						RSVD				ACTION				CHAIN	MATCH				M	RSVD	S	U	EXECUTE	STORE	LOAD				
RW				RW	RO						N/A				RW				RW	RW				RW	N/A	RW	RW	RW	RW	RW	RW	RW	RW	RW
0	0	1	0	0	0	1	0	0	1	0	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	x	0	0	0	0	0			

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

MCONTROL [31:0]

Bits	Name	Description
31-28	TYPE	Indicates the trigger type. 0:The selected trigger is invalid. 2:The selected trigger is an address/data match trigger.
27	DMODE	Setting this field to indicate the trigger is used by Debug Mode. 0:Both Debug-mode and M-mode can write the currently selected trigger registers 1:Only Debug Mode can write the currently selected trigger registers. Writes from M-mode is ignored.
26-21	MASKMAX	Indicates the largest naturally aligned range supported by the hardware is 2^{12} bytes.
15-12	ACTION	Setting this field to select what happens when this trigger matches. 0:Raise a breakpoint exception 1:Enter Debug Mode. (Only supported when DMODE is 1.)
11	CHAIN	Setting this field to enable trigger chain. 0:When this trigger matches, the configured action is taken. 1:While this trigger does not match, it prevents the trigger with the next index from matching. If Number of Triggers is 2, this field is hardwired to 0 on trigger 1 (tselect = 1). If Number of Triggers is 4, this field is hardwired to 0 on trigger 3 (tselect = 3). If Number of Triggers is 8, this field is hardwired to 0 on trigger 3 and trigger 7 (tselect = 3 or 7).
10-7	MATCH	Setting this field to select the matching scheme. 0:Matches when the value equals tdata2. 1:Matches when the top M bits of the value match the top M bits of tdata2. M is 31 minus the index of the least-significant bit containing 0 in tdata2. 2:Matches when the value is greater than (unsigned) or equal to tdata2. 3:Matches when the value is less than (unsigned) tdata2
6	M	Setting this field to enable this trigger in M-mode.
4	S	Setting this field to enable this trigger in S-mode.
3	U	Setting this field to enable this trigger in U-mode.
2	EXECUTE	Setting this field to enable this trigger to compare virtual address of an instruction.
1	STORE	Setting this field to enable this trigger to compare virtual address of a store.

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3
RISC-V PROCESSORS

3 CONTROL STATUS REGISTERS FOR

Bits	Name	Description
0	LOAD	Setting this field to enable this trigger to compare virtual address of a load.

MCONTROL Bits

3.2.46 ICOUNT (0x7A1) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TYPE				DMODE	RSVD																COUNT	M	RSVD	S	U	ACTION					
RW				RW	N/A																RO	RW	N/A	RW	RW	RW					
0	0	1	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	0	x	0	0	0	0	0	0	0	0

ICOUNT [31:0]

Bits	Name	Description
31-28	TYPE	The selected trigger is an instruction count trigger.
27	DMODE	Setting this field to indicate the trigger is used by Debug Mode. 0:Both Debug-mode and M-mode can write the currently selected trigger registers. 1:Only Debug Mode can write the currently selected trigger registers. Writes from M-mode is ignored.
10	COUNT	This field is hardwired to 1 for single-stepping support
9	M	Setting this field to enable this trigger in M-mode.
7	S	Setting this field to enable this trigger in S-mode.
6	U	Setting this field to enable this trigger in U-mode.
5-0	ACTION	Setting this field to select what happens when this trigger matches. 0:Raise a breakpoint exception 1:Enter Debug Mode. (Only supported when DMODE is 1.)

ICOUNT Bits

3.2.47 ITRIGGER (0x7A1) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TYPE				DMODE	RSVD																M	RSVD	S	U	ACTION						
RW				RW	N/A																RW	N/A	RW	RW	RW						
0	0	1	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	x	0	0	0	0	0	0	0	0

ITRIGGER [31:0]

Bits	Name	Description
31-28	TYPE	The selected trigger is an interrupt trigger.
27	DMODE	Setting this field to indicate the trigger is used by Debug Mode. 0:Both Debug-mode and M-mode can write the currently selected trigger registers. 1:Only Debug Mode can write the currently selected trigger registers. Writes from M-mode is ignored.
9	M	Setting this field to enable this trigger in M-mode.
7	S	Setting this field to enable this trigger in S-mode.
6	U	Setting this field to enable this trigger in U-mode.
5-0	ACTION	Setting this field to select what happens when this trigger matches. 0:Raise a breakpoint exception. 1:Enter Debug Mode. (Only supported when DMODE is 1.)

ITRIGGER Bits

3.2.48 ETRIGGER (0x7A1) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TYPE				DMODE	RSVD																NMI	M	RSVD	S	U	ACTION					
RW				RW	N/A																RW	RW	N/A	RW	RW	RW					
0	0	1	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	x	0	0	0	0	0	0	0	0

ETRIGGER [31:0]

Bits	Name	Description
31-28	TYPE	The selected trigger is an exception trigger.
27	DMODE	Setting this field to indicate the trigger is used by Debug Mode. 0:Both Debug-mode and M-mode can write the currently selected trigger registers. 1:Only Debug Mode can write the currently selected trigger registers. Writes from M-mode is ignored.
10	NMI	Setting this field to enable this trigger in non-maskable interrupts, regardless of the values of s, u, and m.
9	M	Setting this field to enable this trigger in M-mode.
7	S	Setting this field to enable this trigger in S-mode.
6	U	Setting this field to enable this trigger in U-mode.
5-0	ACTION	Setting this field to select what happens when this trigger matches. 0:Raise a breakpoint exception 1:Enter Debug Mode. (Only supported when DMODE is 1.)

Bits	Name	Description

ETRIGGER Bits

3.2.49 TDATA2 (0x7A2) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TDATA2 [31:0]

Bits	Name	Description
31-0	DATA	This register provides accesses to the tdata2 register of the currently selected trigger registers selected by the tselect register, and it holds trigger-specific data..

TDATA2 Bits

3.2.50 TDATA3 (0x7A3) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TDATA3 [31:0]

Bits	Name	Description
31-0	DATA	This register provides accesses to the tdata3 register of the currently selected trigger registers selected by the tselect register, and it holds trigger-specific data..

TDATA3 Bits

3.2.51 TEXTRA (0x7A3) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MVALUE						MSELECT	RSVD															SVALUE						SSELECT			

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3
RISC-V PROCESSORS

3 CONTROL STATUS REGISTERS FOR

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RW						RW	N/A														RW						RW				
0	0	0	0	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0

TEXTRA [31:0]

Bits	Name	Description
31-26	MVALUE	Data used together with MSELECT.
25	MSELECT	0:Ignore MVALUE. 1:This trigger will only match if the lower bits of mcontext equal MVALUE.
10-2	SVALUE	Data used together with SSELECT.
1-0	SSELECT	0:Ignore MVALUE 1:This trigger will only match if the lower bits of scontext equal SVALUE 2This trigger will only match if satp.ASID equals SVALUE.

TEXTRA Bits

3.2.52 TINFO (0x7A4) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																INFO															
N/A																RO															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

TINFO [31:0]

Bits	Name	Description
15-0	INFO	One bit for each possible type in tdata1. Bit N corresponds to type N. If the bit is set, then that type is supported by the currently selected trigger. If the currently selected trigger does not exist, this field contains 1. 0:When this bit is set, there is no trigger at this tselect 1:Reserved and hardwired to 0. 2: When this bit is set, the selected trigger supports type of address/data match trigger 3:When this bit is set, the selected trigger supports type of instruction count trigger. 4:When this bit is set, the selected trigger supports type of interrupt trigger 5:When this bit is set, the selected trigger supports type of exception trigger 15:When this bit is set, the selected trigger exists (so enumeration shouldn't terminate), but is not currently available. Others:Reserved for future use.

TINFO Bits

3.2.53 TCONTROL (0x7A5) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								MPTE	RSVD			MTE	RSVD		
N/A																								RW	N/A			RW	N/A		
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	x	x	x	0	x	x	x	

TCONTROL [31:0]

Bits	Name	Description
7	MPTE	M-mode previous trigger enable field. When a trap into M-mode is taken, MPTE is set to the value of MTE.
3	MTE	M-mode trigger enable field. When a trap into M-mode is taken, MTE is set to 0. When the MRET instruction is executed, MTE is set to the value of MPTE. 0:Triggers do not match/fire while the hart is in M-mode. 1:Triggers do match/fire while the hart is in M-mode.

TCONTROL Bits

3.2.54 MCONTEXT (0x7A8) (standard read/write)

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3
RISC-V PROCESSORS

3 CONTROL STATUS REGISTERS FOR

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																										MCONTEXT					
N/A																										RW					
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0

MCONTEXT [31:0]

Bits	Name	Description
5-0	MCONTEXT	Machine mode software can write a context number to this register, which can be used to set triggers that only fire in that specific context.

MCONTEXT Bits

3.2.55 SCONTEXT (0x7AA) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																							SCONTEXT								
N/A																							RW								
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	

SCONTEXT [31:0]

Bits	Name	Description
8-0	SCONTEXT	Machine mode software can write a context number to this register, which can be used to set triggers that only fire in that specific context.

SCONTEXT Bits

3.2.56 DCSR (0x7B0) (debug-mode-only)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XDEBUGVER				RSVD												EBREAKM	RSVD	EBREAKS	EBREAKU	STEPIE	STOPCOUNT	STOPTIME	CAUSE			RSVD	MPRVEN	NMIP	STEP	PRV	
																RW	N/A	RW	RW	RW	RW	RW				RW	RO	N/A	RW	RO	RW
0	1	0	0	x	x	x	x	x	x	x	x	x	x	x	x	0	x	0	0	0	1	1	0	0	0	x	0	0	0	1	1

DCSR [31:0]

Bits	Name	Description
31-28	XDEBUGVER	Version of the external debugger. 0 indicates that no external debugger exists and 4 indicates that the external debugger conforms to the RISC-V External Debug Support (TD003) V0.13
15	EBREAKM	This bit controls the behavior of EBREAK instructions in Machine Mode 0:Generate a regular breakpoint exception 1:Enter Debug Mode
13	EBREAKS	This bit controls the behavior of EBREAK instructions in Supervisor Mode. 0:Generate a regular breakpoint exception 1:Enter Debug Mode
12	EBREAKU	This bit controls the behavior of EBREAK instructions in User/Application Mode 0:Generate a regular breakpoint exception 1:Enter Debug Mode
11	STEPIE	This bit controls whether interrupts are enabled during single stepping 0:Disable interrupts during single stepping 1:Allow interrupts in single stepping
10	STOPCOUNT	This bit controls whether performance counters are stopped in Debug Mode. 0:Do not stop counters in Debug Mode 1:Stop counters in Debug Mode
9	STOPTIME	This bit controls whether timers are stopped in Debug Mode. The processor only drives its stoptime output pin to 1 if it is in Debug Mode and this bit is set. Integration effort is required to make timers in the platform observe this pin to really stop them. 0:Do not stop timers in Debug Mode 1:Stop timers in Debug Mode
8-6	CAUSE	Reason why Debug Mode was entered. When there are multiple reasons to enter Debug Mode, the priority to determine the CAUSE value will be: trigger module > EBREAK > halt-on-reset > halt request > single step. Halt requests are requests issued by the external debugger 0:Reserved 1:EBREAK 2:Trigger module 3:Halt request 4:Single step 5:Halt-on-reset 6-7:Reserved

Bits	Name	Description
4	MPRVEN	This bit controls whether mstatus.MPRV takes effect in Debug Mode. 0:MPRV in mstatus is ignored in Debug Mode. 1:MPRV in mstatus takes effect in Debug Mode.
3	NMIP	When this bit is set, there is a Non-Maskable-Interrupt (NMI) pending for the hart. Since an NMI can indicate a hardware error condition, reliable debugging may no longer be possible once this bit becomes set.
2	STEP	This bit controls whether non-Debug Mode instruction execution is in the single step mode. When set, the hart returns to Debug Mode after a single instruction execution. If the instruction does not complete due to an exception, the hart will immediately enter Debug Mode before executing the trap handler, with appropriate exception registers set. 0:Single Step Mode is off 1:Single Step Mode is on
1-0	PRV	The privilege level that the hart was operating in when Debug Mode was entered. The external debugger can modify this value to change the hart's privilege level when exiting Debug Mode. 0:User/Application 1:Supervisor 2:Reserved 3:Machine

DCSR Bits

3.2.57 DPC (0x7B1) (debug-mode-only)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DPC																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DPC [31:0]

Bits	Name	Description
31-0	DPC	Debug Program Counter. Bit 0 is hardwired to 0.

DPC Bits

3.2.58 DSCRATCH0 (0x7B2) (debug-mode-only)

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3
RISC-V PROCESSORS

3 CONTROL STATUS REGISTERS FOR

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSCRATCH																															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

DSCRATCH0 [31:0]

Bits	Name	Description
31-0	DSCRATCH	A scratch register that is reserved for use by Debug Module.

DSCRATCH0 Bits

3.2.59 DSCRATCH1 (0x7B3) (debug-mode-only)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSCRATCH																															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DSCRATCH1 [31:0]

Bits	Name	Description
31-0	DSCRATCH	A scratch register that is reserved for use by Debug Module.

DSCRATCH1 Bits

3.2.60 MILMB (0x7C0) (non-standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IBPA																						RSVD						RWECC	ECCEN	IEN	
RO																						N/A						RW	RW	RO	
*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	x	x	x	x	x	x	0	0	0	1

MILMB [31:0]

Bits	Name	Description
31-10	IBPA	The base physical address of ILM. It has to be an integer multiple of the ILM size

Bits	Name	Description
3	RWECC	Controls diagnostic accesses of ECC codes of the ILM RAMs. When set, load/store to ILM reads/writes ECC codes to the mecc_code register. This bit can be set for injecting ECC errors to test the ECC handler. 0:Disable diagnostic accesses of ECC codes 1:Enable diagnostic accesses of ECC codes
2-1	ECCEN	Parity/ECC enable control: 0:Disable parity/ECC 1:Reserved 2:Generate exceptions only on uncorrectable parity/ECC errors 3:Generate exceptions on any type of parity/ECC errors
0	IEN	ILM enable control: 0:ILM is disabled 1:ILM is enabled

MILMB Bits

3.2.61 MDLMB (0x7C1) (non-standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBPA										RSVD										RWECC		ECCEN		DEN							
RO										N/A										RW		RW		RO							
*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	x	x	x	x	x	x	0	0	0	1

MDLMB [31:0]

Bits	Name	Description
31-10	DBPA	The base physical address of DLM. It has to be an integer multiple of the DLM size
3	RWECC	Controls diagnostic accesses of ECC codes of the DLM RAMs. When set, load/store to DLM reads/writes ECC codes to the mecc_code register. This bit can be set for injecting ECC errors to test the ECC handler. 0:Disable diagnostic accesses of ECC codes 1:Enable diagnostic accesses of ECC codes
2-1	ECCEN	Parity/ECC enable control: 0:Disable parity/ECC 1:Reserved 2:Generate exceptions only on uncorrectable parity/ECC errors 3:Generate exceptions on any type of parity/ECC errors

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3
RISC-V PROCESSORS

3 CONTROL STATUS REGISTERS FOR

Bits	Name	Description
0	DEN	DLM enable control: 0:DLM is disabled 1:DLM is enabled

MDLMB Bits

3.2.62 MECC_CODE (0x7C2) (non-standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD									INSN	RAMID				P	C	RSVD									CODE						
N/A									RO	RO				RO	RO	N/A									RW						
x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	1

MECC_CODE [31:0]

Bits	Name	Description
22	INSN	Indicates if the parity/ECC error is caused by instruction fetch or data access. 0:Data access 1:Instruction fetch
21-18	RAMID	The ID of RAM that caused parity/ECC errors. This bit is updated on parity/ECC error exceptions. 0-1:Reserved 2:Tag RAM of I-Cache 3:Data RAM of I-Cache 4:Tag RAM of D-Cache 5:Data RAM of D-Cache 6:Tag RAM of TLB 7:Data RAM of TLB 8:ILM 9:DLM 10-15:Reserved
17	P	Precise error. This bit is updated on parity/ECC error exceptions. 0:Imprecise error 1:Precise error
16	C	Correctable error. This bit is updated on parity/ECC error exceptions. 0:Uncorrectable error 1:Correctable error

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3
RISC-V PROCESSORS

3 CONTROL STATUS REGISTERS FOR

Bits	Name	Description
6-0	CODE	This field records the ECC value on ECC error exceptions. This field is also used to read/write the ECC codes when diagnostic access of ECC codes are enabled (mlmb.RWECC or mdlmb.RWECC is 1).

MECC_CODE Bits

3.2.63 MNVEC (0x7C3) (non-standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																MNVEC															
RO																															
*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

MNVEC [31:0]

Bits	Name	Description
31-0	MNVEC	Base address of the NMI handler. Its value is the zero-extended value of the reset_vector.

MNVEC Bits

3.2.64 MXSTATUS (0x7C4) (non-standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																RSVD															
																N/A															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	x	x	0	0

MXSTATUS [31:0]

Bits	Name	Description
5	PDME	For saving previous DME state on entering a trap. This field is hardwired to 0 if data cache and data local memory are not supported.
4	DME	Data Machine Error flag. It indicates an exception occurred at the data cache or data local memory (DLM). Load/store accesses will bypass D-Cache when this bit is set. The exception handler should clear this bit after the machine error has been dealt with.

Bits	Name	Description
1	PPFT_EN	When mcause is imprecise exception (in the form of an interrupt), the PM field records the privileged mode of the instruction that caused the imprecise exception. The PM field encoding is defined as follows: 0: User mode 1: Supervisor mode 2: Reserved 3: Machine mode
0	PFT_EN	Enable performance throttling. When throttling is enabled, the processor executes instructions at the performance level specified in mpft_ctl.T_LEVEL. On entering a trap: PPFT_EN <= PFT_EN; PFT_EN <= mpft_ctl.FAST_INT ? 0 : PFT_EN; On executing an MRET instruction: PFT_EN <= PPFT_EN; This field is hardwired to 0 if the PowerBrake feature is not supported.

MXSTATUS Bits

3.2.65 MPFT_CTL (0x7C5) (non-standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																							FAST_INT	T_LEVEL				RSVD			
N/A																							RW	RW				N/A			
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	x	x	x	x

MPFT_CTL [31:0]

Bits	Name	Description
8	FAST_INT	Fast interrupt response. If this field is set, mxstatus.PFT_EN will be automatically cleared when the processor enters an interrupt handler.
7-4	T_LEVEL	Throttling Level. The processor has the highest performance at throttling level 0 and the lowest performance at throttling level 15. 0:Level 0 (the highest performance) 1-14:Level 1-14 15:Level 15 (the lowest performance)

MPFT_CTL Bits

3.2.66 MHSP_CTL (0x7C6) (non-standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																										M	S	U	SCHM	UDF_EN	OVF_EN
N/A																										RW	RW	RW	RW	RW	RW
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0

MHSP_CTL [31:0]

Bits	Name	Description
5	M	Enables the SP protection and recording mechanism in Machine mode 0:The mechanism is disabled in Machine mode. 1: The mechanism is enabled in Machine mode.
4	S	Enables the SP protection and recording mechanism in Supervisor mode 0:The mechanism is disabled in Supervisor mode 1:The mechanism is enabled in Supervisor mode
3	U	Enables the SP protection and recording mechanism in User mode 0:The mechanism is disabled in User mode 1:The mechanism is enabled in User mode.
2	SCHM	Selects the operating scheme of the stack protection and recording mechanism 0:Stack overflow/underflow detection 1:Top-of-stack recording
1	UDF_EN	Enable bit for the stack underflow protection mechanism. This bit will be cleared to 0 automatically by hardware when a stack protection (overflow or underflow) exception is taken. 0:The stack underflow protection is disabled 1:The stack underflow protection is enabled.
0	OVF_EN	Enable bit for the stack overflow protection and recording mechanism. This bit will be cleared to 0 automatically by hardware when a stack protection (overflow or underflow) exception is taken. 0:The stack overflow protection and recording mechanism are disabled. 1:The stack overflow protection and recording mechanism are enabled.

MHSP_CTL Bits

3.2.67 MSP_BOUND (0x7C7) (non-standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																MSP_BOUND															
																RW															
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	

MSP_BOUND [31:0]

Bits	Name	Description
31-0	MSP_BOUND	Machine SP Bound

MSP_BOUND Bits

3.2.68 MSP_BASE (0x7C8) (non-standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SP_BASE																															
RW																															
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	

MSP_BASE [31:0]

Bits	Name	Description
31-0	SP_BASE	Machine SP base

MSP_BASE Bits

3.2.69 MDCAUSE (0x7C9) (non-standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																										PM		RSVD		MDCAUSE	
																										RW		N/A		RW	
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	x	x	0	0	0

MDCAUSE [31:0]

Bits	Name	Description
6-5	PM	When mcause is imprecise exception (in the form of an interrupt), the PM field records the privileged mode of the instruction that caused the imprecise exception. The PM field encoding is defined as follows: 0: User mode 1: Supervisor mode 2: Reserved 3: Machine mode
2-0	MDCAUSE	This register further disambiguates causes of traps recorded in the mcause register. The value of MDCAUSE for precise exception: When mcause == 1 (Instruction access fault): 0:Reserved; 1:ECC/Parity error; 2:PMP instruction access violation; 3:Bus error; 4:PMA empty hole access When mcause == 2 (Illegal instruction): 0:Please parse the mtval CSR; 1:FP disabled exception; 2:ACE disabled exception When mcause == 5 (Load access fault): 0:Reserved; 1:ECC/Parity error; 2:PMP load access violation; 3:Bus error; 4:Misaligned address; 5:PMA empty hole access; 6:PMA attribute inconsistency; 7:PMA NAMO exception When mcause == 7 (Store access fault): 0:Reserved; 1:ECC/Parity error; 2:PMP load access violation; 3:Bus error; 4:Misaligned address; 5:PMA empty hole access; 6:PMA attribute inconsistency; 7:PMA NAMO exception The value of MDCAUSE for imprecise exception: When mcause == Local Interrupt 16 or Local Interrupt 272 (16 + 256) (ECC error local interrupt) 0:Reserved; 1:LM slave port ECC/Parity error; 2:Imprecise store ECC/Parity error; 3:Imprecise load ECC/Parity error When mcause == Local Interrupt 17 or Local Interrupt 273 (17 + 256) (Bus read/write transaction error local interrupt) 0:Reserved; 1:Bus read error; 2:Bus write error; 3:PMP error caused by load instructions; 4:PMP error caused by store instructions; 5:PMA error caused by load instructions; 6:PMA error caused by store instructions

MDCAUSE Bits

3.2.70 MCACHE_CTL (0x7CA) (non-standard read/write)

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3
RISC-V PROCESSORS

3 CONTROL STATUS REGISTERS FOR

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																	DC_WAROUND		DC_FIRST_WORD	IC_FIRST_WORD	DPREF_EN	IPREF_EN	CCTL_SUEN	DC_RWECC	IC_RWECC	DC_ECCEN		IC_ECCEN		DC_EN	IC_EN
N/A																	RW		RO	RO	RW	RW	RW	RW	RW	RW		RW		RW	RW
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	1	1	0	0	0	0	0	0	0	0	0	0		

MCACHE_CTL [31:0]

Bits	Name	Description
14-13	DC_WAROUND	Cache Write-Around threshold 0:Disables streaming. All cacheable write misses allocate a cache line according to PMA settings. 1:Stop allocating D-Cache entries regardless of PMA settings after consecutive stores to 4 cache lines. 2:Stop allocating D-Cache entries regardless of PMA settings after consecutive stores to 64 cache lines. 3:Stop allocating D-Cache entries regardless of PMA settings after consecutive stores to 128 cache lines.
12	DC_FIRST_WORD	Cache miss allocation filling policy 0:Cache line data is returned critical (double) word first 1:Cache line data is returned the lowest address (double) word first
11	IC_FIRST_WORD	Cache miss allocation filling policy 0:Cache line data is returned critical (double) word first 1:Cache line data is returned the lowest address (double) word first
10	DPREF_EN	This bit controls hardware prefetch for load/store accesses to cacheable memory regions when D-Cache size is not 0 0:Disable hardware prefetch on load/store memory accesses 1:Enable hardware prefetch on load/store memory accesses
9	IPREF_EN	This bit controls hardware prefetch for instruction fetches to cacheable memory regions when Cache size is not 0 0:Disable hardware prefetch on instruction fetches 1:Enable hardware prefetch on instruction fetches
8	CCTL_SUEN	Enable bit for Superuser-mode and User-mode software to access ucctlbeginaddr and ucctlcommand CSRs 0:Disable ucctlbeginaddr and ucctlcommand accesses in S/U mode 1:Enable ucctlbeginaddr and ucctlcommand accesses in S/U mode

Bits	Name	Description
7	DC_RWECC	Controls diagnostic accesses of ECC codes of the data cache RAMs. It is set to enable CCTL operations to access the ECC codes. This bit can be set for injecting ECC errors to test the ECC handler 0:Disable diagnostic accesses of ECC codes 1:Enable diagnostic accesses of ECC codes
6	IC_RWECC	Controls diagnostic accesses of ECC codes of the instruction cache RAMs. It is set to enable CCTL operations to access the ECC codes . This bit can be set for injecting ECC errors to test the ECC handler. 0:Disable diagnostic accesses of ECC codes 1:Enable diagnostic accesses of ECC codes
5-4	DC_ECCEN	Parity/ECC error checking enable control for the data cache. 0:Disable parity/ECC 1:Reserved 2:Generate exceptions only on uncorrectable parity/ECC errors 3:Generate exceptions on any type of parity/ECC errors
3-2	IC_ECCEN	Parity/ECC error checking enable control for the instruction cache 0:Disable parity/ECC 1:Reserved 2:Generate exceptions only on uncorrectable parity/ECC errors 3:Generate exceptions on any type of parity/ECC errors
1	DC_EN	Controls if the data cache is enabled or not. 0:D-Cache is disabled 1:D-Cache is enabled
0	IC_EN	Controls if the instruction cache is enabled or not. 0:I-Cache is disabled 1:I-Cache is enabled

MCACHE_CTL Bits

3.2.71 MCCTLBEGINADDR (0x7CB) (non-standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MCCTLBEGINADDR [31:0]

Bits	Name	Description
31-0	VA	This register holds the address information required by CCTL operations

MCCTLBEGINADDR Bits

3.2.72 MCCTLCOMMAND (0x7CC) (non-standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSD																									VA						
N/A																									RW						
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	

MCCTLCOMMAND [31:0]

Bits	Name	Description
4-0	VA	See CCTL Command Definition Table

MCCTLCOMMAND Bits

Value (DEC)	Value (BIN)	Command	Type
0	0b00_000	L1D_VA_INVALID	VA
1	0b00_001	L1D_VA_WB	VA
2	0b00_010	L1D_VA_WBINVAL	VA
3	0b00_011	L1D_VA_LOCK	VA
4	0b00_100	L1D_VA_UNLOCK	VA
6	0b00_110	L1D_WBINVAL_ALL	-
7	0b00_111	L1D_WB_ALL	-
8	0b01_000	L1I_VA_INVALID	VA
11	0b01_011	L1I_VA_LOCK	VA
12	0b01_100	L1I_VA_UNLOCK	VA
16	0b10_000	L1D_IX_INVALID	Index
17	0b10_001	L1D_IX_WB	Index
18	0b10_010	L1D_IX_WBINVAL	Index
19	0b10_011	L1D_IX_RTAG	Index
20	0b10_100	L1D_IX_RDATA	Index
21	0b10_101	L1D_IX_WTAG	Index
22	0b10_110	L1D_IX_WDATA	Index
23	0b10_111	L1D_INVALID_ALL	-
24	0b11_000	L1I_IX_INVALID	Index
27	0b11_011	L1I_IX_RTAG	Index
28	0b11_100	L1I_IX_RDATA	Index

Value (DEC)	Value (BIN)	Command	Type
29	0b11_101	L1I_IX_WTAG	Index
30	0b11_110	L1I_IX_WDATA	Index

Table 7: CCTL Command Definition

3.2.73 MCCTLDATA (0x7CD) (non-standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																												VA			
N/A																												RW			
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	

MCCTLDATA [31:0]

Bits	Name	Description
4-0	VA	See CCTL Commands Which Access mcctldata Table

MCCTLDATA Bits

Value (DEC)	Value (BIN)	Command	Type
3	0b00_011	L1D_VA_LOCK	VA
11	0b01_011	L1I_VA_LOCK	VA
19	0b10_011	L1D_IX_RTAG	Index
20	0b10_100	L1D_IX_RDATA	Index
21	0b10_101	L1D_IX_WTAG	Index
22	0b10_110	L1D_IX_WDATA	Index
27	0b11_011	L1I_IX_RTAG	Index
28	0b11_100	L1I_IX_RDATA	Index
29	0b11_101	L1I_IX_WTAG	Index
30	0b11_110	L1I_IX_WDATA	Index

Table 8: CCTL Commands Which Access mcctldata

3.2.74 MCOUNTERWEN (0x7CE) (non-standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								HPM6	HPM5	HPM4	HPM3	IR	RSVD	CY	
N/A																								RW	RW	RW	RW	RW	N/A	RW	
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	x	0

MCOUNTERWEN [31:0]

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3
RISC-V PROCESSORS

3 CONTROL STATUS REGISTERS FOR

Bits	Name	Description
6	HPM6	See register description
5	HPM5	See register description
4	HPM4	See register description
3	HPM3	See register description
2	IR	See register description
0	CY	See register description

MCOUNTERWEN Bits

3.2.75 MCOUNTERINTEN (0x7CF) (non-standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								HPM6	HPM5	HPM4	HPM3	IR	RSVD	CY	
N/A																								RW	RW	RW	RW	RW	N/A	RW	
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	x	0

MCOUNTERINTEN [31:0]

Bits	Name	Description
6	HPM6	See register description
5	HPM5	See register description
4	HPM4	See register description
3	HPM3	See register description
2	IR	See register description
0	CY	See register description

MCOUNTERINTEN Bits

3.2.76 MMISC_CTL (0x7D0) (non-standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																							NBLD_EN	RSVD	MSA_UNA	RSVD		BRPE	RVCMPM	VEC_PLIC	RSVD
N/A																							RW	N/A	RW	N/A		RW	RW	RW	N/A
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	x	1	x	x	1	0	0	x	

MMISC_CTL [31:0]

Bits	Name	Description
8	NBLD_EN	This field controls the blocking behavior of load instructions. When this bit is clear, load instructions accessing non-device regions are blocking. When this bit is set, load instructions will not be blocking on such occasions and bus errors will no longer be reported synchronously. 0:Load to memory regions are blocking. 1:Load to memory regions are non-blocking.
6	MSA_UNA	This field controls whether the load/store instructions can access misaligned memory locations without generating Address Misaligned exceptions. Supported instructions: LW/LH/LHU/SW/SH 0:Misaligned accesses generate Address Misaligned exceptions. 1:Misaligned accesses generate Address Misaligned exceptions.
3	BRPE	Branch prediction enable bit. This bit controls all branch prediction structures. 0:Disabled 1:Enabled This bit is hardwired to 0 if branch prediction structure is not supported.
2	RVCOMPM	RISC-V compatibility mode enable bit. If the compatibility mode is turned on, all specific instructions become reserved instructions 0:Disabled 1:Enabled
1	VEC_PLIC	Selects the operation mode of PLIC: 0:Regular mode 1:Vector mode Please note that both this bit and the vector mode enable bit (VECTORED) of the Feature Enable Register in NCEPLIC100 should be turned on for the vectored interrupt support to work correctly. This bit is hardwired to 0 if the vectored PLIC feature is not supported.

MMISC_CTL Bits

3.2.77 MCOUNTERMASK_M (0x7D1) (non-standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																									HPM6	HPM5	HPM4	HPM3	IR	RSVD	CY
N/A																									RW	RW	RW	RW	RW	N/A	RW

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3
RISC-V PROCESSORS

3 CONTROL STATUS REGISTERS FOR

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	x	0

MCOUNTERMASK_M [31:0]

Bits	Name	Description
6	HPM6	See register description
5	HPM5	See register description
4	HPM4	See register description
3	HPM3	See register description
2	IR	See register description
0	CY	See register description

MCOUNTERMASK_M Bits

3.2.78 MCOUNTERMASK_S (0x7D2) (non-standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								HPM6	HPM5	HPM4	HPM3	IR	RSVD	CY	
N/A																								RW	RW	RW	RW	RW	N/A	RW	
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	x	0	

MCOUNTERMASK_S [31:0]

Bits	Name	Description
6	HPM6	See register description
5	HPM5	See register description
4	HPM4	See register description
3	HPM3	See register description
2	IR	See register description
0	CY	See register description

MCOUNTERMASK_S Bits

3.2.79 MCOUNTERMASK_U (0x7D3) (non-standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								HPM6	HPM5	HPM4	HPM3	IR	RSVD	CY	
N/A																								RW	RW	RW	RW	RW	N/A	RW	
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	x	0	

MCOUNTERMASK_U [31:0]

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3
RISC-V PROCESSORS

3 CONTROL STATUS REGISTERS FOR

Bits	Name	Description
6	HPM6	See register description
5	HPM5	See register description
4	HPM4	See register description
3	HPM3	See register description
2	IR	See register description
0	CY	See register description

MCOUNTERMASK_U Bits

3.2.80 MCOUNTEROVF (0x7D4) (non-standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								HPM6	HPM5	HPM4	HPM3	IR	RSVD	CY	
N/A																								RW	RW	RW	RW	RW	N/A	RW	
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	x	0	

MCOUNTEROVF [31:0]

Bits	Name	Description
6	HPM6	See register description
5	HPM5	See register description
4	HPM4	See register description
3	HPM3	See register description
2	IR	See register description
0	CY	See register description

MCOUNTEROVF Bits

3.2.81 MSLIDELEG (0x7D5) (non-standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD													PMOVI	BWEI	IMECCI	RSVD															
N/A													RW	RW	RW	N/A															
x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	

MSLIDELEG [31:0]

Bits	Name	Description
18	PMOVI	Delegate S-mode performance monitor overflow local interrupt to S-mode. 0:Do not delegate to S-mode. 1:Delegate to S-mode.
17	BWEI	Delegate S-mode bus read/write transaction error local interrupt to S-mode 0:Do not delegate to S-mode. 1:Delegate to S-mode.
16	IMECCI	Delegate S-mode slave-port ECC error local interrupt to S-mode 0:Do not delegate to S-mode. 1:Delegate to S-mode.

MSLIDELEG Bits

3.2.82 MCLK_CTL (0x7DF) (non-standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
FUNIT																VI	VR	AQ	DQ	UQ	FP	RSVD	CLKGATE									
RW																RW	RW	RW	RW	RW	RW	N/A	RW									
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	x	x	0	0	0	0	0	1	1	1	

MCLK_CTL [31:0]

Bits	Name	Description
31-16	FUNIT	Level 2 clock gating enable for function units listed in the following table. 16:integer arithmetic unit 17:integer permutation unit 18:integer mask unit 19:integer division unit 20:integer multiply and add unit 21:floating-point multiply and add unit 22:floating-point miscellaneous unit 23:floating-point division unit 24:load/store unit 31:25:Reserved
15	VI	Level 1 clock gating enable for the vector/floating-point issue queues.
14	VR	Level 1 clock gating enable for the vector/floating-point register file.

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3
RISC-V PROCESSORS

3 CONTROL STATUS REGISTERS FOR

Bits	Name	Description
13	AQ	Level 1 clock gating enable for ACE load/store queues.
12	DQ	Level 1 clock gating enable for data cache load/store queues.
11	UQ	Level 1 clock gating enable for uncached queues
10	FP	Level 1 clock gating enable for scalar floating point issue unit and queues.
7-0	CLKGATE	One-hot clock gating levels. 0:Level 1 clock gating in module level 1:Level 2 clock gating in unit level 2:Level 3 clock gating in VPU level 7:3:Reserved

MCLK_CTL Bits

3.2.83 DEXC2DBG (0x7E0) (non-standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
RSVD												PMOV	SPF	LPF	IPF	BWE	SLPECC	ACE	HSP	MEC	RSVD	SEC	UEC	SAF	SAM	LAF	LAM	NMI	II	IAF	IAM						
N/A												RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	N/A	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	x	0	0	0	0	0	0	0	0	0	0						

DEXC2DBG [31:0]

Bits	Name	Description
19	PMOV	Indicates whether performance counter overflow interrupts are redirected to enter Debug Mode 0:Do not redirect 1:Redirect
18	SPF	Indicates whether store page fault exceptions are redirected to enter Debug Mode. 0:Do not redirect 1:Redirect
17	LPF	Indicates whether load fault exceptions are redirected to enter Debug Mode 0:Do not redirect 1:Redirect
16	IPF	Indicates whether instruction page fault exceptions are redirected to enter Debug Mode 0:Do not redirect 1:Redirect

Bits	Name	Description
15	BWE	Indicates whether Bus-write Transaction Error local interrupts are redirected to enter Debug Mode 0:Do not redirect 1:Redirect
14	SLPECC	Indicates whether local memory slave port ECC Error local interrupts are redirected to enter Debug Mode 0:Do not redirect 1:Redirect
13	ACE	Indicates whether ACE-related exceptions are redirected to enter Debug Mode. This bit is present only when mmisc_cfg.ACE is set 0:Do not redirect 1:Redirect
12	HSP	Indicates whether Stack Protection exceptions are redirected to enter Debug Mode. This bit is present only when mmisc_cfg.HSP is set. 0:Do not redirect 1:Redirect
11	MEC	Indicates whether M-mode Environment Call exceptions are redirected to enter Debug Mode 0:Do not redirect 1:Redirect
9	SEC	Indicates whether S-mode Environment Call exceptions are redirected to enter Debug Mode 0:Do not redirect 1:Redirect
8	UEC	Indicates whether U-mode Environment Call exceptions are redirected to enter Debug Mode. 0:Do not redirect 1:Redirect
7	SAF	Indicates whether Store Access Fault exceptions are redirected to enter Debug Mode. 0:Do not redirect 1:Redirect
6	SAM	Indicates whether Store Access Misaligned exceptions are redirected to enter Debug Mode. 0:Do not redirect 1:Redirect
5	LAF	Indicates whether Load Access Fault exceptions are redirected to enter Debug Mode. 0:Do not redirect 1:Redirect

Bits	Name	Description
4	LAM	Indicates whether Load Access Misaligned exceptions are redirected to enter Debug Mode 0:Do not redirect 1:Redirect
3	NMI	Indicates whether Non-Maskable Interrupt 0:Do not redirect 1:Redirect
2	II	Indicates whether Illegal Instruction exceptions are redirected to enter Debug Mode. 0:Do not redirect 1:Redirect
1	IAF	Indicates whether Instruction Access Fault exceptions are redirected to enter Debug Mode 0:Do not redirect 1:Redirect
0	IAM	Indicates whether Instruction Access Misaligned exceptions are redirected to enter Debug Mode. 0:Do not redirect 1:Redirect

DEXC2DBG Bits

3.2.84 DDCAUSE (0x7E1) (non-standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																SUBTYPE								MAINTYPE							
N/A																RO								RO							
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDCAUSE [31:0]

Bits	Name	Description
15-8	SUBTYPE	Subtypes for main type. The table below lists the subtypes for DCSR.CAUSE==1 and DDCAUSE.MAINTYPE==3. 0:Illegal instruction 1:Privileged instruction 2:Non-existent CSR 3:Privilege CSR access 4:Read-only CSR update

Bits	Name	Description
7-0	MAINTYPE	Cause for redirection to Debug Mode. 0:Software Breakpoint (EBREAK) 1:Instruction Access Misaligned (IAM) 2:Instruction Access Fault (IAF) 3:Illegal Instruction (II) 4:Non-Maskable Interrupt (NMI) 5:Load Access Misaligned (LAM) 6:Load Access Fault (LAF) 7:Store Access Misaligned (SAM) 8:Store Access Fault (SAF) 9:U-mode Environment Call (UEC) 10:S-mode Environment Call (SEC) 11:Instruction page fault 12:M-mode Environment Call (MEC) 13:Load page fault 14:Reserved 15:Store/AMO page fault 16:Imprecise ECC error 17:Bus write transaction error 18:Performance Counter overflow 19-31:Reserved 32:Stack overflow exception 33:Stack underflow exception 34:ACE disabled exception 35-39:Reserved 40-47:ACE exception ≥48:Reserved

DDCAUSE Bits

3.2.85 UITB (0x800) (non-standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																RW														RSVD	HW
																														N/A	RO
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	0

UITB [31:0]

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3
RISC-V PROCESSORS

3 CONTROL STATUS REGISTERS FOR

Bits	Name	Description
31-2	ADDR	The base address of the CoDense instruction table. This field is reserved if uitb.HW == 1.
0	HW	This bit specifies if the CoDense instruction table is hardwired. 0:The instruction table is located in memory. uitb.ADDR should be initialized to point to the table before using the CoDense instructions. 1:The instruction table is hardwired. Initialization of uitb.ADDR is not needed before using the CoDense instructions.

UITB Bits

3.2.86 UCODE (0x801) (non-standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RSVD																																OV
N/A																																RW
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	

UCODE [31:0]

Bits	Name	Description
0	OV	Overflow flag. It will be set by DSP instructions with a saturated result. 0:A saturated result is not generated 1:A saturated result is generated

UCODE Bits

3.2.87 UDCAUSE (0x809) (non-standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																												UDCAUSE			
N/A																												RW			
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	

UDCAUSE [31:0]

Bits	Name	Description
2-0	UDCAUSE	This register further disambiguates causes of traps recorded in the ucause register. See the list below for details. The value of UDCAUSE for precise exception: When ucause == 1 (Instruction access fault) 0:Reserved 1:ECC/Parity error 2:PMP instruction access violation 3:Bus error 4:PMA empty hole access When ucause == 2 (Illegal instruction) 0:Please parse the utval CSR 1:FP disabled exception 2:ACE disabled exception When ucause == 5 (Load access fault) 0:Reserved 1:ECC/Parity error 2:PMP load access violation 3:Bus error 4:Misaligned address 5:PMA empty hole access 6:PMA attribute inconsistency 7:PMA NAMO exception When ucause == 7 (Store access fault) 0:Reserved 1:ECC/Parity error 2:PMP store access violation 3:Bus error 4:Misaligned address 5:PMA empty hole access 6:PMA attribute inconsistency 7:PMA NAMO exception

UDCAUSE Bits

3.2.88 UCCTLBEGINADDR (0x80B) (non-standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																															
W																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3
RISC-V PROCESSORS

3 CONTROL STATUS REGISTERS FOR

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

UCCTLBEGINADDR [31:0]

Bits	Name	Description
31-0	VA	It is an alias to the mcctlbeginaddr register and it is only accessible to Supervisor-mode and User-mode software when mcache_ctl.CCTL_SUEN is 1. Otherwise illegal instruction exceptions will be triggered.

UCCTLBEGINADDR Bits

3.2.89 UCCTLCOMMAND (0x80C) (non-standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																										VA					
N/A																										RW					
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	

UCCTLCOMMAND [31:0]

Bits	Name	Description
4-0	VA	See User CCTL Command Definition Table

UCCTLCOMMAND Bits

Value (DEC)	Value (BIN)	Command	Type	User Mode Allowed
0	0b00_000	L1D_VA_INVALID	VA	1
1	0b00_001	L1D_VA_WB	VA	1
2	0b00_010	L1D_VA_WBINVAL	VA	1
3	0b00_011	L1D_VA_LOCK	VA	0
4	0b00_100	L1D_VA_UNLOCK	VA	0
6	0b00_110	L1D_WBINVAL_ALL	-	0
7	0b00_111	L1D_WB_ALL	-	0
8	0b01_000	L1I_VA_INVALID	VA	1
11	0b01_011	L1I_VA_LOCK	VA	0
12	0b01_100	L1I_VA_UNLOCK	VA	0
16	0b10_000	L1D_IX_INVALID	Index	0
17	0b10_001	L1D_IX_WB	Index	0
18	0b10_010	L1D_IX_WBINVAL	Index	0
19	0b10_011	L1D_IX_RTAG	Index	0
20	0b10_100	L1D_IX_RDATA	Index	0

Value (DEC)	Value (BIN)	Command	Type	User Mode Allowed
21	0b10_101	L1D_IX_WTAG	Index	0
22	0b10_110	L1D_IX_WDATA	Index	0
23	0b10_111	L1D_INVALID_ALL	-	0
24	0b11_000	L1I_IX_INVALID	Index	0
27	0b11_011	L1I_IX_RTAG	Index	0
28	0b11_100	L1I_IX_RDATA	Index	0
29	0b11_101	L1I_IX_WTAG	Index	0
30	0b11_110	L1I_IX_WDATA	Index	0

Table 9: CCTL Command Definition

3.2.90 SLIE (0x9C4) (non-standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD													PMOVI	BWEI	IMECCI	RSVD															
N/A													RW	RW	RW	N/A															
x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

SLIE [31:0]

Bits	Name	Description
18	PMOVI	Enable S-mode performance monitor overflow local interrupt. 0:Local interrupt is not enabled. 1:Local interrupt is enabled
17	BWEI	Enable S-mode bus read/write transaction error local interrupt. The processor may receive bus errors on load/store instructions or cache writebacks. 0:Local interrupt is not enabled. 1:Local interrupt is enabled
16	IMECCI	Enable S-mode slave-port ECC error local interrupt. 0:Local interrupt is not enabled. 1:Local interrupt is enabled

SLIE Bits

3.2.91 SLIP (0x9C5) (non-standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD													PMOVI	BWEI	IMECCI	RSVD															

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3
RISC-V PROCESSORS

3 CONTROL STATUS REGISTERS FOR

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
N/A													RW	RW	RW	N/A															
x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

SLIP [31:0]

Bits	Name	Description
18	PMOVI	Pending control and status of S-mode performance monitor overflow local interrupt. 0:Local interrupt is not enabled. 1:Local interrupt is enabled
17	BWEI	Pending control and status of S-mode bus read/write transaction error local interrupt. The processor may receive bus errors on load/store instructions or cache writebacks. 0:Local interrupt is not enabled. 1:Local interrupt is enabled
16	IMECCI	Pending control and status of S-mode slave-port ECC error local interrupt.. 0:Local interrupt is not enabled. 1:Local interrupt is enabled

SLIP Bits

3.2.92 SDCAUSE (0x9C9) (non-standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								PM		RSVD		SDCAUSE			
N/A																								RW		N/A		RW			
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	x	x	0	0	0

SDCAUSE [31:0]

Bits	Name	Description
6-5	PM	When scause is imprecise exception (in the form of an interrupt), the PM field records the privileged mode of the instruction that caused the imprecise exception. The PM field encoding is defined as follows: 0:User mode 1:Supervisor mode 2:Reserved 3:Machine mode

Bits	Name	Description
2-0	SDCAUSE	This register further disambiguates causes of traps recorded in the scause register. See the list below for details. The value of SDCAUSE for precise exception: When scause == 1 (Instruction access fault): 0:Reserved; 1:ECC/Parity error; 2:PMP instruction access violation; 3:Bus error; 4:PMA empty hole access When scause == 2 (Illegal instruction): 0:Please parse the stval CSR; 1:FP disabled exception; 2:ACE disabled exception When scause == 5 (Load access fault): 0:Reserved; 1:ECC/Parity error; 2:PMP load access violation; 3:Bus error; 4:Misaligned address; 5:PMA empty hole access; 6:PMA attribute inconsistency; 7:PMA NAMO exception When scause == 7 (Store access fault): 0:Reserved; 1:ECC/Parity error; 2:PMP load access violation; 3:Bus error; 4:Misaligned address; 5:PMA empty hole access; 6:PMA attribute inconsistency; 7:PMA NAMO exception The value of SDCAUSE for imprecise exception: When scause == Local Interrupt 16 or Local Interrupt 272 (16 + 256) (ECC error local interrupt) 0:Reserved; 1:LM slave port ECC/Parity error; 2:Imprecise store ECC/Parity error; 3:Imprecise load ECC/Parity error When scause == Local Interrupt 17 or Local Interrupt 273 (17 + 256) (Bus read/write transaction error local interrupt) 0:Reserved; 1:Bus read error; 2:Bus write error; 3:PMP error caused by load instructions; 4:PMP error caused by store instructions; 5:PMA error caused by load instructions; 6:PMA error caused by store instructions

SDCAUSE Bits

3.2.93 SCCTLDATA (0x9CD) (non-standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																												VA			
N/A																												RW			
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	

SCCTLDATA [31:0]

Bits	Name	Description
4-0	VA	See CCTL Commands Which Access mcctldata Table

SCCTLDATA Bits

3.2.94 SCOUNTERINTEN (0x9CF) (non-standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								HPM6	HPM5	HPM4	HPM3	IR	RSVD	CY	
N/A																								RW	RW	RW	RW	RW	N/A	RW	
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	x	0

SCOUNTERINTEN [31:0]

Bits	Name	Description
6	HPM6	See register description
5	HPM5	See register description
4	HPM4	See register description
3	HPM3	See register description
2	IR	See register description
0	CY	See register description

SCOUNTERINTEN Bits

3.2.95 SCOUNTERMASK_M (0x9D1) (non-standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								HPM6	HPM5	HPM4	HPM3	IR	RSVD	CY	
N/A																								RW	RW	RW	RW	RW	N/A	RW	
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	x	0

SCOUNTERMASK_M [31:0]

Bits	Name	Description
6	HPM6	See register description
5	HPM5	See register description
4	HPM4	See register description
3	HPM3	See register description
2	IR	See register description
0	CY	See register description

SCOUNTERMASK_M Bits

3.2.96 SCOUNTERMASK_S (0x9D2) (non-standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								HPM6	HPM5	HPM4	HPM3	IR	RSVD	CY	
N/A																								RW	RW	RW	RW	RW	N/A	RW	
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	x	0

SCOUNTERMASK_S [31:0]

Bits	Name	Description
6	HPM6	See register description
5	HPM5	See register description
4	HPM4	See register description
3	HPM3	See register description
2	IR	See register description
0	CY	See register description

SCOUNTERMASK_S Bits

3.2.97 SCOUNTERMASK_U (0x9D3) (non-standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								HPM6	HPM5	HPM4	HPM3	IR	RSVD	CY	
N/A																								RW	RW	RW	RW	RW	N/A	RW	
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	x	0	

SCOUNTERMASK_U [31:0]

Bits	Name	Description
6	HPM6	See register description
5	HPM5	See register description
4	HPM4	See register description
3	HPM3	See register description
2	IR	See register description
0	CY	See register description

SCOUNTERMASK_U Bits

3.2.98 SCOUNTEROVF (0x9D4) (non-standard read/write)

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3
RISC-V PROCESSORS

3 CONTROL STATUS REGISTERS FOR

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								HPM6	HPM5	HPM4	HPM3	IR	RSVD	CY	
N/A																								RW	RW	RW	RW	RW	N/A	RW	
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	x	0	

SCOUNTEROVF [31:0]

Bits	Name	Description
6	HPM6	See register description
5	HPM5	See register description
4	HPM4	See register description
3	HPM3	See register description
2	IR	See register description
0	CY	See register description

SCOUNTEROVF Bits

3.2.99 SCOUNTINHIBIT (0x9E0) (non-standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								HPM6	HPM5	HPM4	HPM3	IR	TM	CY	
N/A																								RW	RW	RW	RW	RW	RW	RW	
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	

SCOUNTINHIBIT [31:0]

Bits	Name	Description
6	HPM6	See register description
5	HPM5	See register description
4	HPM4	See register description
3	HPM3	See register description
2	IR	See register description
1	TM	See register description
0	CY	See register description

SCOUNTINHIBIT Bits

3.2.100 SHPMEVENT3 (0x9E3) (non-standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																							SEL				TYPE				

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3
RISC-V PROCESSORS

3 CONTROL STATUS REGISTERS FOR

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
N/A																							RW					RW			
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0

SHPMEVENT3 [31:0]

Bits	Name	Description
8-4	SEL	Event Selectors table
3-0	TYPE	Event Selectors table

SHPMEVENT3 Bits

3.2.101 SHPMEVENT4 (0x9E4) (non-standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																							SEL					TYPE			
N/A																							RW					RW			
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0

SHPMEVENT4 [31:0]

Bits	Name	Description
8-4	SEL	Event Selectors table
3-0	TYPE	Event Selectors table

SHPMEVENT4 Bits

3.2.102 SHPMEVENT5 (0x9E5) (non-standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																							SEL					TYPE			
N/A																							RW					RW			
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0

SHPMEVENT5 [31:0]

Bits	Name	Description
8-4	SEL	Event Selectors table
3-0	TYPE	Event Selectors table

SHPMEVENT5 Bits

3.2.103 SHPMEVENT6 (0x9E6) (non-standard read/write)



HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3
RISC-V PROCESSORS

3 CONTROL STATUS REGISTERS FOR

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																						SEL					TYPE				
N/A																						RW					RW				
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	

SHPMEVENT6 [31:0]

Bits	Name	Description
8-4	SEL	Event Selectors table
3-0	TYPE	Event Selectors table

SHPMEVENT6 Bits

3.2.104 MCYCLE (0xB00) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNTER																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

MCYCLE [31:0]

Bits	Name	Description
31-0	COUNTER	the lower 32 bits of Machine Cycle Counter

MCYCLE Bits

3.2.105 MINSTRET (0xB02) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNTER																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

MINSTRET [31:0]

Bits	Name	Description
31-0	COUNTER	the lower 32 bits of Machine Instruction-Retired Counter

MINSTRET Bits

3.2.106 MHPMCOUNTER3 (0xB03) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNTER																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MHPMCOUNTER3 [31:0]

Bits	Name	Description
31-0	COUNTER	count the num- ber of events selected by mhpmevent3

MHPMCOUNTER3 Bits

3.2.107 MHPMCOUNTER4 (0xB04) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNTER																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MHPMCOUNTER4 [31:0]

Bits	Name	Description
31-0	COUNTER	count the num- ber of events selected by mhpmevent4

MHPMCOUNTER4 Bits

3.2.108 MHPMCOUNTER5 (0xB05) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNTER																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MHPMCOUNTER5 [31:0]

Bits	Name	Description
31-0	COUNTER	count the num- ber of events selected by mhpmevent5

MHPMCOUNTER5 Bits

3.2.109 MHPMCOUNTER6 (0xB06) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNTER																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MHPMCOUNTER6 [31:0]

Bits	Name	Description
31-0	COUNTER	count the number of events selected by mhpmevent6

MHPMCOUNTER6 Bits

3.2.110 MCYCLEH (0xB80) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNTER																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MCYCLEH [31:0]

Bits	Name	Description
31-0	COUNTER	the higher 32 bits of Machine Cycle Counter

MCYCLEH Bits

3.2.111 MINSTRETH (0xB82) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNTER																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MINSTRETH [31:0]

Bits	Name	Description
31-0	COUNTER	the higher 32 bits of Machine Instruction-Retired Counter

MINSTRETH Bits

3.2.112 MHPMCOUNTER3H (0xB83) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNTER																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MHPMCOUNTER3H [31:0]

Bits	Name	Description
31-0	COUNTER	count the num- ber of events selected by mhpmevent3

MHPMCOUNTER3H Bits

3.2.113 MHPMCOUNTER4H (0xB84) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNTER																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MHPMCOUNTER4H [31:0]

Bits	Name	Description
31-0	COUNTER	count the num- ber of events selected by mhpmevent4

MHPMCOUNTER4H Bits

3.2.114 MHPMCOUNTER5H (0xB85) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNTER																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MHPMCOUNTER5H [31:0]

Bits	Name	Description
31-0	COUNTER	count the num- ber of events selected by mhpmevent5

MHPMCOUNTER5H Bits

3.2.115 MHPMCOUNTER6H (0xB86) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNTER																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MHPMCOUNTER6H [31:0]

Bits	Name	Description
31-0	COUNTER	count the number of events selected by mhpmevent6

MHPMCOUNTER6H Bits

3.2.116 PMACFG0 (0xBC0) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMA3CFG								PMA2CFG								PMA1CFG								PMA0CFG							
RW								RW								RW								RW							
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PMACFG0 [31:0]

Bits	Name	Description
31-24	PMA3CFG	See PMACFG register
23-16	PMA2CFG	See PMACFG register
15-8	PMA1CFG	See PMACFG register
7-0	PMA0CFG	See PMACFG register

PMACFG0 Bits

3.2.117 PMACFG1 (0xBC1) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMA7CFG								PMA6CFG								PMA5CFG								PMA4CFG							
RW								RW								RW								RW							
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PMACFG1 [31:0]

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3
RISC-V PROCESSORS

3 CONTROL STATUS REGISTERS FOR

Bits	Name	Description
31-24	PMA7CFG	See PMACFG register
23-16	PMA6CFG	See PMACFG register
15-8	PMA5CFG	See PMACFG register
7-0	PMA4CFG	See PMACFG register

PMACFG1 Bits

3.2.118 PMACFG2 (0xBC2) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMA11CFG								PMA10CFG								PMA9CFG								PMA8CFG							
RW								RW								RW								RW							
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PMACFG2 [31:0]

Bits	Name	Description
31-24	PMA11CFG	See PMACFG register
23-16	PMA10CFG	See PMACFG register
15-8	PMA9CFG	See PMACFG register
7-0	PMA8CFG	See PMACFG register

PMACFG2 Bits

3.2.119 PMACFG3 (0xBC3) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMA15CFG								PMA14CFG								PMA13CFG								PMA12CFG							
RW								RW								RW								RW							
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PMACFG3 [31:0]

Bits	Name	Description
31-24	PMA15CFG	See PMACFG register
23-16	PMA14CFG	See PMACFG register
15-8	PMA13CFG	See PMACFG register
7-0	PMA12CFG	See PMACFG register

PMACFG3 Bits

7	6	5	4	3	2	1	0
RSVD	NAMO	MTYP				ETYP	
N/A	RW	RW				RW	
x	0	0	0	0	0	0	0

Table 10: PMACFG [7:0]

Bits	Name	Description
6	NAMO	Indicate whether Atomic Memory Operation instructions (including LR/SC) are not supported in this region. When this bit is set and an AMO instruction is encountered in this region, an access fault PMA NAMO exception will be generated. 0: AMO instructions (including LR/SC) are supported in the region. 1: AMO instructions (including LR/SC) are not supported in the region.
5-2	MTYP	Memory type attribute. This field defines the cacheability and idempotency of memory regions. In the table below, “Device” regions are non-idempotent regions and “Memory” regions are idempotent. The non-cacheable memory regions (type 2 and 3) are also referred to as uncached regions by this document. 0: Device, Non-bufferable 1: Device, bufferable 2: Memory, Non-cacheable, Non-bufferable 3: Memory, Non-cacheable, Bufferable 4: Memory, Write-through, No-allocate 5: Memory, Write-through, Read-allocate 6: Reserved. Hardware converts the written value to 4 7: Reserved. Hardware converts the written value to 5 8: Memory, Write-back, No-allocate 9: Memory, Write-back, Read-allocate 10: Memory, Write-back, Write-allocate 11: Memory, Write-back, Read and Write-allocate 12 - 14: Reserved. Hardware converts the written value to 15 15: Empty hole, nothing exists. The instruction fetch will generate an instruction access fault. The load instruction access will generate a load access fault. The store instruction access will generate a store access fault.

Bits	Name	Description
1-0	ETYP	Entry address matching mode. 0: OFF: This PMA entry is disabled. 1: Reserved. 2: Reserved. 3: NAPOT: Naturally aligned power-of-2 region. The granularity is 4K bytes. This mode makes use of the low-order bits of the associated address register to encode the size of the range. This field will be 0 when it is set to 1 or 2.

Table 11: PMACFG Bit Fields

3.2.120 PMAADDR[PMAADDR0] (0xBD0 + 0x1 * n) (standard read/write)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMAADDR_31_2																RW														RSVD	
																														N/A	
*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	x	x

PMAADDR [31:0]

Bits	Name	Description
31-2	PMAADDR_31_2	Register Content : Match Size(Byte) aaaa. . . aaaaaaaaaa Reserved aaaa. . . aa011111111 Reserved aaaa. . . a011111111 2^{12} aaaa. . . 0111111111 2^{13} aa01. . . 1111111111 2^{XLEN} a011. . . 1111111111 2^{XLEN+1} 0111. . . 1111111111 2^{XLEN+2} 1111. . . 1111111111 Reserved

PMAADDR Bits

3.2.121 CYCLE (0xC00) ()

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CYCLE																															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CYCLE [31:0]

Bits	Name	Description
31-0	CYCLE	Cycle Counter

CYCLE Bits

3.2.122 CYCLEH (0xC80) ()

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CYCLEH																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CYCLEH [31:0]

Bits	Name	Description
31-0	CYCLEH	Cycle Counter Higher 32-bit

CYCLEH Bits

3.2.123 MVENDORID (0xF11) (standard read only)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MVENDORID																															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	1	1	0

MVENDORID [31:0]

Bits	Name	Description
31-0	MVENDORID	The manufacturer ID

MVENDORID Bits

3.2.124 MARCHID (0xF12) (standard read only)

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3
RISC-V PROCESSORS

3 CONTROL STATUS REGISTERS FOR

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RSVD	CPU_ID																															
N/A	RO																															
x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1

MARCHID [31:0]

Bits	Name	Description
30-0	CPU_ID	CPU ID

MARCHID Bits

3.2.125 MIMPID (0xF13) (standard read only)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAJOR																								MINOR				EXTENSION			
RO																								RO				RO			
*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	

MIMPID [31:0]

Bits	Name	Description
31-8	MAJOR	Revision major
7-4	MINOR	Revision minor
3-0	EXTENSION	Revision extension

MIMPID Bits

3.2.126 MHARTID (0xF14) (standard read only)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MHARTID																															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MHARTID [31:0]

Bits	Name	Description
31-0	MHARTID	Hart ID

MHARTID Bits

4 Exceptions and interrupts for TRAP processor

This chapter describes the TRAP of the RISC-V processor, which is the processor's exceptions and interrupts.

Please refer to [Section 4.4](#) for all peripherals in this product that support interrupt generation and their interrupt vector table.

4.1 RISC-V Processor TRAP Overview

This product uses a high-performance RISC-V processor as the central processing unit. The processor complies with the RISC-V specification. RISC-V refers to all program execution stream conversions caused by the processor itself or peripherals as traps. When a trap occurs, the processor interrupts execution of the current program and responds instead to the trap's handler.

According to the RISC-V specification, traps are divided into two types: exceptions and interrupts. An exception refers to a TRAP that is raised by the processor itself due to an exception in instruction execution. Interrupts refer to requests issued by various peripherals to the processor, requiring the processor to respond.

Interrupts can be divided into local interrupts and external interrupts. Local interrupts are mainly caused by the processor itself or its private devices, such as software interrupts, timer interrupts, etc. Local interrupts can be managed by accessing the processor's CSR(control and status registers). Various interrupts generated by most peripherals of this product are managed by the platform interrupt controller PLIC. The two interrupt request outputs (Target 0 and Target 1) of the PLIC are connected to the Machine External Interrupt of the RISC-V processor processor respectively, and the Supervisor External Interrupt.

platform interrupt controller PLIC manages all on-chip peripheral interrupts. The RISC-V specification defines all functions of the PLIC. The PLIC on this product is compatible with the RISC-V specification and supports enhanced features: interrupt nesting extension and interrupt vector extension.

4.1.1 exception

RISC-V processor supports the following exceptions:

- Instruction address misaligned
 - instruction jumps to a non-aligned address
- Error Instruction access fault
 - Bus error while fetching instructions
 - PMP error when retrieving instructions
 - PMA error when fetching fingers
- Illegal instruction
 - Executing illegal instructions
 - Access illegal CSR
- Breakpoint
- Data load Load address misaligned Load address
- Data loading Load error Load access fault
 - Bus error while loading data in the Load command
 - Load command PMP error while loading data

- Load command
- Store or AMO command address is misaligned Store/AMO address misaligned
- Error in Store or AMO command Store/AMO access fault
 - Store instruction
 - PMP error while storing data
 - PMA error when storing data in the Store command
- User mode Environment call from U-mode
- Supervision Mode Environment call from S-mode
- Stack overflow exception
- Stack overflow exception

4.1.2 interrupt

The interrupts supported by RISC-V processors can be divided into local interrupts and external interrupts:

local interrupts are generated by the processor's own devices, including:

- Machine Mode M-Mode software interrupt, generated by software interrupt controller PLICSW
- Machine mode M-Mode timer interrupt, generated by machine timer MCHTMR
- Supervisory Mode S-Mode software interrupt
- Supervision Mode S-Mode timer interrupt
- Bus access error interrupt in machine mode (M-mode)
- Performance monitor overflow interrupt in machine mode (M-mode)
- Bus write transaction error interrupt (S-mode)
- Performance monitor overflow interrupt in supervisory mode (S-mode)

External interrupt refers to machine external interrupt and supervisor external interrupt. External interrupts are managed by the platform interrupt controller PLIC. The PLIC arbitrates each interrupt based on priority and sends qualified interrupts to the processor.

4.1.3 Platform Interrupt Controller PLIC

This product's Platform-Level Interrupt Controller manages interrupts generated by all peripherals on the chip.

PLIC has the following characteristics:

- PLIC outputs interrupts to the processor's external interrupt
- supports 32 priority levels.
- supports configurable interrupt edge triggering or level triggering
- supports interrupt nesting, allowing high-priority interrupts to interrupt executing low-priority interrupts
- supports interrupt vectors. The interrupt vector mode can simplify the interrupt declaration procedure defined by the RISC-V specification and reduce interrupt response delay

4.1.4 Software Interrupt Controller PLICSW

The machine software interrupts of the RISC-V processor of this product are managed by the software interrupt controller PLICSW. The programming model of PLICSW is similar to PLIC, but it only contains one interrupt source. Users can set the interrupt pending bit of PLICSW to generate machine software interrupts.

4.1.5 Machine Timer MCHTMR

RISC-V specification defines machine timers. Machine timers are required to run at a fixed clock frequency and generate timer interrupts. The machine timer interrupt of the RISC-V processor of this product is generated by the machine timer MCH_TMR. The machine timer supports a 64-bit counter and a 64-bit comparator. When the counter count reaches or exceeds the comparator value, a Machine timer interrupt is generated.

4.2 TRAP and Processor Privilege Mode

The RISC-V processor used in this product supports three privileged modes defined by the RISC-V specification: Machine Mode, Supervisor Mode, and User Mode.

When RISC-V processors run, they are always in a privileged mode. Among them, machine mode is the privileged mode with the highest authority, followed by supervision mode, and user mode has the lowest authority. Privilege mode can be used to isolate software from the on-chip resources allocated to software. When HART runs in a lower-privilege mode, it is not allowed to access resources exclusive to the high-privilege mode (such as a certain section of memory, a certain peripheral, etc.). Users can thus establish a trusted execution environment to protect sensitive information or resources from illegal access.

The RISC-V specification stipulates that when all TRAP occurs, processor mode will be switched to machine mode to process TRAP services. Note that even if some interrupts are designed for low-privilege mode, such as supervisor software interrupts, when these interrupts occur, the processors still go into machine mode to process their interrupt service routines.

However, the RISC-V specification also allows TRAP to be redirected to lower-privilege mode. For example, setting the corresponding bit in the medeleg or mideleg CSR can authorize exceptions or interrupts to supervisory mode. When an authorized exception or interrupt occurs in the processor, HART is no longer in machine mode, but handles TRAP's service procedures in supervisory mode.

It should be noted that TRAP does not cause the processor privilege level to shift from high-privilege mode to low-privilege mode. For example, if medeleg CSR authorizes illegal instruction exceptions to supervisor mode, then the processor executes the illegal instruction in supervisor mode and user mode, and the processor will handle the raised exception in supervisor mode. However, if an illegal instruction is executed in machine mode, the processor will still handle the exception in machine mode even if the illegal instruction exception has been authorized to supervisory mode.

By the same token, external machine interrupts, machine software interrupts, and machine timer interrupts cannot be authorized to the supervision mode.

When the processor is in a higher privilege mode, it does not respond to interrupts authorized to a lower privilege mode. For example, supervisory software interrupts are authorized to supervisory mode by setting the mideleg register. If a supervisory software interrupt occurs while the processor is in machine mode, the processor will not respond to the interrupt until it exits machine mode.

On this product, the PLIC supports two interrupt request outputs, which are connected to the Machine External interrupt and Supervisor External interrupt of the RISC-V processor. Please note that don't be misled by the name of the interrupt. After the interrupt occurs, the processor will respond to the interrupt in machine mode. The processor will respond to the supervisory external interrupt in supervisory mode only after the Supervisor External interrupt is authorized to supervisory mode, and machine external interrupts cannot be authorized to

supervisory mode.

On this product, the interrupt request output of the PLIC is connected to the Machine External interrupt of the RISC-V processor. The machine external interrupt cannot be authorized to supervisory mode or user mode.

users can run software in supervisory mode or user mode, but processor privileged mode switches to machine mode when an interrupt or exception occurs.

If the user does not require software to be distinguished by the privileged mode of the processor and given different access rights to resources. In this case, only the machine external interrupt request output of PLIC can be used. Such regular programs and interrupt service response programs run in machine mode.

If the user wants to distinguish software through the privilege mode of the processor, and distinguish interrupt requests from software with different privilege modes. At this time, PLIC's machine external interrupt request output and supervisory external interrupt request output can be used simultaneously, and supervisory external interrupts can be authorized to supervisory mode. In this way, the processor can respond to supervisory external interrupt service procedures in supervisory mode. During an interrupt, the processor's privileged mode does not have to switch to machine mode, achieving complete isolation of supervisory mode and machine mode software.

4.3 Interrupt nesting

interrupt nesting extension means that this product allows the processor to suspend low-priority interrupt service routines and respond instead to high-priority interrupt service routines. The platform level interrupt controller PLIC(Platform Level Interrupt Controlled) of this product adds this extended function to the PLIC defined in the RISC-V specification, allowing the processor to better meet the requirements of real-time system applications.

Users can turn on interrupt nesting by setting the PREEMPT bit of the PLIC's Feature Enable Register to 1. When the processor responds to the interrupt service routine, if the MIE bit of the mstatus CSR is 1, a higher-priority interrupt can interrupt the current interrupt service routine. After the processor responds to a high-priority interrupt, it continues to process the interrupted interrupt service routine.

In the interrupt service routine, the PLIC does not respond to interrupts of the same or lower priority.

中断嵌套的具体实现过程为:

- When an interrupt claim occurs, the PLIC will save the current Priority Threshold register, Assign the priority of newly arriving interrupts to the Priority Threshold Register.
- At the end of the interrupt completion, the PLIC will restore the Priority Threshold Register to the previous value.

On this product, PLIC manages almost all peripheral interrupts and outputs interrupt requests to the processor's machine external interrupts and supervisory external interrupts. Therefore, the interrupt priority setting of the PLIC is only valid for peripheral interrupts managed by the PLIC.

When the software sets the MIE bit of mstatus CSR to 1 in the interrupt service routine, PLIC allows interrupts with higher interrupt priority to be used as new external interrupts to the machine to interrupt the current interrupt. In addition, processor exceptions and other local interrupts, such as software interrupts or timer interrupts, can also interrupt the current interrupt. The priority of these exceptions and other interrupts is not affected by the priority setting of the PLIC.

4.4 interrupt vector

When a trap occurs, the processor jumps to mtvec CSR(control and status registers)

. The software can use the mcause CSR to determine the source of the trap, whether it is an interrupt or an exception. If the trap that occurs is an external interrupt (external to the machine or supervisory external interrupt), the software needs to further determine the source of the interrupt through PLIC's interrupt pending register.

Interrupt vector extension is an enhanced feature of PLIC that reduces interrupt response latency on peripherals of this product. Interrupt vector expansion means that when a trap is triggered by an external interrupt, the processor will read the address pointer of the interrupt service program from the interrupt vector table and assign it to the program counter PC of the processor. In this way, the processor will directly jump to the corresponding interrupt service routine, and the software does not need to first read the mcause CSR and then read the claim/complete register of the PLIC to determine the source of the interrupt.

Users can turn on interrupt vector mode by setting the VECTORED bit of the PLIC Feature Enable Register to 1 and setting the VEC_PLIC bit of the MMISC CSR to 1.

The RISC-V specification stipulates that by default, all traps are handled in machine mode. At this time, the base address of the interrupt vector table is stored in the processor's mtvec(machine trap vector base address register)CSR.

For external interrupt to the machine (output by Target 0 of PLIC, and set to 1 MEIP that triggers mip CSR):

- is mtvec, and the address of the interrupt vector with sequence number N is: base address + 4 x N.

If the user does not delegate supervisory external interrupt to supervisory mode, for supervisory external interrupt (output by Target 1 of PLIC, set to 1 SEIP that triggers mip CSR):

- is mtvec + 1024, and the address of the interrupt vector with sequence number N is: base address + 4 x N

If the user configures mideleg by (machine interrupt delegation)CSR SEI position 1, external interruption of supervision (supervisor external interrupt) Authorized to supervisor mode, then for supervisory external interrupts (The Target 1 output of the PLIC causes the SEIP of the sip CSR to be set to 1.) The privileged mode of the processor is no longer converted to machine mode, but to supervisor mode. In this case, the base address of the interrupt vector table is stored in the processor's stvec(Supervisor Trap Vector Base Address Register)CSR.

- The address of the interrupt vector with sequence number N is: base address + 4 x N

The interrupt vector allocation of this product is as follows:

Interrupt Number	Interrupt Name	Description
1	GPIO0_A	
2	GPIO0_B	
3	GPIO0_C	
4	GPIO0_D	
5	GPIO0_E	
6	GPIO0_F	

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3
TRAP PROCESSOR

4 EXCEPTIONS AND INTERRUPTS FOR

Interrupt Number	Interrupt Name	Description
7	GPIO0_V	
8	GPIO0_W	
9	GPIO0_X	
10	GPIO0_Y	
11	GPTMR0	
12	GPTMR1	
13	GPTMR2	
14	GPTMR3	
15	OWR0	
16	OWR1	
17	EUI0	
18	EUI1	
19	UART0	
20	UART1	
21	UART2	
22	UART3	
23	UART4	
24	UART5	
25	UART6	
26	UART7	
27	I2C0	
28	I2C1	
29	I2C2	
30	I2C3	
31	SPI0	
32	SPI1	
33	SPI2	
34	SPI3	
35	TSNS	
36	MBX0A	
37	MBX0B	
38	EWDG0	
39	EWDG1	
40	HDMA	
41	LOBS	
42	ADC0	
43	ADC1	
44	ACMP0[0]	
45	ACMP0[1]	

Interrupt Number	Interrupt Name	Description
46	MCAN0	
47	MCAN1	
48	MCAN2	
49	MCAN3	
50	PTPC	
51	QEIO	
52	QEIO	
53	PWM0	
54	PWM1	
55	SDM0	
56	TRGM[0]	
57	TRGM[1]	
58	ENET0	
59	NTMR0	
60	USB0	
61	ESC	
62	ESC_SYNC0	
63	ESC_SYNC1	
64	ESC_RESET	
65	XPIO	
66	PPI	
67	XDMA	
68	PGPIO	
69	PEWDG	
70	PTMR	
71	PUART	
72	FUSE	
73	DGO_PAD_WAKEUP	
74	DGO_CNT_WAKEUP	
75	BROWNOUT	
76	SYSCTL	
77	CPU0	
78	DEBUG0	
79	DEBUG1	

Table 12: IRQ List

4.5 interrupt response process

For the general interrupt service procedure process, taking machine mode (M-Mode) as an example, the processor hardware will take the following actions:

- Updates mepc CSR value to current PC(program counter)
- Updates the value of mstatus CSR:
 - Updates MPP bit field value to current privileged mode
 - Updates MPIE value to MIE
 - MIE Position 0
- Updates the value of the mcause CSR to include current interrupt source information
- Processor privileged mode changed to machine mode
- Updates PC(program counter) based on the value of mtvec CSR
 - If interrupt vectors are not supported, the value of mtvec CSR is assigned to the PC
 - If interrupt vectors are supported, the value of the base address + 4 * N is assigned to the PC, where N is the interrupt ID number of the PLIC

◦

After that, the software that interrupts the service routine will start executing, and the software should:

- saves the necessary processor integer registers and pushes them onto the stack
- If interrupt vectors are not supported, or if the TRAP occurring is an exception or a local interrupt, save the mcause CSR
- If you want to support interrupt nesting, you should:
 - Save mepc CSR
 - Save mstatus CSR
 - Save other necessary CSRs as needed, such as floating point/DSP extension related CSRs
 - sets the MIE bit of mstatus CSR to 1, allowing new interrupts to be generated
- Start executing interrupt service routine
 - If interrupt vectors are not supported, or the TRAP occurring is an exception or a local interrupt, you need to determine the cause of the TRAP based on mcause. and execute the corresponding processing procedures. If it is an external interrupt sent to the processor by the PLIC, You need to read the PLIC's Claim/Complete register to complete interrupt declaration. The interrupt ID number is obtained to determine the interrupt source, and then the corresponding interrupt service procedure is executed.
 - If interrupt vectors are supported, directly read the interrupt identification bit of the corresponding peripheral device, determine the cause of the interrupt, and execute the interrupt service program

After completing the corresponding service program for TRAP, the processing flow is as follows:

- reads the PLIC's Claim/Complete register to indicate that the interrupt is complete.
- If it is an interrupt that allows nesting, you should:
 - Set the MIE bit of mstatus CSR to 0 and no longer respond to new interrupts
 - Insert a FENCE instruction to ensure that the previous read PLIC interrupt assertion/completion register operation is completed
 - Recovering mepc CSR

- Recovering mstatus CSR
- Recovers other previously saved CSRs, such as floating point/DSP extension related CSRs
- Software restores processor integer registers that were previously pushed onto the stack
- software executes the MRET command and returns from machine mode.

Once the MRET instruction is executed, the processor hardware performs the following actions:

- Restore the value of mepc CSR to PC(program counter)
- Recovers the value of mstatus CSR:
 - Restores the privileged mode of the processor based on the MPP bit field value
 - Restore MPIE values to MIE

5 Machine Timer MCHTMR

This chapter describes the functions and features of the machine timer MCHTMR.

5.1 Feature Summary

This chapter describes the main features of the machine timer MCHTMR:

- 64-bit real-time counter
- 64-bit counter comparator
- supports 32-bit and 64-bit register access
- supports sending machine timer interrupts to the RISC-V core

MCHTMR is shown as [Figure 2](#).

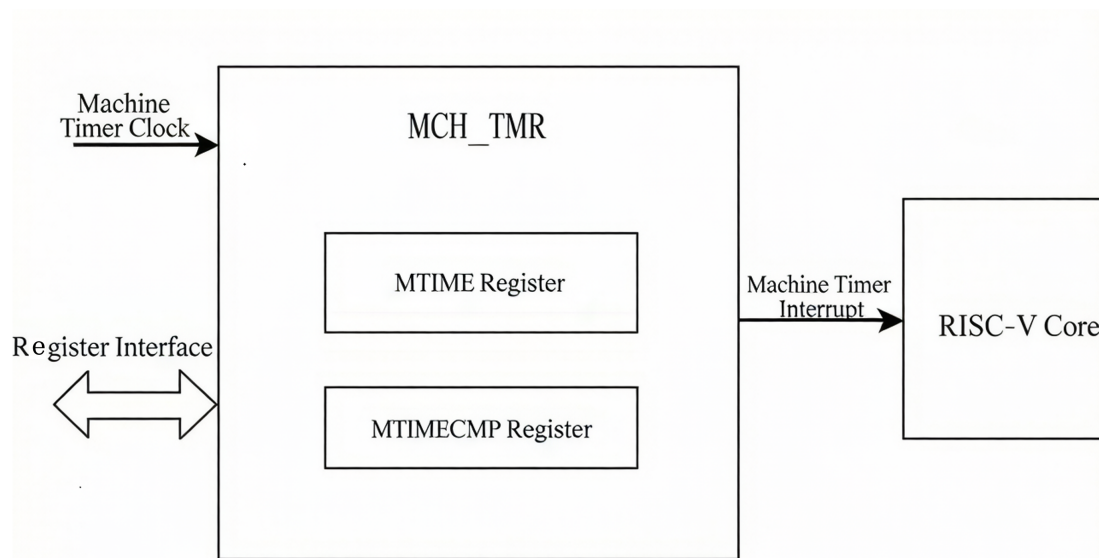


Figure 2: MCHTMR block diagram

5.2 Function Description

This chapter describes the functions of the machine timer MCHTMR.

5.2.1 Timing and Timing Function

MCHTMR has a built-in 64-bit counter, and the value of the counter is increased by 1 each time the machine timer clock rolls. Read the MTIME register to get the current count value. The

timing function is realized by setting the MTIMECMP register. When the counter value is greater than or equal to the MTIMECMP register, MCHTMR issues a machine timer interrupt to the RISC-V core.

When a machine timer interrupt is active, a write to the MTIMECMP register clears the interrupt.

5.2.2 Timer reset

The internal counter starts from 1. Overwriting the MTIME register can reset the counter to a specific value. Certain rules need to be followed to modify the MTIME register: the

MTIME register is divided into two 32-bit registers, MTIME[63:32] and MTIME[31:0]. When modifying the MTIME register, the two 32-bit registers must be operated respectively in a 32-bit manner.

- If you are going to set the target value of MTIME[31:29] to 7, then:
 1. Set MTIME[31:0] to 0
 2. Set MTIME[63:32] target values
 3. Set MTIME[31:0] target value
- If the target value of MTIME[31:29] is to be set to other than 7, then:
 1. Set the MTIME[31:0] target value
 2. Set the MTIME[63:32] target value

5.3 MCHTMR register description

MCHTMR is as follows:

MCHTMR base address: 0xE6000000

Offset	Register Name	Description	Reset Value
0x0000	MTIME	Machine Time	0x00000000000020210
0x0008	MTIMECMP	Machine Time Compare	0x00000000000020210

Table 13: MCHTMR Register List

5.4 MCHTMR register details

5.4.1 MTIME (0x0)

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
MTIME																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0

MTIME [63:32]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MTIME																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0

MTIME [31:0]

Bits	Name	Description
63-0	MTIME	Machine time

MTIME Bits

5.4.2 MTIMECMP (0x8)

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32	
																MTIMECMP																
RW																																
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	

MTIMECMP [63:32]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0															
																MTIMECMP																														
																																RW														
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0															

MTIMECMP [31:0]

Bits	Name	Description
63-0	MTIMECMP	Machine time compare

MTIMECMP Bits

6 Platform-level interrupt controller PLIC

This chapter describes the functions and features of the platform-level interrupt controller PLIC.

6.1 Feature Summary

This chapter describes the main features of the platform-level interrupt controller PLIC:

- Configurable interrupt triggering method
- supports vector interrupts
- supports 32 priority levels
- supports interrupt priority preemption

6.2 Function Description

This chapter describes the functions of the platform-level interrupt controller PLIC.

PLIC collects the interrupt requests from all interrupt sources and sends them to the interrupt gateway. After completing the priority processing based on the configuration, it forwards the interrupt requests to the RISC-V core.

Interrupt configuration includes enabling and priority settings. It should be noted that the software should not change the interrupt configuration when there are unprocessed interrupts.

6.2.1 Vector Interrupt Extension

RISC-V standard requires the PLIC to notify the RISC-V core when receiving an external interrupt. The RISC-V core reads the CLAIM_COMPLETE register of the PLIC to obtain the interrupt ID. This reading operation is called an interrupt response (claim). After obtaining the interrupt ID, the RISC-V core executes the corresponding interrupt handler, and then writes the ID back to the CLAIM_COMPLETE register of PLIC. This write operation is called interrupt complete. The PLIC of this product of also supports vector interrupt extension, that is, when the PLIC notifies the RISC-V core of the interrupt request, it will issue the interrupt ID together. The RISC-V core sends the interrupt response to the PLIC in hardware. The software does not need to read the PLIC to obtain the interrupt ID and execute the interrupt response to speed up interrupt processing.

Enable the vector interrupt extension by configuring the `FEATURE_ENABLE.VECTORED` register.

6.2.2 Interrupt Priority Preemption

PLIC supports interrupt preemption priority. In the PRIORITY register, the priority of each interrupt ID can be configured. The maximum valid priority value is 31. The larger the value, the higher the interrupt priority. When priority preemption is enabled, if the RISC-V core has responded to a certain interrupt but the interruption has not been completed yet, and if a new interrupt with a higher priority appears at this time, PLIC will send a new interrupt request to the RISC-V core again. Setting the register `FEATURE.PREEMPT` to 1 enables the priority preemption function.

In the interrupt handling program, the global interrupt is enabled by configuring the [MIE] bit field in the machine mode status register (MSTATUS) of the control status register module of the RISC-V processor, allowing high-priority interrupts to preempt the current interrupt handling program being executed.

In non-vector mode, the RISC-V core should enable interrupt preemption after saving the context and com-

pleting the interrupt response. Since both interrupt response and interrupt completion involve executing loadstore instructions in the device area, they will be automatically sequenced, so there is no need to additionally disable global interrupts, nor is it necessary to insert a FENCE instruction after issuing the interrupt completion operation. The suggested non-vector mode interrupt handler is as follows:

1. Save registers and CSR to stack
2. Send interrupt response to PLIC
3. Set the [MIE] bit field of the Configuration Machine Mode Status Register (MSTATUS) to 1 to enable global interrupts
4. Handling interrupts
5. Send interrupt completion to PLIC
6. Recovery registers and CSR
7. Returns from interrupt handler

In vector mode, the RISC-V core can enable interrupt preemption after saving the context. To avoid conflicts with the next interrupt response, global interrupt enable needs to be turned off before the execution interrupt completes. In addition, the FENCE io instruction should be executed once before exiting the interrupt handler to ensure that the interrupt completion operation has reached the PLIC.

The suggested vector mode interrupt handler is as follows:

1. Save registers and CSR to stack
2. The [MIE] bit field of the Configuration Machine Mode Status Register (MSTATUS) is set to 1 to enable global interrupts
3. Handling interrupts
4. Set the [MIE] bit field of the Configuration Machine Mode Status Register (MSTATUS) to 0 to disable global interrupts
5. Send interrupt completion to PLIC
6. Recovery registers and CSR
7. Execute the FENCE io command to ensure that the interrupt completion operation has been delivered to the PLIC
8. Returns from interrupt handler

The register THRESHOLD can set the interrupt priority threshold. Only interrupts exceeding this threshold will be notified to the RISC-V core. When an interrupt has been responded to but not completed, the THRESHOLD register will be updated in real time to reflect the priority of the current interrupt being processed, in order to block interrupts with lower priorities and allow higher-priority interrupts to preempt.

6.3 PLIC Register Description

PLIC is as follows:

PLIC base address: 0xE4000000

Offset	Register Name	Description	Reset Value
0x0000	FEATURE	Feature enable register	0x00000000
0x0004	PRIORITY[PRIORITY1]	Source priority	0x00000001
0x0008	PRIORITY[PRIORITY2]	Source priority	0x00000001

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

6 PLATFORM-LEVEL INTERRUPT

CONTROLLER PLIC

Offset	Register Name	Description	Reset Value
0x000C	PRIORITY[PRIORITY3]	Source priority	0x00000001
0x0010	PRIORITY[PRIORITY4]	Source priority	0x00000001
0x0014	PRIORITY[PRIORITY5]	Source priority	0x00000001
0x0018	PRIORITY[PRIORITY6]	Source priority	0x00000001
0x001C	PRIORITY[PRIORITY7]	Source priority	0x00000001
0x0020	PRIORITY[PRIORITY8]	Source priority	0x00000001
0x0024	PRIORITY[PRIORITY9]	Source priority	0x00000001
0x0028	PRIORITY[PRIORITY10]	Source priority	0x00000001
0x002C	PRIORITY[PRIORITY11]	Source priority	0x00000001
0x0030	PRIORITY[PRIORITY12]	Source priority	0x00000001
0x0034	PRIORITY[PRIORITY13]	Source priority	0x00000001
0x0038	PRIORITY[PRIORITY14]	Source priority	0x00000001
0x003C	PRIORITY[PRIORITY15]	Source priority	0x00000001
0x0040	PRIORITY[PRIORITY16]	Source priority	0x00000001
0x0044	PRIORITY[PRIORITY17]	Source priority	0x00000001
0x0048	PRIORITY[PRIORITY18]	Source priority	0x00000001
0x004C	PRIORITY[PRIORITY19]	Source priority	0x00000001
0x0050	PRIORITY[PRIORITY20]	Source priority	0x00000001
0x0054	PRIORITY[PRIORITY21]	Source priority	0x00000001
0x0058	PRIORITY[PRIORITY22]	Source priority	0x00000001
0x005C	PRIORITY[PRIORITY23]	Source priority	0x00000001
0x0060	PRIORITY[PRIORITY24]	Source priority	0x00000001
0x0064	PRIORITY[PRIORITY25]	Source priority	0x00000001
0x0068	PRIORITY[PRIORITY26]	Source priority	0x00000001
0x006C	PRIORITY[PRIORITY27]	Source priority	0x00000001
0x0070	PRIORITY[PRIORITY28]	Source priority	0x00000001
0x0074	PRIORITY[PRIORITY29]	Source priority	0x00000001
0x0078	PRIORITY[PRIORITY30]	Source priority	0x00000001
0x007C	PRIORITY[PRIORITY31]	Source priority	0x00000001
0x0080	PRIORITY[PRIORITY32]	Source priority	0x00000001
0x0084	PRIORITY[PRIORITY33]	Source priority	0x00000001
0x0088	PRIORITY[PRIORITY34]	Source priority	0x00000001
0x008C	PRIORITY[PRIORITY35]	Source priority	0x00000001
0x0090	PRIORITY[PRIORITY36]	Source priority	0x00000001
0x0094	PRIORITY[PRIORITY37]	Source priority	0x00000001
0x0098	PRIORITY[PRIORITY38]	Source priority	0x00000001
0x009C	PRIORITY[PRIORITY39]	Source priority	0x00000001
0x00A0	PRIORITY[PRIORITY40]	Source priority	0x00000001
0x00A4	PRIORITY[PRIORITY41]	Source priority	0x00000001
0x00A8	PRIORITY[PRIORITY42]	Source priority	0x00000001

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

6 PLATFORM-LEVEL INTERRUPT

CONTROLLER PLIC

Offset	Register Name	Description	Reset Value
0x00AC	PRIORITY[PRIORITY43]	Source priority	0x00000001
0x00B0	PRIORITY[PRIORITY44]	Source priority	0x00000001
0x00B4	PRIORITY[PRIORITY45]	Source priority	0x00000001
0x00B8	PRIORITY[PRIORITY46]	Source priority	0x00000001
0x00BC	PRIORITY[PRIORITY47]	Source priority	0x00000001
0x00C0	PRIORITY[PRIORITY48]	Source priority	0x00000001
0x00C4	PRIORITY[PRIORITY49]	Source priority	0x00000001
0x00C8	PRIORITY[PRIORITY50]	Source priority	0x00000001
0x00CC	PRIORITY[PRIORITY51]	Source priority	0x00000001
0x00D0	PRIORITY[PRIORITY52]	Source priority	0x00000001
0x00D4	PRIORITY[PRIORITY53]	Source priority	0x00000001
0x00D8	PRIORITY[PRIORITY54]	Source priority	0x00000001
0x00DC	PRIORITY[PRIORITY55]	Source priority	0x00000001
0x00E0	PRIORITY[PRIORITY56]	Source priority	0x00000001
0x00E4	PRIORITY[PRIORITY57]	Source priority	0x00000001
0x00E8	PRIORITY[PRIORITY58]	Source priority	0x00000001
0x00EC	PRIORITY[PRIORITY59]	Source priority	0x00000001
0x00F0	PRIORITY[PRIORITY60]	Source priority	0x00000001
0x00F4	PRIORITY[PRIORITY61]	Source priority	0x00000001
0x00F8	PRIORITY[PRIORITY62]	Source priority	0x00000001
0x00FC	PRIORITY[PRIORITY63]	Source priority	0x00000001
0x0100	PRIORITY[PRIORITY64]	Source priority	0x00000001
0x0104	PRIORITY[PRIORITY65]	Source priority	0x00000001
0x0108	PRIORITY[PRIORITY66]	Source priority	0x00000001
0x010C	PRIORITY[PRIORITY67]	Source priority	0x00000001
0x0110	PRIORITY[PRIORITY68]	Source priority	0x00000001
0x0114	PRIORITY[PRIORITY69]	Source priority	0x00000001
0x0118	PRIORITY[PRIORITY70]	Source priority	0x00000001
0x011C	PRIORITY[PRIORITY71]	Source priority	0x00000001
0x0120	PRIORITY[PRIORITY72]	Source priority	0x00000001
0x0124	PRIORITY[PRIORITY73]	Source priority	0x00000001
0x0128	PRIORITY[PRIORITY74]	Source priority	0x00000001
0x012C	PRIORITY[PRIORITY75]	Source priority	0x00000001
0x0130	PRIORITY[PRIORITY76]	Source priority	0x00000001
0x0134	PRIORITY[PRIORITY77]	Source priority	0x00000001
0x0138	PRIORITY[PRIORITY78]	Source priority	0x00000001
0x013C	PRIORITY[PRIORITY79]	Source priority	0x00000001
0x1000	PENDING[PENDING0]	Pending status	0x00000000
0x1004	PENDING[PENDING1]	Pending status	0x00000000
0x1008	PENDING[PENDING2]	Pending status	0x00000000

Offset	Register Name	Description	Reset Value
0x100C	PENDING[PENDING3]	Pending status	0x00000000
0x1080	TRIGGER[TRIGGER0]	Trigger type	0x00000000
0x1084	TRIGGER[TRIGGER1]	Trigger type	0x00000000
0x1088	TRIGGER[TRIGGER2]	Trigger type	0x00000000
0x108C	TRIGGER[TRIGGER3]	Trigger type	0x00000000
0x1100	NUMBER	Number of supported interrupt sources and targets	-
0x1104	INFO	Version and the maximum priority	-
0x2000	TARGETINT[TARGET0][INTEN][INTEN0]	machine interrupt enable	0x00000000
0x2004	TARGETINT[TARGET0][INTEN][INTEN1]	machine interrupt enable	0x00000000
0x2008	TARGETINT[TARGET0][INTEN][INTEN2]	machine interrupt enable	0x00000000
0x200C	TARGETINT[TARGET0][INTEN][INTEN3]	machine interrupt enable	0x00000000
0x2010	TARGETINT[TARGET0][INTEN][INTEN4]	machine interrupt enable	0x00000000
0x2014	TARGETINT[TARGET0][INTEN][INTEN5]	machine interrupt enable	0x00000000
0x2080	TARGETINT[TARGET1][INTEN][INTEN0]	supervisor interrupt enable	0x00000000
0x2084	TARGETINT[TARGET1][INTEN][INTEN1]	supervisor interrupt enable	0x00000000
0x2088	TARGETINT[TARGET1][INTEN][INTEN2]	supervisor interrupt enable	0x00000000
0x208C	TARGETINT[TARGET1][INTEN][INTEN3]	supervisor interrupt enable	0x00000000
0x2090	TARGETINT[TARGET1][INTEN][INTEN4]	supervisor interrupt enable	0x00000000
0x2094	TARGETINT[TARGET1][INTEN][INTEN5]	supervisor interrupt enable	0x00000000
0x200000	TARGETCONFIG[TARGET0][THRESHOLD]	Target0 priority threshold	0x00000000
0x200004	TARGETCONFIG[TARGET0][CLAIM]	Target claim and complete	0x00000000
0x200400	TARGETCONFIG[TARGET0][PPS]	Preempted priority stack	0x00000000
0x201000	TARGETCONFIG[TARGET1][THRESHOLD]	Target1 priority threshold	0x00000000
0x201004	TARGETCONFIG[TARGET1][CLAIM]	Target claim and complete	0x00000000
0x201400	TARGETCONFIG[TARGET1][PPS]	Preempted priority stack	0x00000000

Table 14: PLIC Register List

6.4 PLIC register details

The register of PLIC is detailed as follows:

6.4.1 FEATURE (0x0)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																														VECTORED	PREEMPT
N/A																														RW	RW
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

FEATURE [31:0]

Bits	Name	Description
1	VECTORED	Vector mode enable 0: Disabled 1: Enabled
0	PREEMPT	Preemptive priority interrupt enable 0: Disabled 1: Enabled

FEATURE Bits

6.4.2 PRIORITY (0x4 + 0x4 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRIORITY																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

PRIORITY [31:0]

Bits	Name	Description
31-0	PRIORITY	Interrupt source priority. The valid range of this field is 0-31. 0: Never interrupt 1-31: Interrupt source priority. The larger the value, the higher the priority.

PRIORITY Bits

6.4.3 PENDING (0x1000 + 0x4 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTERRUPT																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PENDING [31:0]

Bits	Name	Description
31-0	INTERRUPT	The interrupt pending status of interrupt sources. Every interrupt source occupies 1 bit.

PENDING Bits

6.4.4 TRIGGER (0x1080 + 0x4 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTERRUPT																															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TRIGGER [31:0]

Bits	Name	Description
31-0	INTERRUPT	The interrupt trigger type of interrupt sources. Every interrupt source occupies 1 bit. 0: Level-triggered interrupt 1: Edge-triggered interrupt

TRIGGER Bits

6.4.5 NUMBER (0x1100)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NUM_TARGET																NUM_INTERRUPT															
RO																RO															
*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

NUMBER [31:0]

Bits	Name	Description
31-16	NUM_TARGET	The number of supported targets
15-0	NUM_INTERRUPT	The number of supported interrupt sources

NUMBER Bits

6.4.6 INFO (0x1104)

CONTROLLER PLIC

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAX_PRIORITY																VERSION															
RO																RO															
*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

INFO [31:0]

Bits	Name	Description
31-16	MAX_PRIORITY	The maximum priority supported
15-0	VERSION	The version of the PLIC design

INFO Bits

6.4.7 TARGETINT[INTEN] (0x2000 + 0x80 * n + 0x4 * m)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTERRUPT																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TARGETINT[INTEN] [31:0]

Bits	Name	Description
31-0	INTERRUPT	The interrupt enable bit for interrupt. Every interrupt source occupies 1 bit.

TARGETINT[INTEN] Bits

6.4.8 TARGETCONFIG[THRESHOLD] (0x200000 + 0x1000 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
THRESHOLD																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TARGETCONFIG[THRESHOLD] [31:0]

Bits	Name	Description
31-0	THRESHOLD	Interrupt priority threshold.

TARGETCONFIG[THRESHOLD] Bits

6.4.9 TARGETCONFIG[CLAIM] (0x200004 + 0x1000 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																						INTERRUPT_ID									
N/A																						RW									
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	

TARGETCONFIG[CLAIM] [31:0]

Bits	Name	Description
9-0	INTERRUPT_ID	On reads, indicating the interrupt source that has being claimed. On writes, indicating the interrupt source that has been handled (completed).

TARGETCONFIG[CLAIM] Bits

6.4.10 TARGETCONFIG[PPS] (0x200400 + 0x1000 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRIORITY_PREEMPTED																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TARGETCONFIG[PPS] [31:0]

Bits	Name	Description
31-0	PRIORITY_PREE MPTED	Each bit indicates if the corresponding priority level has been preempted by a higher-priority interrupt.

TARGETCONFIG[PPS] Bits

7 Platform-level Software Interrupt Controller PLICSW

This chapter describes the functions and features of the platform-level software interrupt controller PLICSW.

7.1 Feature Summary

This chapter describes the main features of the platform-level interrupt controller PLIC:

- supports software programming to generate interrupts

7.2 Function Description

This section describes the functions of the platform-level interrupt controller PLIC.

The RISC-V core writes 1 to the register PENDING to trigger a software interrupt. After receiving the interrupt, the RISC-V core needs to read the CLAIM register to complete the interrupt response (claim). After the interrupt processing is completed, it needs to write the CLAIM register to perform the interrupt completion operation.

7.3 PLICSW Register Description

PLIC is as follows:

PLICSW base address: 0xE6400000

Offset	Register Name	Description	Reset Value
0x1000	PENDING	Pending status	0x00000000
0x2000	INTEN	Interrupt enable	0x00000000
0x200004	CLAIM	Claim and complete.	0x00000000

Table 15: PLIC_SW Register List

7.3.1 PLICSW register details

PLIC is as follows:

7.3.2 PENDING (0x1000)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
RSVD																																INTERRUPT	RSVD
N/A																																RW	N/A
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	x			

PENDING [31:0]

Bits	Name	Description
1	INTERRUPT	writing 1 to trigger software interrupt

PENDING Bits

7.3.3 INTEN (0x2000)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																															INTERRUPT
N/A																															RW
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0

INTEN [31:0]

Bits	Name	Description
0	INTERRUPT	enable software interrupt

INTEN Bits

7.3.4 CLAIM (0x200004)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																															INTERRUPT_ID
N/A																															RW
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0

CLAIM [31:0]

Bits	Name	Description
0	INTERRUPT_ID	On reads, indicating the interrupt source that has being claimed. On writes, indicating the interrupt source that has been handled (completed).

CLAIM Bits

8 Power Management

This chapter describes the power management system.

8.1 Power Supply System Structure

This product is divided into 2 main power domains in terms of power structure, as shown in [Figure 3](#) :

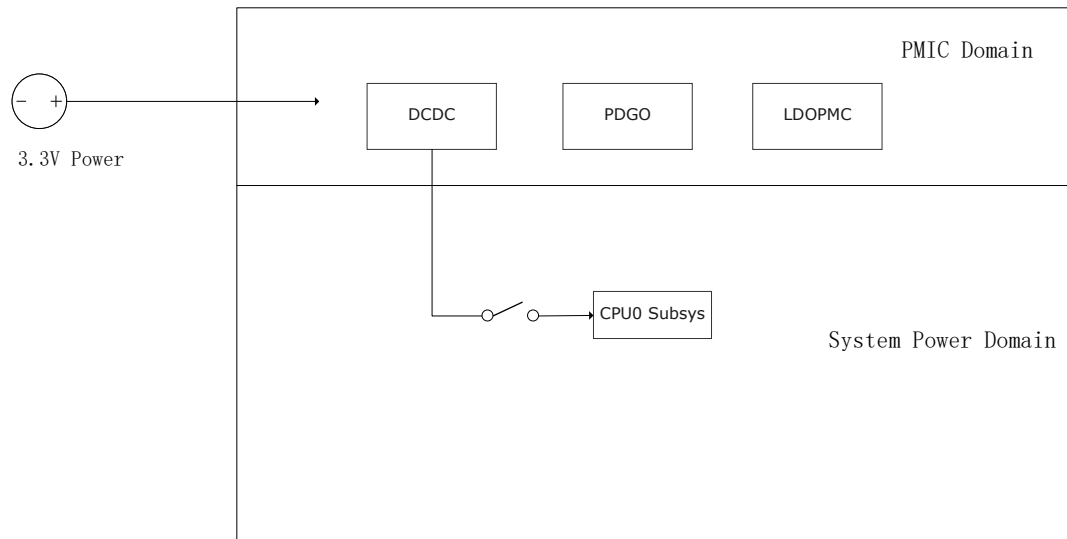


Figure 3: power domain diagram

Two power domains include:

- system power domain contains most of the functional modules in the chip, including the subsystem (CPU0) that can independently switch the power supply, and several other functional modules
- The power management domain has a DCDC that can be used to supply power to the system power domain, and several functional modules that can wake up the system, which can maintain low-power operation when the system power domain is turned off. The PDGO module in the power management domain can run with extremely low power consumption and save necessary data when other modules in the power management domain are turned off.

The main functional modules in each power domain can be viewed in [Figure 1](#) .

In typical application scenarios, the chip can work with a single 3.3V power supply.

8.2 Power Supply System

chip has 2 LDO linear power supplies and 1 DCDC switching power supply to support a single 3.3V power supply application. The block diagram of the

power supply system is as follows:

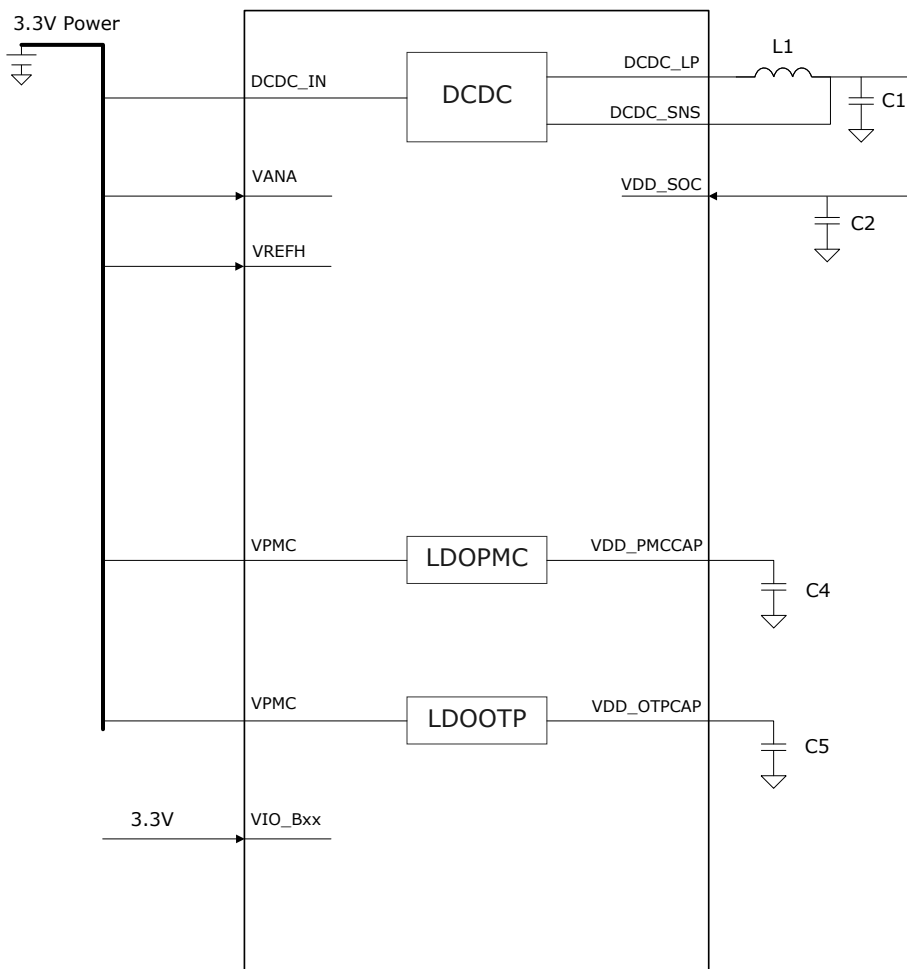


Figure 4: Block diagram of a power supply system using built-in DCDC

Note that when using built-in DCDC, 16-bit accuracy ADCs will be reduced to a certain extent; for precautions on the use of on-chip ADCs, please refer to the ADC-related application documentation at <https://www.hpmicro.com/design-resources/chip-information>.

chip supports using an external power supply to power the system power domain instead of using built-in DCDC, as shown in Figure [Figure 5](#) :

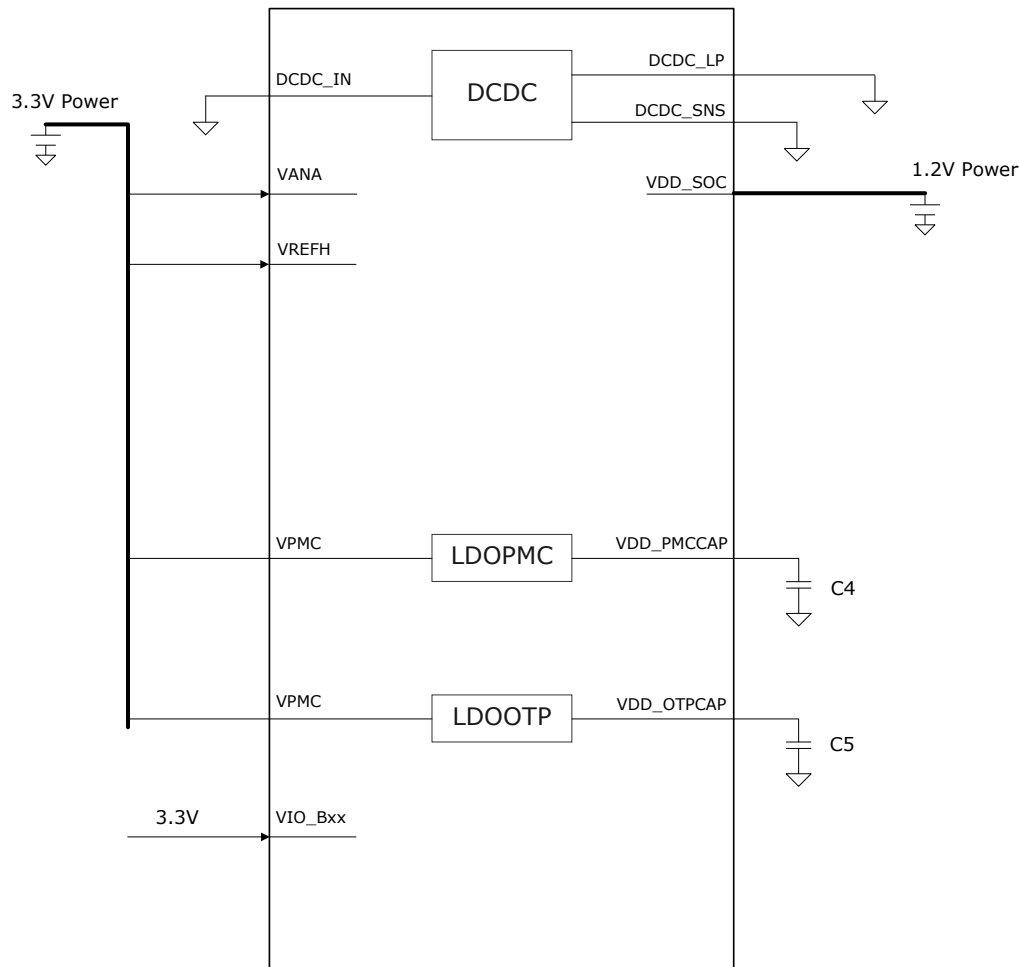


Figure 5: Block diagram of a power supply system without built-in DCDC

8.2.1 Switching Power Supply DCDC

DCDC is located in the power management domain and converts 3.3V external power supply into a typical 1.1V power supply, which can provide power to circuits in the system power domain. The

DCDC output voltage is adjustable and supports dynamic voltage frequency adjustment.

DCDC_IN is the power input pin of DCDC. DCDC_LP is the power output pin of DCDC and should be connected to a 4.7uH inductor. DCDC_SNS is the feedback pin of DCDC and should be connected to the power output of DCDC.

When users choose to use an external power supply instead of on-chip DCDC to supply power to the system power domain, it is recommended to connect the relevant pins of DCDC_IN, DCDC_LP and DCDC_SNS to ground through a 10k ohm resistor.

8.2.2 Power Management Domain Linear Regulator LDOPMC

LDOPMC is located in the power management domain and converts 3.3V external power supply into a typical 1.1V power supply to provide power to digital circuits other than PDGO modules in the power management domain.

LDOPMC uses the VPMC pin as the power input and requires a 4.7uF decoupling capacitor connected to

the VDD_PMCCAP pin.

8.2.3 OTP Linear Regulator LDOOTP

LDOOTP is located in the power management domain and converts 3.3V external power supply into a typical 2.5V power supply to provide power to the OTP programming circuit.

LDOOTP can only be enabled during OTP programming operations and should remain turned off at other times.

LDOOTP uses the VPMC pin as the power input and requires a 4.7uF decoupling capacitor connected to the VDD_OTPCAP pin.

8.3 IO power supply

chip are divided into multiple groups (Banks), and IO in the same IO Bank (Bxx) uses the same IO power supply pin VIO_B*. The Bank to which the IO belongs can be found in [Chapter 19](#). The correspondence between

IO Bank and VIO pins varies depending on the package model and can be viewed in [Table 17](#).

8.4 Power Supply Pin Notes

Power Pin	Voltage Range	Description
VPMC	3.0V ~ 3.6V	LDOPMC power input, LDOOTP power input, power supply to PDGO analog circuits and digital circuits in the power management domain
VDD_PMCCAP	-	The decoupling capacitor connection pin of LDOPMC supplies power to the digital circuits in the power management domain excluding PDGO
VDD_OTPCAP	-	LDOOTP's decoupling capacitor is connected to pin
VANA	3.0V ~ 3.6V	supplies power to ADC and ACMP analog circuits
VREFH	2.4V ~ VANA	ADC High Reference Voltage High-bit Reference Voltage of the DAC Built in ACMP
DCDC_IN	3.0V ~ 3.6V	DCDC Power Input
DCDC_LP	-	DCDC power output, which can power digital circuits in the system power domain
DCDC_SNS	-	DCDC power supply feedback pin should be connected with DCDC_LP
VDD_SOC	0.925V ~ 1.26V	System Power Domain Digital Circuit Supply Pin
VIO_B00 ~ VIO_B02	1.8V or 3.3V	IO Power Supply Pin
DCDC_GND	-	DCDC Ground Pin
VSS	-	Ground Pin
VSSA	-	Analog Circuit Ground Pin
VREFL	-	ADC low reference voltage, need to be grounded

Power Pin	Voltage Range	Description

Table 16: Power and Ground Pin Descriptions

The lead-out of power-related pins varies in different package models, and there may be situations where power pins are shared within the package. [Table 17](#) displays detailed information.

Internal power supply	Description	BGA196 pins	eLQFP100 pin	BGA121 pins
B00	IO Bank B00 Power Supply	VIO_B01	VIO_B01	VIO_B01
B01	IO Bank B01 Power Supply	VIO_B01	VIO_B01	VIO_B01
B02	IO Bank B02 Power Supply	VANA	VIO_B02	VIO_B02
DCDC_SNS	DCDC Power Feedback	VDD_SOC	DCDC_SNS	VDD_SOC
VUSB	USB Analog Circuits 3.3V Power Supply	VIO_B01	VIO_B01	VIO_B01
VPLL	PLL Analog Circuits 3.3V Power Supply	VIO_B01	VIO_B01	VIO_B01

Table 17: Power supply and package pin reuse

8.5 Overview of Low Power Consumption

This product supports multiple power consumption modes. Their power consumption levels range from high to low and wake-up time from short to long:

- Running Mode (RUN)
- WAIT
- Stop mode (STOP)
- Sleep mode (STANBY)
- Shutdown mode (SHUTDOWN)

operating mode, the CPU executes instructions normally, and all necessary functional modules work normally. Unneeded functional modules can be turned off, and the CPU clock frequency and power supply voltage can be adjusted by software. In the

wait mode, the CPU core clock is turned off due to the trigger of the WFI command, and other functional modules remain in the running mode. When an interrupt occurs, the CPU can immediately resume operation and handle the interrupt. The

stop mode is triggered by the WFI command of the CPU. Through pre-configuration, the power supply of all subsystems and modules in the system power domain, including the CPU itself, can be flexibly turned off or maintained under the control of the SYSCTL module. In

sleep mode, the entire system power domain is in a reset or power down state. This mode can be triggered by the WFI command of the CPU or through software operation. In the

shutdown mode, all parts of the system power domain and power management domain except the PDGO module are in a reset or power-down state, and only PDGO is left working. This mode can be triggered through software operation.

8.6 IO related to power management functions

The PDGO module in the power management domain is the foundation of the chip power management system, and its IO can be used as related functions of power management:

pin	Default features	Description
RESETN	RESETN	global reset input pin, which can reset the entire system, or can be used to exit in shutdown mode
WAKEUP	WAKEUP	Power wake-up input pin, which can be used to exit in shutdown mode and other low-power modes

Table 18: Power Management Related IO

9 System Reset

This chapter describes the system reset of this product.

9.1 Reset Overview

reset system supports resetting different ranges of the chip.

global reset resets the power management domain and the system power domain as a whole. The reset of the power domain of the

system does not affect the power management domain.

9.2 Global Reset

The global reset will reset the power management domain and the system power domain. There are two situations that will cause the global reset to occur:

- Power-up of the VPMC power supply pin or triggering low voltage detection
- The RESETN pin is pulled low by an external circuit

RESETN pin generates a global reset by default. The reset range can be modified by configuring the PDGO module register so that the PDGO state in the power management domain is not affected. See [Section 11.3](#) for details.

9.3 Reset the system power domain

Reset of the system power domain

- Cold Reset: The system power domain retains the value of the OTP shadow register and does not reset the DEBUG-related circuits.

There are multiple reset sources for resetting the system power domain, including:

- VPMC supply voltage is too low (Brownout)
- Debug Reset: Reset request issued by an external debugger
- Security Violations
- Watchdog timed out
- Software Reset

In addition to the reset triggered by the above reset sources, the software can also individually reset a subsystem by controlling the SYSCTL module. See the relevant chapters of SYSCTL for details.

10 Clock System

This chapter describes the clock system of this product.

10.1 Clock System Overview

clock system mainly consists of a clock source and a functional clock, and its structure is shown in Figure 6 .

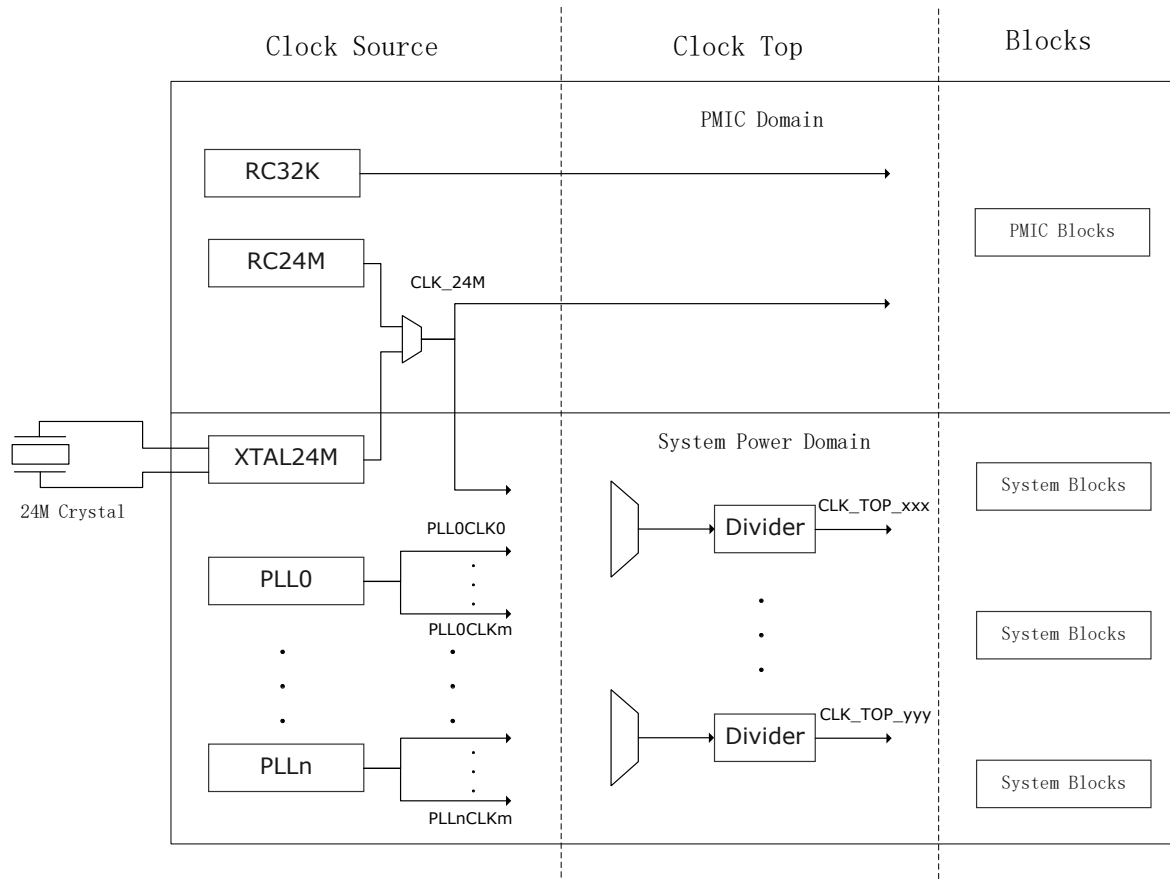


Figure 6: Clock System Diagram

clock source includes external crystal oscillator XTAL, internal RC oscillator, phase-locked loop PLL, etc. The clock source can generate clocks with different frequencies and different precisions.

function clock CLK_TOP_* is the clock after selecting and dividing the clock source.

10.2 Clock Source

10.2.1 32KHz clock source CLK_32K

The modules that generate CLK_32K clock are:

- RC32K-Internal 32KHz RC oscillator

CLK_32K directly provides a working clock for the PDGO module in the power management domain.

10.2.2 24MHz clock source CLK_24M

The modules that generate a 24MHz clock source are:

- XTAL24M- 24MHz high-speed oscillator, supports 24MHz external crystal oscillator, and also supports external input of 24 MHz active clock through pins
- RC24M-Internal 24MHz RC oscillator

M and RC24M together form the CLK_24M clock source. CLK_24M comes from the output clock of XTAL24M when XTAL24M is enabled and frequency stable, and from the output clock of RC24M when XTAL24M is turned off or has not yet stabilized. Switching between the two clocks occurs automatically.

XTAL24M will be automatically enabled by default. Please refer to [Chapter 13](#) for its switching control in low power mode.

CLK_24M provides clocks directly to modules in the power management domain.

10.2.3 Phase-locked loop PLL

product has 3 built-in PLLs, each PLL has multiple clock outputs, and a total of 7 clock sources are used to generate clocks of various frequencies required by the system. The

PLL uses the XTAL24M clock as the reference clock by default. The default operating frequency has been preset for each PLL, as shown in [Table 19](#) : For the modification method of the frequency and frequency division coefficient of

PLL, please refer to [Chapter 14](#) .

10.3 Function Clock

functional clock can select and divide multiple clock sources according to configuration, and then provide running clocks to each module in the system's power domain. There are 8 clock sources in the

system, so each functional clock has a 1-out-8 multiplexer to select the clock source. The correspondence between the selection values and the clock source is shown in [Table 19](#) .

Select value	Clock Source	Default frequency	Description
0	CLK_24M	24 MHz	24 MHz Clock Source
1	PLL0CLK0	600MHz	Default as processor clock source
2	PLL0CLK1	375MHz	
3	PLL1CLK0	400MHz	Default as clock source for bus and basic peripherals
4	PLL1CLK1	333.33MHz	
5	PLL1CLK2	266.67MHz	
6	PLL2CLK0	500MHz	
7	PLL2CLK1	250MHz	

Table 19: Clock source selection and default frequency

The frequency division factor can be set separately for each functional clock to divide the frequency of the selected clock source. The frequency division factor ranges from any integer from 1 to 256.

For ease of use, clock source selection and frequency division coefficients have been preset for each functional clock. See [Table 20](#) for details. Please refer to [Section 13.2](#) for clock source selection and frequency

division coefficient modification methods.

Function clock name	Default clock source	Default frequency division coefficient	Default frequency	Application
CLK_TOP_CPU0	1 (PLL0 CLK0)	1	300 MHz	CPU0 Core/ILM0/DLM0/FGPIO0
CLK_TOP_MCT0	0 (CLK_24M)	0	24MHz	MCHTMR0 Timing
CLK_TOP_AHB0	3 (PLL 1CLK0)	1	200 MHz	AHB System Bus
CLK_TOP_AXIF	3 (PLL 1CLK0)	1	200MHz	AXI High-Speed Bus
CLK_TOP_AXIS	3 (PLL 1CLK0)	1	200MHz	AXI System Bus
CLK_TOP_AXIC	3 (PLL 1CLK0)	1	200 MHz	CONN Subsystem Bus
CLK_TOP_TMR0	3 (PLL 1CLK0)	3	100 MHz	GPTMR0 Timing
CLK_TOP_TMR1	3 (PLL 1CLK0)	3	100 MHz	GPTMR1 Timing
CLK_TOP_TMR2	3 (PLL 1CLK0)	3	100MHz	GPTMR2 Timing
CLK_TOP_TMR3	3 (PLL 1CLK0)	3	100MHz	GPTMR3 Timing
CLK_TOP_OWR0	3 (PLL 1CLK0)	4	80 MHz	OWR0 Interface
CLK_TOP_OWR1	3 (PLL 1CLK0)	4	80 MHz	OWR1 Interface
CLK_TOP_I2C0	3 (PLL 1CLK0)	4	80 MHz	I2C0 Interface
CLK_TOP_I2C1	3 (PLL1CLK0)	4	80 MHz	I2C1 Interface
CLK_TOP_I2C2	3 (PLL 1CLK0)	4	80MHz	I2C2 Interface
CLK_TOP_I2C3	3 (PLL 1CLK0)	4	80MHz	I2C3 Interface
CLK_TOP_SPI0	3 (PLL 1CLK0)	4	80 MHz	SPI0 Interface
CLK_TOP_SPI1	3 (PLL1CLK0)	4	80 MHz	SPI1 Interface
CLK_TOP_SPI2	3 (PLL 1CLK0)	4	80 MHz	SPI2 Interface
CLK_TOP_SPI3	3 (PLL 1CLK0)	4	80 MHz	SPI3 Interface
CLK_TOP_URT0	3 (PLL 1CLK0)	4	80 MHz	UART0 Interface
CLK_TOP_URT1	3 (PLL 1CLK0)	4	80 MHz	UART1 Interface
CLK_TOP_URT2	3 (PLL 1CLK0)	4	80 MHz	UART2 Interface
CLK_TOP_URT3	3 (PLL 1CLK0)	4	80 MHz	UART3 Interface
CLK_TOP_URT4	3 (PLL 1CLK0)	4	80 MHz	UART4 Interface
CLK_TOP_URT5	3 (PLL 1CLK0)	4	80 MHz	UART5 Interface
CLK_TOP_URT6	3 (PLL 1CLK0)	4	80MHz	UART6 Interface
CLK_TOP_URT7	3 (PLL 1CLK0)	4	80 MHz	UART7 Interface
CLK_TOP_CAN0	3 (PLL 1CLK0)	4	80 MHz	CAN0 Interface
CLK_TOP_CAN1	3 (PLL 1CLK0)	4	80 MHz	CAN1 Interface
CLK_TOP_CAN2	3 (PLL 1CLK0)	4	80 MHz	CAN2 Interface

Function clock name	Default clock source	Default frequency division coefficient	Default frequency	Application
CLK_TOP_CAN3	3 (PLL 1CLK0)	4	80 MHz	CAN3 Interface
CLK_TOP_XPI0	5 (PLL 1CLK2)	0	266 MHz	XPI0 Interface
CLK_TOP_ESC0	6 (PLL 2CLK0)	4	100 MHz> ESC Master Clock	
CLK_TOP_ETH0	7 (PLL 2CLK1)	1	125 MHz	ENET0 Interface
CLK_TOP_PTP0	3 (PLL 1CLK0)	1	200 MHz	PTP0 Timing
CLK_TOP_REF0	6 (PLL2CLK0)	9	50MHz	REF0
CLK_TOP_REF1	6 (PLL2CLK0)	9	50MHz	REF1
CLK_TOP_NTM0	6 (PLL 2CLK0)	4	100 MHz	NTM0 Timing
CLK_TOP_ANA0	3 (PLL1CLK0)	3	100 MHz	ADC0 Alternative Clock
CLK_TOP_ANA1	3 (PLL1CLK0)	3	100 MHz	ADC1 Alternative Clock

Table 20: Function Clock Summary

* The actual frequency depends on the startup configuration.

ADC adopts a two-stage multi-selection structure, that is, each ADC can select a functional clock from CLK_TOP_AHB0 and CLK_TOP_ANAX. When the former is selected, the ADC has the same frequency as the system bus and can synchronize with other peripherals. When the latter is selected, an independent clock frequency can be used. The functional clock structure of the

ADC is shown in [Figure 7](#).

CLK_TOP_[BUS] in the figure corresponds to CLK_TOP_AHB0 in this product. The default configuration selects CLK_TOP_[BUS].

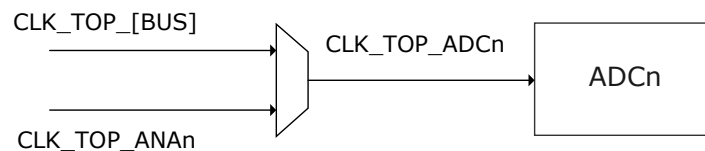


Figure 7: ADC functional clock structure

10.4 Module that directly uses the clock source

system do not use functional clocks but are directly connected to the clock source. Among them, all modules in the power management domain and some modules in the system power domain are listed in [Table 21](#).

module	Clock Source
Power Management Domain Module	
PDGO	CLK_32K
PTMR	CLK_24M
PWDG	CLK_24M and CLK_32K
PUART	CLK_24M
PGPIO	CLK_24M
System Power Domain Module	
WDG0 ~ WDG1	CLK_32K and bus clock
USB0	CLK_24M

Table 21: Module Clock List

11 Power Management Domain Switch Module PDGO

This chapter describes the main functions and uses of the switch module PDGO in the power management domain.

11.1 Main features

PDGO is powered by 3.3V VPMC. Its main functions include:

- 32KHz clock source management
- Global reset pin RESETN control
- Power Management Domain Switch Control
- supports software wake-up function: single wake-up or periodic wake-up
- supports 8 set of 32-bit general purpose registers

11.2 CLK_32K Clock Source Management

power management domain. After the system is powered on in the power management domain, PDGO will use the IRC 32K clock as the CLK_32K clock source to work.

can calibrate the accuracy of IRC 32K through Fuse, and users can also calibrate the accuracy of IRC 32K by setting the DGO_RC32K_CFG register.

11.3 Global reset pin RESETN control

The global reset pin RESETN is a dedicated pin. The user can set the RET bit of the DGO_CTRL register to 1. In this way, the pin reset only resets parts other than PDGO in the system power domain and power management domain, and does not affect the status of PDGO.

In addition, the global reset pin can also be used as a wake-up source to exit shutdown mode. See [Subsection 11.4.2](#) for details.

11.4 Power Management Domain Switch Control

PDGO can control entry and exit of shutdown mode. In shutdown mode, the entire system power domain and parts of the power management domain except the PDGO module are in a reset or power-down state, leaving only PDGO working. In this mode, the system has extremely low power consumption.

11.4.1 Entering shutdown mode

The way to enter shutdown mode is triggered by software operation. If a non-zero count value is written to the COUNTER bit of the DGO_TURNOFF register, the counter is decremented by 1 every 24M clock cycle until the counter is 1, and the system enters shutdown mode. After entering shutdown mode, this register is reset.

11.4.2 Exit of shutdown mode

There are the following ways to exit shutdown mode:

- Software counts down a single wake-up or periodic wake-up
- Wake-up Pin WAKEUP
- Reset Pin RESETN

11.4.2.1 Software Countdown Single or Periodic Wake-up

Before performing the shutdown operation, write the target wake-up time (non-zero) into the software wake-up register WAKEUP[COUNTER]. If you need to turn on the periodic wake-up function (cyclic wake-up mode), you also need to write the wake-up period (interval) into the WKUP_CYC[INTERVAL] register. Then enable software wake-up by configuring the WKUP_EN[CYC] and IRQ0_EN[CYC] registers. The software wake-up counter is decremented by 1 every clock cycle (internal 32KHz RC clock). When the count reaches 1, a wake-up event will be generated.

During a single wake-up, the counter stops when it drops to 0. The user can also turn off software wake-up in advance by writing 0 directly to the WAKEUP[COUNTER] register.

During cyclic wake-up, after the counter is reduced to 0, the counting of the next wake-up cycle will automatically restart, and the cycle will go back and forth; if you need to turn off periodic wake-up, you can write 0 to the WAKEUP[COUNTER] and WKUP_CYC[INTERVAL] registers.

Since the countdown to wake-up is not calculated from the time of shutdown, but from the time when the WAKEUP[COUNTER] register is configured, the user needs to control the time difference between the shutdown countdown and the wake-up countdown.

11.4.2.2 pin WAKEUP wake-up

The pin wake-up function can be enabled and enabled by configuring the WKUP_EN[WKUP] and IRQ0_EN[WKUP] registers. When the system is in shutdown mode, if a valid high level is detected on the wake-up pin WAKEUP, a wake-up event can be generated, thus exiting shutdown mode. The WAKEUP pin turns on the pull-down function by default, and the user can configure the pull-down function on or off through the PIN_CTRL[WKUP] register.

11.4.2.3 Reset Pin RESETN Wake-up

When the system is in shutdown mode, if a valid low level is detected on the reset pin RESETN, a wake-up event can be generated, thereby exiting shutdown mode. The RESETN pin turns on the pull-up function by default, and the user can configure the pull-up function on or off through the PIN_CTRL[RSTN] register.

Note: Wake-up events will only take effect when the system enters shutdown mode.

11.5 General purpose register

PDGO has a 32-bit general GPR register in 8 group, which supports saving user data when a RESETN reset is generated. At this time, the RET bit field of the DGO_CTRL register needs to be set to 1 in advance. It should be noted that it is also necessary to ensure that the PDGO is powered on.

11.6 PDGO register description

PDGO is as follows:

PDGO base address: 0xF4134000

Offset	Register Name	Description	Reset Value
0x0000	DGO_TURNOFF	Trunoff Control Register	0x00000000

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

11 POWER MANAGEMENT DOMAIN SWITCH

MODULE PDGO

Offset	Register Name	Description	Reset Value
0x0004	DOGO_RC32K_CFG	RC32K CLOCK TRIM Register	0x00000000
0x0008	DOGO_CTRL	Control Register	0x00000000
0x0010	PIN_CTRL	IO Control Register	0x00000000
0x0014	PIN_STS	IO Status Register	0x00000000
0x0018	WAKEUP	Wakeup Register	0x00000000
0x001C	WKUP_CYC	Wakeup Cycle Register	0x00000000
0x0020	WKUP_EN	Wakeup Enable Register	0x00000000
0x0024	WKUP_STS	Wakeup Status Register	0x00000000
0x0060	IRQ0_EN	Interrupt 0 Enable Register	0x00000000
0x0064	IRQ0_STS	Interrupt 0 Status Register	0x00000000
0x0068	IRQ1_EN	Interrupt 1 Enable Register	0x00000000
0x006C	IRQ1_STS	Interrupt 1 Status Register	0x00000000
0x0080	GPR[0]	General Purpose Register	0x00000000
0x0084	GPR[1]	General Purpose Register	0x00000000
0x0088	GPR[2]	General Purpose Register	0x00000000
0x008C	GPR[3]	General Purpose Register	0x00000000
0x0090	GPR[4]	General Purpose Register	0x00000000
0x0094	GPR[5]	General Purpose Register	0x00000000
0x0098	GPR[6]	General Purpose Register	0x00000000
0x009C	GPR[7]	General Purpose Register	0x00000000
0x0100	IOFILTER[0]	WUIO Filter Register	0x00000000
0x0104	IOFILTER[1]	WUIO Filter Register	0x00000000
0x0108	IOFILTER[2]	WUIO Filter Register	0x00000000
0x010C	IOFILTER[3]	WUIO Filter Register	0x00000000
0x0200	PCAP[0][CTRL]	Pulse Capture Control Register	0x00000000
0x0204	PCAP[0][CP_CFG]	Compare and Period Configuration Register	0x00000000
0x0210	PCAP[0][COUNTER]	Counter Register	0x00000000
0x0214	PCAP[0][DUMP]	DUMP Register	0x00000000
0x0280	PCNT[0][CNT_CFG]	Pulse Count Configuration Register	0x00000000
0x0284	PCNT[0][CALIB_CFG]	Calibration Configuration Register	0x00000000
0x0288	PCNT[0][RESOLUTION]	Resolution Configuration Register	0x00000000
0x028C	PCNT[0][HOMING_CFG]	Homing Configuration Register	0x00000000
0x0290	PCNT[0][ANALYSE_CFG]	Analyse Configuration Register	0x00000000
0x0294	PCNT[0][DIRECTION]	Direction Configuration Register	0x00000000
0x0298	PCNT[0][CMP0]	Compare0 Configuration Register	0x00000000

Offset	Register Name	Description	Reset Value
0x029C	PCNT[0][CMP1]	Compare1 Configuration Register	0x00000000
0x02A0	PCNT[0][CNT]	Counter Register	0x00000000
0x02A4	PCNT[0][TACHO]	Tacho Register	0x00000000
0x02A8	PCNT[0][DUMP]	DUMP Register	0x00000000
0x02AC	PCNT[0][STS]	Status Register	0x00000000

Table 22: PDGO Register List

11.7 PDGO register details

PDGO are as follows:

11.7.1 DGO_TURNOFF (0x0)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNTER																WO															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DGO_TURNOFF [31:0]

Bits	Name	Description
31-0	COUNTER	turnoff counter, counter stops when it counts down to 0, the turnoff occurs when the counter value is 1.

DGO_TURNOFF Bits

11.7.2 DGO_RC32K_CFG (0x4)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNTER	RSVD								CAPEX7_TRIM	CAPEX6_TRIM	RSVD											CAP_TRIM									
RW	N/A								RW	RW	N/A											RW									
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DGO_RC32K_CFG [31:0]

MODULE PDGO

Bits	Name	Description
31	COUNTER	IRC32K trim happened, this bit set by hardware after trim value loaded, and stop load, write 0 will clear this bit and reload trim value 0: irc is not trimmed 1: irc is trimmed
23	CAPEX7_TRIM	IRC32K bit 7
22	CAPEX6_TRIM	IRC32K bit 6
8-0	CAP_TRIM	capacitor trim bits

DGO_RC32K_CFG Bits

11.7.3 DGO_CTRL (0x8)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RSVD															RET	RSVD															LESP	LOWP
N/A															RW	N/A															RW	RW
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

DGO_CTRL [31:0]

Bits	Name	Description
16	RET	Retention, retain contents in DGO.GPR[n] register after RSTN (When PMIC has power) 0: Disable 1: Enable
1	LESP	Bandgap works in power save mode, bandgap function normally 0: bandgap works in high performance mode 1: bandgap works in power saving mode
0	LOWP	Bandgap works in low power mode, bandgap function limited 0: bandgap works in normal mode 1: bandgap works in low power mode

DGO_CTRL Bits

11.7.4 PIN_CTRL (0x10)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD							RSTN	RSVD							WKUP	RSVD										WUIO3	WUIO2	WUIO1	WUIO0		

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

11 POWER MANAGEMENT DOMAIN SWITCH

MODULE PDGO

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
N/A							RW	N/A							RW	N/A												RW	RW	RW	RW
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PIN_CTRL [31:0]

Bits	Name	Description
24	RSTN	resetrn pin pull up disable 0: Enable Pull Up 1: Disable Pull Up
16	WKUP	wakeup pin pull down disable 0: Enable Pull Down 1: Disable Pull Down
3	WUIO3	WUIO pin pull down disable 0: Enable Pull Down 1: Disable Pull Down
2	WUIO2	WUIO pin pull down disable 0: Enable Pull Down 1: Disable Pull Down
1	WUIO1	WUIO pin pull down disable 0: Enable Pull Down 1: Disable Pull Down
0	WUIO0	WUIO pin pull down disable 0: Enable Pull Down 1: Disable Pull Down

PIN_CTRL Bits

11.7.5 PIN_STS (0x14)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMOD	RSVD						RSTN	RSVD						WKUP	RSVD												WUIO3	WUIO2	WUIO1	WUIO0	
RO	N/A						RO	N/A						RO	N/A												RO	RO	RO	RO	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PIN_STS [31:0]

Bits	Name	Description
31	TMOD	TMOD Pin Status
24	RSTN	RSTN Pin Status
16	WKUP	WKUP pin status

Bits	Name	Description
3	WUIO3	WUIO3 pin status
2	WUIO2	WUIO2 pin status
1	WUIO1	WUIO1 pin status
0	WUIO0	WUIO0 pin status

PIN_STS Bits

11.7.6 WAKEUP (0x18)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNTER																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

WAKEUP [31:0]

Bits	Name	Description
31-0	COUNTER	software wake-up counter, counter stops when it counts down(base on 32KHz clock) to 0, the wakeup occurs when the counter value is 1. wakeup once: write counter value here and set CYCLE register to be 0. wakeup in cycle: write wakeup cycle in CYCLE register

WAKEUP Bits

11.7.7 WKUP_CYC (0x1C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTERVAL																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

WKUP_CYC [31:0]

Bits	Name	Description
31-0	INTERVAL	interval between wakeups, count by 32KHz clock cycles.

WKUP_CYC Bits

11.7.8 WKUP_EN (0x20)

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

11 POWER MANAGEMENT DOMAIN SWITCH

MODULE PDGO

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																RSVD	RSVD						PCNT0	RSVD		RSVD	PCAP0	RSVD	WUIO	CYC	WKUP
N/A																N/A	N/A						RW	N/A		N/A	RW	N/A	RW	RW	RW
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

WKUP_EN [31:0]

Bits	Name	Description
8	PCNT0	Allow wake-up via Pulse PCNT0 events. 0: Disable 1: Enable Note: wake-up interrupt request shloud be enable at the same time.
4	PCAP0	Allow wake-up via PCAP0 events. 0: Disable 1: Enable Note: wake-up interrupt request shloud be enable at the same time.
2	WUIO	Allow wake-up via WU00/WU01/WU02/WU03 pins. 0: Disable 1: Enable Note: wake-up interrupt request shloud be enable at the same time.
1	CYC	Allow software wake-up via counter cycle. 0: Disable 1: Enable Note: wake-up interrupt request shloud be enable at the same time.
0	WKUP	Allow wake-up via the WKUP pin. 0: Disable 1: Enable Note: wake-up interrupt request shloud be enable at the same time.

WKUP_EN Bits

11.7.9 WKUP_STS (0x24)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	RSVD															RSVD	RSVD						PCNT0	RSVD		RSVD	PCAP0	RSVD	WUIO	CYC	WKUP

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

11 POWER MANAGEMENT DOMAIN SWITCH

MODULE PDGO

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
N/A	N/A														N/A	N/A							W1C	N/A		N/A	W1C	N/A	W1C	W1C	W1C
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

WKUP_STS [31:0]

Bits	Name	Description
8	PCNT0	PCNT0 events wake-up status. 0: wake-up is invalid 1: wake-up is valid
4	PCAP0	PCAP0 events wake-up status. 0: wake-up is invalid 1: wake-up is valid
2	WUIO	WUIO0/WUIO1/WUIO2/WUIO3 pins wake-up status. 0: wake-up is invalid 1: wake-up is valid
1	CYC	counter cycle software wake-up status. 0: wake-up is invalid 1: wake-up is valid
0	WKUP	WKUP pin wake-up status. 0: wake-up is invalid 1: wake-up is valid

WKUP_STS Bits

11.7.10 IRQ0_EN (0x60)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD														CYC	WKUP	RSVD										WUIO3	WUIO2	WUIO1	WUIO0		
N/A														RW	RW	N/A										RW	RW	RW	RW		
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

IRQ0_EN [31:0]

Bits	Name	Description
17	CYC	Allow software counter cycle wake-up interrupt request. 0: Disable 1: Enable
16	WKUP	Allow WKUP Pin wake-up interrupt request. 0: Disable 1: Enable

Bits	Name	Description
3	WUIO3	Allow WUIO3 Pin wake-up interrupt request. 0: Disable 1: Enable
2	WUIO2	Allow WUIO2 wake-up interrupt request. 0: Disable 1: Enable
1	WUIO1	Allow WUIO1 wake-up interrupt request. 0: Disable 1: Enable
0	WUIO0	Allow WUIO0 wake-up interrupt request. 0: Disable 1: Enable

IRQ0_EN Bits

11.7.11 IRQ0_STS (0x64)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD														CYC	WKUP	RSVD										WUIO3	WUIO2	WUIO1	WUIO0		
N/A														W1C	W1C	N/A										W1C	W1C	W1C	W1C		
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

IRQ0_STS [31:0]

Bits	Name	Description
17	CYC	software counter cycle wake-up interrupt status. 0: is invalid 1: is valid
16	WKUP	WKUP Pin wake-up interrupt status. 0: is invalid 1: is valid
3	WUIO3	WUIO3 Pin wake-up interrupt status. 0: is invalid 1: is valid
2	WUIO2	WUIO2 wake-up interrupt status. 0: is invalid 1: is valid
1	WUIO1	WUIO1 wake-up interrupt status. 0: is invalid 1: is valid

Bits	Name	Description
0	WUIO0	WUIO0 wake-up interrupt status. 0: is invalid 1: is valid

IRQ0_STS Bits

11.7.12 IRQ1_EN (0x68)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								PCNT0_SEQERR	PCNT0_DECERR	PCNT0_TACHO	RSVD	PCNT0_MATCH0	PCNT0_HOME	PCNT0_Z	PCNT0_UPDATE	RSVD								RSVD			RSVD	PCAP0_MATCH	PCAP0_DUMP	PCAP0_PERIOD	
								N/A	RW	RW	RW	N/A	RW	RW	RW												RW	N/A			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				

IRQ1_EN [31:0]

Bits	Name	Description
23	PCNT0_SEQERR	Allow pulse counter interrupt request when phase sequence error happens. 0: Disable 1: Enable
22	PCNT0_DECERR	Allow pulse counter interrupt request when decoder error happens. 0: Disable 1: Enable
21	PCNT0_TACHO	Allow pulse counter interrupt request when tacho done happens. 0: Disable 1: Enable
19	PCNT0_MATCH0	Allow pulse counter interrupt request when match 0 events happens. 0: Disable 1: Enable
18	PCNT0_HOME	Allow pulse counter interrupt request when Phase h (HOME) edges or level match events happens. 0: Disable 1: Enable
17	PCNT0_Z	Allow pulse counter interrupt request when Phase Z (INDEX) edges or level match events happens. 0: Disable 1: Enable

Bits	Name	Description
16	PCNT0_UPDATE	Allow pulse counter interrupt request when counter updates. 0: Disable 1: Enable
2	PCAP0_MATCH	Allow PCAP0 interrupt request when match events happens. 0: Disable 1: Enable
1	PCAP0_DUMP	Allow PCAP0 interrupt request when dump events happens. 0: Disable 1: Enable
0	PCAP0_PERIOD	Allow PCAP0 interrupt request when period done. 0: Disable 1: Enable

IRQ1_EN Bits

11.7.13 IRQ1_STS (0x6C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								PCNT0_SEQERR	PCNT0_DECERR	PCNT0_TACHO	RSVD	PCNT0_MATCH0	PCNT0_HOME	PCNT0_Z	PCNT0_UPDATE	RSVD								RSVD	RSVD	PCAP0_MATCH	PCAP0_DUMP	PCAP0_PERIOD			
N/A								W1C	W1C	W1C	N/A	W1C	W1C	W1C	W1C	N/A								N/A	N/A	W1C	W1C	W1C			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

IRQ1_STS [31:0]

Bits	Name	Description
23	PCNT0_SEQERR	pulse counter interrupt status when phase sequence error happens. 0: is invalid 1: is valid
22	PCNT0_DECERR	pulse counter interrupt status when decoder error happens. 0: is invalid 1: is valid
21	PCNT0_TACHO	pulse counter interrupt status when tacho done happens. 0: is invalid 1: is valid
19	PCNT0_MATCH0	pulse counter interrupt status when match 0 events happens. 0: is invalid 1: is valid

Bits	Name	Description
18	PCNT0_HOME	pulse counter interrupt status when Phase H (HOME) edges or level match events happens. 0: is invalid 1: is valid
17	PCNT0_Z	pulse counter interrupt status when Phase Z (INDEX) edges or level match events happens. 0: is invalid 1: is valid
16	PCNT0_UPDATE	pulse counter interrupt status when counter updates. 0: is invalid 1: is valid
2	PCAP0_MATCH	PCAP0 interrupt status when match events happens. 0: is invalid 1: is valid
1	PCAP0_DUMP	PCAP0 interrupt status when dump events happens. 0: is invalid 1: is valid
0	PCAP0_PERIOD	PCAP0 interrupt status when period done. 0: is invalid 1: is valid

IRQ1_STS Bits

11.7.14 GPR (0x80 + 0x4 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPR [31:0]

Bits	Name	Description
31-0	VALUE	Value

GPR Bits

11.7.15 IOFILTER (0x100 + 0x4 * n)

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

11 POWER MANAGEMENT DOMAIN SWITCH

MODULE PDGO

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EN	RSVD					IRQMODE	RSVD		NORM	INV	RSVD	BYPASS	MODE		RSVD					FILTER_LEN											
RW	N/A					RW	N/A		RW	RW	N/A	RW	RW		N/A					RW											
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOFILTER [31:0]

Bits	Name	Description
31	EN	Allow this IO filter channel output 0: Disable 1: Enable
25-24	IRQMODE	Filter IRQ Mode: 0: Disable 1: Level Mode 2: Rising Edge 3: Rising Edge or Falling Edge
21	NORM	Initial output value
20	INV	Invert Filter output result.
18	BYPASS	Bypass Mode 0: Disable Bypass Mode 1: Enable Bypass Mode, use filter input as result output
17-16	MODE	Filter Mode: 0: Rapid change mode 1: Delay filter mode 2: Stable low mode 3: Stable high mode
9-0	FILTER_LEN	Filter Length(Base on 32K clock)

IOFILTER Bits

11.7.16 PCAP[CTRL] (0x200 + 0x18 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EN	RSVD					SELECT	RSVD		MATCH_MODE		RSVD					DUMP_MODE		RSVD		COUNT_MODE					RSVD					COUNT_MODE	
RW	N/A					RW	N/A		RW		N/A					RW		N/A		RW					N/A					RW	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCAP[CTRL] [31:0]

Bits	Name	Description
31	EN	Enable PCAP counter 0: Disable 1: Enable Note: Other registers should be configured before enable this bit.
25-24	SELECT	Select Input for counter: 0: Use WUIO0 as input 1: Use WUIO1 as input 2: Use WUIO2 as input 3: Use WUIO3 as input
19-16	MATCH_MODE	Match Mode Selection. Generate Match Pulse when : 0 : Disable Match Mode; 1 : Reserved; 2 : COUNTER >= COMPARE; 3 : COUNTER <= COMPARE; 4 : HIGH_DUMP >= COMPARE; 5 : HIGH_DUMP <= COMPARE; 6 : LOW_DUMP >= COMPARE; 7 : LOW_DUMP <= COMPARE; 8 : (HIGH_DUMP + LOW_DUMP) >= COMPARE; 9 : (HIGH_DUMP + LOW_DUMP) <= COMPARE; 10 : HIGH_DUMP - LOW_DUMP >= COMPARE; 11 : HIGH_DUMP - LOW_DUMP <= COMPARE; 12 : HIGH_DUMP - LOW_DUMP >= COMPARE and HIGH_DUMP >= LOW_DUMP; 13 : HIGH_DUMP - LOW_DUMP <= COMPARE and LOW_DUMP >= HIGH_DUMP; 14 : HIGH_DUMP - LOW_DUMP >= COMPARE and HIGH_DUMP <= LOW_DUMP; 15 : HIGH_DUMP - LOW_DUMP <= COMPARE and LOW_DUMP <= HIGH_DUMP;
9-8	DUMP_MODE	Dump Mode, select the time using to update LOW_DUMP(input signal's low level time) and HIGH_DUMP(input signal's high level time): 00: Disable Dump 01: Update at rising edge. (input selection via SELECT bit field) 10: Update at falling edge. (input selected via SELECT bit field) 11: Update when COUNTER counts period done.

Bits	Name	Description
3-0	COUNT_MODE	Count Mode - when Bit[0]: 0: Count up 1: Count down - when Bit[1]: 0: Oneshot mode 1: Period Mode - when Bit[3:2]: 00: Count by internal 32KHz clock 01: Count updates when rising edge 10: Count updates when falling edge

PCAP[CTRL] Bits

11.7.17 PCAP[CP_CFG] (0x204 + 0x18 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMPARE																PERIOD															
RW																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCAP[CP_CFG] [31:0]

Bits	Name	Description
31-16	COMPARE	Compare Value Note: This is a SYNC register, and it won't be valid immediately after change. It can be change only after last configuration valid.
15-0	PERIOD	Period Value Note: This is a SYNC register, and it won't be valid immediately after change. It can be change only after last configuration valid.

PCAP[CP_CFG] Bits

11.7.18 PCAP[COUNTER] (0x210 + 0x18 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																COUNTER															
N/A																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

PCAP[COUNTER] [31:0]

Bits	Name	Description
15-0	COUNTER	Counter Value

PCAP[COUNTER] Bits

11.7.19 PCAP[DUMP] (0x214 + 0x18 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LOW_DUMP																HIGH_DUMP															
RW																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCAP[DUMP] [31:0]

Bits	Name	Description
31-16	LOW_DUMP	Dump Value: Low Level Time(Base on 32K Clock)
15-0	HIGH_DUMP	Dump Value: High Level Time(Base on 32K Clock)

PCAP[DUMP] Bits

11.7.20 PCNT[CNT_CFG] (0x280 + 0x30 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EN	RSVD																		OVRD	RSVD			SEQ	RSVD	BIAS			DECODE_MODE			
RW	N/A																		RW	N/A			RW	N/A	RW			RW			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

PCNT[CNT_CFG] [31:0]

Bits	Name	Description
31	EN	Enable Pulse Counter 0: Disable 1: Enable Note: Other registers should be configured before enable this bit.

Bits	Name	Description
12	OVRD	Software Override Direction 0: Disable 1: Enable
8	SEQ	Change the direction of decoder 0: Disable 1: Enable If enable: In PD Mode: modify the positive direction by using high level in D channel as forward . In UD Mode: modify the positive direction by using pulse in U channel as forward, pulse in D channel as reverse. In ABZ Mode: modify the positive direction by using B channel's phase lead over A channel as forward. In UVW Mode: modify the positive direction by using W channel lead as forward.
6-4	BIAS	Bias compensation, compensation for decoded phase
3-0	DECODE_MODE	Decode Mode: 0: MODE_AB_4X 1: MODE_AB_2X 2: MODE_AB_1X 3: Reserved 4: MODE_PD_1X 5: MODE_PDF1X (Disable checking of direction change when input pulse is valid) 6: MODE_PD_2X 7: MODE_PDF2X (Disable checking of direction change when input pulse is valid) 8: MODE_UD_1X 9: MODE_UDF1X (Disable checking if both channel U and D are valid at the same time) 10: MODE_UD_2X 11: MODE_UDF2X (Disable checking if both channel U and D are valid at the same time) 12: MODE_UVW6X (120°) 13: MODE_UVW6Y (60°) 14: Reserved 15: Reserved

PCNT[CNT_CFG] Bits

11.7.21 PCNT[CALIB_CFG] (0x284 + 0x30 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD										ZMODE		ZSTATE				RSVD		H_SEL		RSVD		Z_SEL		RSVD		B_SEL		RSVD		A_SEL	
N/A										RW		RW				N/A		RW		N/A		RW		N/A		RW		N/A		RW	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

PCNT[CALIB_CFG] [31:0]

Bits	Name	Description
21-20	ZMODE	Phase Z Mode Select 0: Disable 1: Edge Mode(Use Edge in ZSTATE Register) 2: Level Mode(Match Phase A,B and Z) 3: Level Mode(Only Match Phase Z)
19-16	ZSTATE	Phase Z Matching Status -When in level mode: A bit2 of 1 indicates at the rising edge of the Z phase, and 0 at the falling edge. The remaining bits correspond to the level state of the HZBA phase. -When edge mode: If bit3 is 1: Match Z-phase rising edge, reverse If bit2 is 1: Match Z-phase falling edge, forward If bit1 is 1: Match Z-phase falling edge, reverse If bit0 is 1: Match Z-phase rising edge, forward.
13-12	H_SEL	Select Input for Phase H: 0: WUIO0 1: WUIO1 2: WUIO2 3: WUIO3
9-8	Z_SEL	Select Input for Phase Z: 0: WUIO0 1: WUIO1 2: WUIO2 3: WUIO3
5-4	B_SEL	Select Input for Phase B: 0: WUIO0 1: WUIO1 2: WUIO2 3: WUIO3

Bits	Name	Description
1-0	A_SEL	Select Input for Phase A: 0: WUIO0 1: WUIO1 2: WUIO2 3: WUIO3

PCNT[CALIB_CFG] Bits

11.7.22 PCNT[RESOLUTION] (0x288 + 0x30 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																RESOLUTION															
																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCNT[RESOLUTION] [31:0]

Bits	Name	Description
31-0	RESOLUTION	Resolution Value Note: The Field in this register is the same as COUNTER register, so if target resolution value is TARGET_LINES , it should be shifted according to the current DECODE_MODE, and then decrease 1 because it counts from 0, it means when: MODE_AB_4X: RESOLUTION = (TARGET_LINES « 2) - 1; MODE_AB_2X: RESOLUTION = (TARGET_LINES « 1) - 1; MODE_AB_1X: RESOLUTION = (TARGET_LINES « 0) - 1; MODE_PD_1X: RESOLUTION = (TARGET_LINES « 0) - 1; MODE_PDF1X: RESOLUTION = (TARGET_LINES « 0) - 1; MODE_PD_2X: RESOLUTION = (TARGET_LINES « 1) - 1; MODE_PDF2X: RESOLUTION = (TARGET_LINES « 1) - 1; MODE_UD_1X: RESOLUTION = (TARGET_LINES « 0) - 1; MODE_UDF1X: RESOLUTION = (TARGET_LINES « 0) - 1; MODE_UD_2X: RESOLUTION = (TARGET_LINES « 1) - 1; MODE_UDF2X: RESOLUTION = (TARGET_LINES « 1) - 1; MODE_UVW6X: RESOLUTION = (TARGET_LINES « 3) - 1; MODE_UVW6Y: RESOLUTION = (TARGET_LINES « 3) - 1;

PCNT[RESOLUTION] Bits

11.7.23 PCNT[HOMING_CFG] (0x28C + 0x30 * n)

MODULE PDGO

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD										RNG_CMP_SRC		RSVD		CMP0_MODE		RSVD		DMODE		DSTATE				RSVD		HMODE		HSTATE			
N/A										RW		N/A		RW		N/A		RW		RW				N/A		RW		RW			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCNT[HOMING_CFG] [31:0]

Bits	Name	Description
21-20	RNG_CMP_SRC	Select Compare Source for Comparator 0 in Range Match mode (Mode1): 0: Compare value from counter (Line+Phase), compare when counter updates 1: Compare value from counter (Revolution+Line+Phase), compare when counter updates 2: Compare value from tacho_line(16bits), compare after each tacho 3: Compare value from tacho_time(16bits), compare after each tacho

Bits	Name	Description
18-16	CMP0_MODE	Match Mode for Comparator 0: Comparison point 0 matches mode: Mode0 : Disable. Mode1 : Range Match: If $CMP0 \leq CMP1$, match valid when $CMP0 \leq RNG\_CMP\_SRC \leq CMP1$; If $CMP0 > CMP1$, match valid when $CMP0 \leq RNG\_CMP\_SRC$ or $RNG\_CMP\_SRC \leq CMP1$; Note: The interrupt will be only available when entry and exit compare, i.e., it is generated only twice, but you can check whether it matches or not through the status bit in register. Mode2 : when the COUNTER value matches the CMP0; Mode3 : when the COUNTER value matches the CMP0; and the direction update value should be the same as the current direction; Mode4: when counting up, the COUNTER value matches the CMP0; Mode5: when counting up, the COUNTER value matches the CMP0; and the direction update value should be the same as the current direction; Mode6: when counting down, the COUNTER value matches the CMP0; Mode7: when counting down, the COUNTER value matches the CMP0; and the direction update value should be the same as the current direction; Direction update value: direction decoded by the decoder according to the Phase (e.g. Phase 0 -> 1 -> 2 -> 3 -> 4 -> 5 -> 0 mean forward. Current Direction: direction decoded by the comparator based on the line field in counter.
13-12	DMODE	Dump Mode Selection 0: Disable 1: Edge Mode(Use Edge in DSTATE Register) 2: Level Mode(Match Phase A,B and H) 3: Level Mode(Only Match Phase H)

Bits	Name	Description
11-8	DSTATE	Dump Match State When level mode: The remaining bits correspond to the level state of the HZBA phase, respectively. When edge mode: Bit3: H-phase rising edge, reverse Bit2: H-phase falling edge, forward Bit1: H-phase falling edge, reverse Bit0: H-phase rising edge, forward.
5-4	HMODE	Phase H Mode Selection 0: Disable 1: Edge Mode(Use Edge in HSTATE Register) 2: Level Mode(Match Phase A,B and H) 3: Level Mode(Only Match Phase H)
3-0	HSTATE	H-Phase Match State When level mode: Each bit correspond to the level state of the HZBA phase, respectively. When edge mode: If bit3 is 1: Match H-phase rising edge, reverse If bit2 is 1: Match H-phase falling edge, forward If bit1 is 1: Match H-phase falling edge, reverse If bit0 is 1: Match H-phase rising edge, forward.

PCNT[HOMING_CFG] Bits

11.7.24 PCNT[ANALYSE_CFG] (0x290 + 0x30 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD													ERS	TACHO_MODE		TACHO_LEN															
N/A													RW	RW		RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

PCNT[ANALYSE_CFG] [31:0]

Bits	Name	Description
18	ERS	Whether to discard this result and automatically restart the tachometer immediately after a Dec_Error or Seq_Error is detected.

Bits	Name	Description
17-16	TACHO_MODE	Tacho Mode Selection 0: Disable tacho 1: Period Mode, specify the length of time, measure the number of complete lines traveled within the length of time, and record the actual time spent. 2: Travel Mode, specify the number of lines, measure the time taken to reach the number of lines.
15-0	TACHO_LEN	Tacho Length Period Mode: specify the length of time for each tacho. Travel Mode: specify the number of lines for each tacho. Note: This is a SYNC register, and it won't be valid immediately after change. It can be change only after last configuration valid.

PCNT[ANALYSE_CFG] Bits

11.7.25 PCNT[DIRECTION] (0x294 + 0x30 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																															DIR
N/A																															RW
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCNT[DIRECTION] [31:0]

Bits	Name	Description
0	DIR	Software Override Direction Value 0: Forward 1: Reverse

PCNT[DIRECTION] Bits

11.7.26 PCNT[COMP0] (0x298 + 0x30 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP0																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCNT[COMP0] [31:0]

Bits	Name	Description
31-0	CMP0	Comnpare Point 0 Note: This is a SYNC register, and it won't be valid immediately after change. It can be change only after last configuration valid.

PCNT[CMP0] Bits

11.7.27 PCNT[CMP1] (0x29C + 0x30 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP1																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCNT[CMP1] [31:0]

Bits	Name	Description
31-0	CMP1	Comnpare Point 1 Note: This is a SYNC register, and it won't be valid immediately after change. It can be change only after last configuration valid.

PCNT[CMP1] Bits

11.7.28 PCNT[CNT] (0x2A0 + 0x30 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNTER																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCNT[CNT] [31:0]

Bits	Name	Description
31-0	COUNTER	Pulse Counter Value Note: - Pulse Counter Value update when: - Software Update: According to the current DECODE_MODE, the software directly writes the target value to the Revolution field and Line field of this register; - Phase Z (Index) match: When Phase Z matches, clear Line field; Revolution field will be increased by 1 if forward, Revolution field remains unchanged if reverse, and the Phase field will be updated by decoder. - Phase H (Home) match: when Phase H matches, clear Revolution field; and the Line field and Phase field will be updated by decoder. - Decoder update: Revolution field, Line field and Phase field will be updated by decoder. - This is a SYNC register, and it won't be valid immediately after software write. It can be change only after last configuration valid. - Pulse Counter Value include: - Revolution field; - Line field, the width of this bit field depends on RESOLUTION register; - Phase field, the width of this bit field depends on DECODE_MODE register, when: - MODE_AB_4X: 2 Bits, Bit[1:0]; - MODE_AB_2X: 1 Bit, Bit[0]; - MODE_AB_1X: 0 Bit, no Phase field; - MODE_PD_1X: 0 Bit, no Phase field; - MODE_PDF1X: 0 Bit, no Phase field; - MODE_PD_2X: 1 Bit, Bit[0]; - MODE_PDF2X: 1 Bit, Bit[0]; - MODE_UD_1X: 0 Bit, no Phase field; - MODE_UDF1X: 0 Bit, no Phase field; - MODE_UD_2X: 1 Bit, Bit[0]; - MODE_UDF2X: 1 Bit, Bit[0]; - MODE_UVW6X: 3 Bits, Bit[2:0]; - MODE_UVW6Y: 3 Bits, Bit[2:0];

PCNT[CNT] Bits

11.7.29 PCNT[TACHO] (0x2A4 + 0x30 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LINE																TIME															
RW																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCNT[TACHO] [31:0]

Bits	Name	Description
31-16	LINE	Line Result Period Mode: count up from 0 when forward; count down from 0 when forward. Travel Mode: count up from 0 when forward; count down from 0xFFFF when forward.
15-0	TIME	Time Result Period Mode: The actual time taken to complete the complete Lines in a specified time. Travel Mode: The actual time taken to complete the specified Lines.

PCNT[TACHO] Bits

11.7.30 PCNT[DUMP] (0x2A8 + 0x30 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DUMP																RO															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCNT[DUMP] [31:0]

Bits	Name	Description
31-0	DUMP	Dump Result: Save COUNTER value when DUMP match.

PCNT[DUMP] Bits

11.7.31 PCNT[STS] (0x2AC + 0x30 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																MATCH0															
N/A																RO															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCNT[STS] [31:0]

Bits	Name	Description
0	MATCH0	MATCH0 Status 0:invalid 1:valid

PCNT[STS] Bits

12 Power Management Domain Configuration Module PCFG

This chapter describes the main features of the Power Domain Manager PCFG.

12.1 Feature Summary

PCFG is an integrated control module for power, clock and analog peripherals within the power management domain. Users can implement it through PCFG:

- Linear Regulator LDOOTP Control
- Switching Power Supply DCDC Configuration
- Power Management Domain Gated Clock Management
- Debugging Control

12.2 Function Description

This section describes the functions of PCFG.

12.2.1 Linear Regulator LDOOTP Configuration

linear regulator LDOOTP is the power supply for the on-chip one-time programmable memory OTP. Before burning OTP, you must first enable the ENABLE bit in the LDO 2P5 register and wait for the READY bit to be valid. The ENABLE bit must be cleared to turn off the LDO after OTP programming ends.

12.2.2 Switching Power Supply DCDC Configuration

Switching power supply DCDC can be used to provide power to the system power domain. The output voltage of DCDC is set through the VOLT bit in the DCDC_MODE register, and the default voltage is 1.15V.

DCDC is turned on by default. When the chip uses an external power supply to directly supply power to VDD_SOC, the MODE bit of the DCDC_MODE register should be set to 0 to turn off the DCDC.

When the system enters sleep mode (see [Section 15.4](#) for details), the DCDC can also be turned off or enter a low power consumption state. The relevant control is TRAP and RETENTION in the POWER_TRAP register. The combination relationship is as shown in [Table 23](#) :

TRAP bits	RETENTION	Settings
0	-	when the system is sleeping, the DCDC's working state remains unchanged
1	0	When the system is sleeping, DCDC turns off
1	1	When the system is sleeping, DCDC enters low power mode

Table 23: DCDC low power consumption state control

When DCDC enters low power mode, the output voltage is set in STBY_VOLT in the DCDC_LPMODE register. You can set a lower voltage depending on the application, such as using the memory contents of the system's

power domain to maintain the voltage value.

DCDC supports obtaining the current current intensity. After the current measurement function is turned on, it can be read from the LEVEL of the DCDC_CURRENT register.

12.2.3 Clock gating for Power Management Domain Module

Users can turn on or off the clocks of some modules in the power management domain through the SCG_CTRL register.

- OTP
- APB SRAM
- VAD
- PGPIO
- PIOC
- PTMR
- PWDG
- PUART

12.2.4 System power domain wake-up

The wakeup sources in the power management domain are:

- OTP interrupt
- VAD interrupt
- PTMR interrupt
- PUART interrupt
- PWDG interrupt
- PGPIO interrupt
- Power Management Domain Security Violations

allows the above wake-up sources to wake up the system through the WAKE_MASK register of the PCFG module, and the wake-up source status can be read and cleared in the WAKE_CAUSE register.

12.2.5 Debugging Control

Users can set it in processor debugging mode through the DEBUG_STOP register, some timing modules stop counting:

- HART0 = 1'b1, CPU0 Timing module stops counting during debugging
- HART1 = 1'b1, the timing module stops counting while CPU1 is debugging

The controlled timing modules are:

- GPTMR0 ~ 7
- WDG0 ~ 3
- PWM0 ~ 3
- PTMR
- PWDG

12.2.6 Power Management Domain Reset Control Module PPOR

PPOR manages the reset of the system's power domain.

Users can turn these resets on or off through the RESET_ENABLE register.

Users can check the source of the most recent reset through the RESET_FLAG register.

RESET_COLDregister controls whether a cold or warm reset will be triggered for each reset source signal

Users can reset the countdown counter through the COUNTER bit field configuration software of SOFTWARE_RESET. The user can configure COUNTER arbitrarily. When this counter counts down to 2, a software reset is generated. Writing 0 to COUNTER turns off software resets and cancels the countdown to software resets in progress.

12.3 PCFG register description

PCFG is as follows:

PCFG base address: 0xF4104000

Offset	Register Name	Description	Reset Value
0x0000	BANDGAP	BANGGAP control	0x00101010
0x0004	LDO1P1	1V LDO config	0x0001044C
0x0008	LDO2P5	2.5V LDO config	0x000009C4
0x0010	DCDC_MODE	DCDC mode select	0x0001047E
0x0014	DCDC_LPMODE	DCDC low power mode	0x00000384
0x0018	DCDC_PROT	DCDC protection	0x00000010
0x001C	DCDC_CURRENT	DCDC current estimation	0x00000000
0x0020	DCDC_ADVMODE	DCDC advance setting	0x03120040
0x0024	DCDC_ADVPARAM	DCDC advance parameter	0x000006E1C
0x0028	DCDC_MISC	DCDC misc parameter	0x00070100
0x002C	DCDC_DEBUG	DCDC Debug	0x000005DBF
0x0030	DCDC_START_TIME	DCDC ramp time	0x0001193F
0x0034	DCDC_RESUME_TIME	DCDC resume time	0x000008C9F
0x0040	POWER_TRAP	SOC power trap	0x00000000
0x0044	WAKE_CAUSE	Wake up source	0x00000000
0x0048	WAKE_MASK	Wake up mask	0x00000000
0x004C	SCG_CTRL	Clock gate control in PMIC	0xFFFFFFFF
0x0060	RC24M	RC 24M config	0x00000310
0x0064	RC24M_TRACK	RC 24M track mode	0x00000000
0x0068	TRACK_TARGET	RC 24M track target	0x00000000
0x006C	STATUS	RC 24M track status	0x00000000

Offset	Register Name	Description	Reset Value

Table 24: PCFG Register List

12.4 PCFG register details

PCFG are as follows:

12.4.1 BANDGAP (0x0)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VBG_TRIMMED	RSVD										VBG_1P0_TRIM				RSVD			VBG_P65_TRIM				RSVD			VBG_P50_TRIM						
RW	N/A										RW				N/A			RW				N/A			RW						
0	x	x	x	x	x	x	x	x	x	x	1	0	0	0	0	x	x	x	1	0	0	0	0	x	x	x	1	0	0	0	0

BANDGAP [31:0]

Bits	Name	Description
31	VBG_TRIMMED	Bandgap trim happened, this bit set by hardware after trim value loaded, and stop load, write 0 will clear this bit and reload trim value 0: bandgap is not trimmed 1: bandgap is trimmed
20-16	VBG_1P0_TRIM	Banggap 1.0V output trim value
12-8	VBG_P65_TRIM	Banggap 1.0V output trim value
4-0	VBG_P50_TRIM	Banggap 1.0V output trim value

BANDGAP Bits

12.4.2 LDO1P1 (0x4)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD															ENABLE	RSVD					VOLT										
N/A															RW	N/A					RW										
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	x	x	x	x	0	1	0	0	0	1	0	0	1	1	0	0

LDO1P1 [31:0]

CONFIGURATION MODULE PCFG

Bits	Name	Description
16	ENABLE	LDO enable 0: turn off LDO 1: turn on LDO
11-0	VOLT	LDO output voltage in mV, value valid through 700-1320, , step 20mV. Hardware select voltage no less than target if not on valid steps, with maximum 1320mV. 700: 700mV 720: 720mV ... 1320:1320mV

LDO1P1 Bits

12.4.3 LDO2P5 (0x8)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD			READY	RSVD											ENABLE	RSVD			VOLT												
N/A			RO	N/A											RW	N/A			RW												
x	x	x	0	x	x	x	x	x	x	x	x	x	x	x	0	x	x	x	x	1	0	0	1	1	1	0	0	0	1	0	0

LDO2P5 [31:0]

Bits	Name	Description
28	READY	Ready flag, will set 1ms after enabled or voltage change 0: LDO is not ready for use 1: LDO is ready
16	ENABLE	LDO enable 0: turn off LDO 1: turn on LDO
11-0	VOLT	LDO output voltage in mV, value valid through 2125-2900, step 25mV. Hardware select voltage no less than target if not on valid steps, with maximum 2900mV. 2125: 2125mV 2150: 2150mV ... 2900:2900mV

LDO2P5 Bits

12.4.4 DCDC_MODE (0x10)

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

12 POWER MANAGEMENT DOMAIN

CONFIGURATION MODULE PCFG

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD			READY	RSVD									MODE			RSVD			VOLT												
N/A			RO	N/A									RW			N/A			RW												
x	x	x	0	x	x	x	x	x	x	x	x	x	0	0	1	x	x	x	x	0	1	0	0	0	1	1	1	1	1	1	0

DCDC_MODE [31:0]

Bits	Name	Description
28	READY	Ready flag 0: DCDC is applying new change 1: DCDC is ready
18-16	MODE	DCDC work mode XX0: trun off 001: basic mode 011: generic mode 101: automatic mode 111: expert mode
11-0	VOLT	DCDC voltage in mV in normal mode, value valid through 600-1375, , step 25mV. Hardware select voltage no less than target if not on valid steps, with maximum 1375mV. 600: 600mV 625: 625mV ... 1375:1375mV

DCDC_MODE Bits

12.4.5 DCDC_LPMODE (0x14)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
RSVD																				STBY_VOLT																
N/A																				RW																
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	1	1	1	0	0	0	0	1	0	0					

DCDC_LPMODE [31:0]

CONFIGURATION MODULE PCFG

Bits	Name	Description
11-0	STBY_VOLT	DCDC voltage in mV in standby mode, , value valid through 600-1375, , step 25mV. Hardware select voltage no less than target if not on valid steps, with maximum 1375mV. 600: 600mV 625: 625mV ... 1375:1375mV

DCDC_LPMODE Bits

12.4.6 DCDC_PROT (0x18)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD			ILIMIT_LP	RSVD			OVERLOAD_LP	RSVD							POWER_LOSS_FLAG	DISABLE_OVERVOLTAGE	RSVD					OVERVOLT_FLAG	DISABLE_SHORT	RSVD		SHORT_CURRENT	RSVD			SHORT_FLAG	
N/A			RW	N/A			RO	N/A							RO	RW	N/A					RO	RW	N/A		RW	N/A			RO	
x	x	x	0	x	x	x	0	x	x	x	x	x	x	x	0	0	x	x	x	x	x	x	0	0	x	x	1	x	x	x	0

DCDC_PROT [31:0]

Bits	Name	Description
28	ILIMIT_LP	over current setting for low power mode 0:250mA 1:200mA
24	OVERLOAD_LP	over current in low power mode 0: current is below setting 1: overcurrent happened in low power mode
16	POWER_LOSS_FLAG	power loss 0: input power is good 1: input power is too low
15	DISABLE_OVERVOLTAGE	ouput over voltage protection 0: protection enabled, DCDC will shut down is output voltage is unexpected high 1: protection disabled, DCDC continue to adjust output voltage
8	OVERVOLT_FLAG	output over voltage flag 0: output is normal 1: output is unexpected high

Bits	Name	Description
7	DISABLE_SHORT	disable output short circuit protection 0: short circuits protection enabled, DCDC shut down if short circuit on output detected 1: short circuit protection disabled
4	SHORT_CURRENT	short circuit current setting 0: 2.0A, 1: 1.3A
0	SHORT_FLAG	short circuit flag 0: current is within limit 1: short circuits detected

DCDC_PROT Bits

12.4.7 DCDC_CURRENT (0x1C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																ESTI_EN	RSVD						VALID	RSVD			LEVEL				
N/A																RW	N/A						RO	N/A			RO				
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	x	x	x	x	x	x	0	x	x	x	0	0	0	0	0

DCDC_CURRENT [31:0]

Bits	Name	Description
15	ESTI_EN	enable current measure
8	VALID	Current level valid 0: data is invalid 1: data is valid
4-0	LEVEL	DCDC current level, current level is num * 50mA

DCDC_CURRENT Bits

12.4.8 DCDC_ADVMODE (0x20)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD					EN_RCSCALE			RSVD		DC_C		DC_R			RSVD										EN_FF_DET	EN_FF_LOOP	EN_AUTOLP	EN_DCM_EXIT	EN_SKIP	EN_IDLE	EN_DCM
N/A					RW			N/A		RW		RW			N/A										RW	RW	RW	RW	RW	RW	RW
x	x	x	x	x	0	1	1	x	x	0	1	0	0	1	0	x	x	x	x	x	x	x	x	x	1	0	0	0	0	0	0

DCDC_ADVMODE [31:0]

CONFIGURATION MODULE PCFG

Bits	Name	Description
26-24	EN_RCSCALE	Enable RC scale
21-20	DC_C	Loop C number
19-16	DC_R	Loop R number
6	EN_FF_DET	enable feed forward detect 0: feed forward detect is disabled 1: feed forward detect is enabled
5	EN_FF_LOOP	enable feed forward loop 0: feed forward loop is disabled 1: feed forward loop is enabled
4	EN_AUTOLP	enable auto enter low power mode 0: do not enter low power mode 1: enter low power mode if current is detected low
3	EN_DCM_EXIT	avoid over voltage 0: stay in DCM mode when voltage excess 1: change to CCM mode when voltage excess
2	EN_SKIP	enable skip on narrow pulse 0: do not skip narrow pulse 1: skip narrow pulse
1	EN_IDLE	enable skip when voltage is higher than threshold 0: do not skip 1: skip if voltage is excess
0	EN_DCM	DCM mode 0: CCM mode 1: DCM mode

DCDC_ADVMODE Bits

12.4.9 DCDC_ADVPARAM (0x24)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																MIN_DUT						RSVD	MAX_DUT								
N/A																RW						N/A	RW								
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	1	0	1	1	1	0	x	0	0	1	1	1	0	0

DCDC_ADVPARAM [31:0]

Bits	Name	Description
14-8	MIN_DUT	minimum duty cycle
6-0	MAX_DUT	maximum duty cycle

DCDC_ADVPARAM Bits

12.4.10 DCDC_MISC (0x28)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD			EN_HYST	RSVD		HYST_SIGN	HYST_THRS	RSVD			RC_SCALE	RSVD	DC_FF			RSVD						OL_THRE	RSVD			OL_HYST	RSVD	DELAY	CLK_SEL	EN_STEP	
N/A			RW	N/A		RW	RW	N/A			RW	N/A	RW			N/A						RW	N/A			RW	N/A	RW	RW	RW	RW
x	x	x	0	x	x	0	0	x	x	x	0	x	1	1	1	x	x	x	x	x	x	0	1	x	x	x	0	x	0	0	0

DCDC_MISC [31:0]

Bits	Name	Description
28	EN_HYST	hysteresis enable
25	HYST_SIGN	hysteresis sign
24	HYST_THRS	hysteresis threshold
20	RC_SCALE	Loop RC scale threshold
18-16	DC_FF	Loop feed forward number
9-8	OL_THRE	overload threshold in low power mode
4	OL_HYST	voltage ripple threshold in low power mode 0: 12.5mV 1: 25mV
2	DELAY	enable delay 0: delay disabled, 1: delay enabled
1	CLK_SEL	clock selection 0: select DCDC internal oscillator 1: select RC24M oscillator
0	EN_STEP	enable stepping in voltage change 0: stepping disabled, 1: stepping enabled

DCDC_MISC Bits

12.4.11 DCDC_DEBUG (0x2C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																						UPDATE_TIME									
N/A																						RW									
x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	1	0	1	1	1	0	1	1	0	1	1	1	1	1	1

DCDC_DEBUG [31:0]

Bits	Name	Description
19-0	UPDATE_TIME	DCDC voltage change time in 24M clock cycles, default value is 1mS

DCDC_DEBUG Bits

12.4.12 DCDC_START_TIME (0x30)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD												START_TIME																			
N/A												RW																			
x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	1	0	0	0	1	1	0	0	1	0	0	1	1	1	1	1	1

DCDC_START_TIME [31:0]

Bits	Name	Description
19-0	START_TIME	Start delay for DCDC to turn on, in 24M clock cycles, default value is 3mS

DCDC_START_TIME Bits

12.4.13 DCDC_RESUME_TIME (0x34)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD												RESUME_TIME																			
N/A												RW																			
x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	1	0	0	0	1	1	0	0	1	0	0	1	1	1	1	1

DCDC_RESUME_TIME [31:0]

Bits	Name	Description
19-0	RESUME_TIME	Resume delay for DCDC to recover from low power mode, in 24M clock cycles, default value is 1.5mS

DCDC_RESUME_TIME Bits

12.4.14 POWER_TRAP (0x40)

CONFIGURATION MODULE PCFG

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRIGGERED	RSVD														RETENTION	RSVD														TRAP	
RW	N/A														RW	N/A														RW	
0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0

POWER_TRAP [31:0]

Bits	Name	Description
31	TRIGGERED	Low power trap status, this bit will set when power related low power flow triggered, write 1 to clear this flag. 0: low power trap is not triggered 1: low power trap triggered
16	RETENTION	DCDC enter standby mode, which will reduce voltage for memory content retention 0: Shutdown DCDC 1: reduce DCDC voltage
0	TRAP	Enable trap of SOC power supply, trap is used to hold SOC in low power mode for DCDC to enter further low power mode, this bit will self-clear when power related low power flow triggered 0: trap not enabled, pmic side low power function disabled 1: trap enabled, STOP operation leads to PMIC low power flow if SOC is not retentioned.

POWER_TRAP Bits

12.4.15 WAKE_CAUSE (0x44)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAUSE																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

WAKE_CAUSE [31:0]

Bits	Name	Description
31-0	CAUSE	wake up cause, each bit represents one wake up source, write 1 to clear the register bit 0: wake up source is not active during last wakeup 1: wake up source is active during last wakeup bit 0: pmic_enable bit 7: UART interrupt bit 8: TMR interrupt bit 9: WDG interrupt bit10: GPIO in PMIC interrupt bit16: batt security interrupt bit17:batt gpio interrupt bit19:rtc interrupt bit31: pin wakeup

WAKE_CAUSE Bits

12.4.16 WAKE_MASK (0x48)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MASK																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

WAKE_MASK [31:0]

Bits	Name	Description
31-0	MASK	mask for wake up sources, each bit represents one wakeup source 0: allow source to wake up system 1: disallow source to wakeup system bit 0: pmic_enable bit 7: UART interrupt bit 8: TMR interrupt bit 9: WDG interrupt bit10: GPIO in PMIC interrupt bit16: batt security interrupt bit17:batt gpio interrupt bit19:rtc interrupt bit31: pin wakeup

WAKE_MASK Bits

12.4.17 SCG_CTRL (0x4C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCG																															
RW																															
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	

SCG_CTRL [31:0]

Bits	Name	Description
31-0	SCG	control whether clock being gated during PMIC low power flow, 2 bits for each peripheral 00,01: reserved 10: clock is always off 11: clock is always on bit6-7:gpio bit8-9:ioc bit10-11: timer bit12-13:wdog bit14-15:uart

SCG_CTRL Bits

12.4.18 RC24M (0x60)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RC_TRIMMED	RSVD																				TRIM_C			RSVD			TRIM_F					
RW	N/A																				RW			N/A			RW					
0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	1	1	x	x	x	1	0	0	0	0	

RC24M [31:0]

Bits	Name	Description
31	RC_TRIMMED	RC24M trim happened, this bit set by hardware after trim value loaded, and stop load, write 0 will clear this bit and reload trim value 0: RC is not trimmed 1: RC is trimmed
10-8	TRIM_C	Coarse trim for RC24M, bigger value means faster
4-0	TRIM_F	Fine trim for RC24M, bigger value means faster

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

12 POWER MANAGEMENT DOMAIN

CONFIGURATION MODULE PCFG

Bits	Name	Description

RC24M Bits

12.4.19 RC24M_TRACK (0x64)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
RSVD															SEL24M	RSVD															RETURN	RSVD		TRACK
N/A															RW	N/A															RW	N/A		RW
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	x	x	x	x	x	x	x	x	x	x	x	0	x	x	x	0			

RC24M_TRACK [31:0]

Bits	Name	Description
16	SEL24M	Select track reference 0: select 32K as reference 1: select 24M XTAL as reference
4	RETURN	Retrun default value when XTAL loss 0: remain last tracking value 1: switch to default value
0	TRACK	track mode 0: RC24M free running 1: track RC24M to external XTAL

RC24M_TRACK Bits

12.4.20 TRACK_TARGET (0x68)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRE_DIV																TARGET															
RW																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TRACK_TARGET [31:0]

Bits	Name	Description
31-16	PRE_DIV	Divider for reference source
15-0	TARGET	Target frequency multiplier of divided source

TRACK_TARGET Bits

12.4.21 STATUS (0x6C)



CONFIGURATION MODULE PCFG

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD											SEL32K	RSVD			SEL24M	EN_TRIM	RSVD				TRIM_C			RSVD			TRIM_F				
N/A											RO	N/A			RO	RO	N/A				RO			N/A			RO				
x	x	x	x	x	x	x	x	x	x	x	0	x	x	x	0	0	x	x	x	x	0	0	0	x	x	x	0	0	0	0	0

STATUS [31:0]

Bits	Name	Description
20	SEL32K	track is using XTAL32K 0: track is not using XTAL32K 1: track is using XTAL32K
16	SEL24M	track is using XTAL24M 0: track is not using XTAL24M 1: track is using XTAL24M
15	EN_TRIM	default value takes effect 0: default value is invalid 1: default value is valid
10-8	TRIM_C	default coarse trim value
4-0	TRIM_F	default fine trim value

STATUS Bits

12.5 PPOR register description

PPOR is as follows:

PPOR base address: 0xF4100000

Offset	Register Name	Description	Reset Value
0x0000	RESET_FLAG	flag indicate reset source	0x00000000
0x0004	RESET_STATUS	reset source status	0x00000000
0x0008	RESET_HOLD	reset hold attribute	0x00000000
0x000C	RESET_ENABLE	reset source enable	0xFFFFFFFF
0x0010	RESET_TYPE	reset type triggered by reset	0x00000000
0x001C	SOFTWARE_RESET	Software reset counter	0x00000000

Table 25: PPOR Register List

12.6 PPOR register details

PPOR are as follows:

12.6.1 RESET_FLAG (0x0)



CONFIGURATION MODULE PCFG

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLAG																															
W1C																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RESET_FLAG [31:0]

Bits	Name	Description
31-0	FLAG	reset reason of last hard reset, write 1 to clear each bit 0: brownout 1: temperature 4: debug reset 5: jtag soft reset 8: cpu0 lockup(not available) 9: cpu1 lockup(not available) 10: cpu0 request(not available) 11: cpu1 request(not available) 16: watch dog 0 17: watch dog 1 18: watch dog 2(not available) 19: watch dog 3(not available) 24: pmic watch dog 30: jtag ieee reset 31: software

RESET_FLAG Bits

12.6.2 RESET_STATUS (0x4)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STATUS																															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RESET_STATUS [31:0]

Bits	Name	Description
31-0	STATUS	current status of reset sources 0: brownout 1: temperature 4: debug reset 5: jtag soft reset 8: cpu0 lockup(not available) 9: cpu1 lockup(not available) 10: cpu0 request(not available) 11: cpu1 request(not available) 16: watch dog 0 17: watch dog 1 18: watch dog 2(not available) 19: watch dog 3(not available) 24: pmic watch dog 30: jtag ieee reset 31: software

RESET_STATUS Bits

12.6.3 RESET_HOLD (0x8)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HOLD																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RESET_HOLD [31:0]

Bits	Name	Description
31-0	HOLD	hold attribute, when set, SOC keep in reset status until reset source release, or, reset will be released after SOC enter reset status 0: brownout 1: temperature 4: debug reset 5: jtag soft reset 8: cpu0 lockup(not available) 9: cpu1 lockup(not available) 10: cpu0 request(not available) 11: cpu1 request(not available) 16: watch dog 0 17: watch dog 1 18: watch dog 2(not available) 19: watch dog 3(not available) 24: pmic watch dog 30: jtag ieee reset 31: software

RESET_HOLD Bits

12.6.4 RESET_ENABLE (0xC)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE																															
RW																															
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	

RESET_ENABLE [31:0]

Bits	Name	Description
31-0	ENABLE	enable of reset sources 0: brownout 1: temperature 4: debug reset 5: jtag soft reset 8: cpu0 lockup(not available) 9: cpu1 lockup(not available) 10: cpu0 request(not available) 11: cpu1 request(not available) 16: watch dog 0 17: watch dog 1 18: watch dog 2(not available) 19: watch dog 3(not available) 24: pmic watch dog 30: jtag ieee reset 31: software

RESET_ENABLE Bits

12.6.5 RESET_TYPE (0x10)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TYPE																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RESET_TYPE [31:0]

Bits	Name	Description
31-0	TYPE	reset type of reset sources, 0 for cold reset, all system control setting cleared except debug/fuse/ioc; 1 for hot reset, keep system control setting and debug/fuse/ioc setting, only clear some subsystem 0: brownout 1: temperature 4: debug reset 5: jtag soft reset 8: cpu0 lockup(not available) 9: cpu1 lockup(not available) 10: cpu0 request(not available) 11: cpu1 request(not available) 16: watch dog 0 17: watch dog 1 18: watch dog 2(not available) 19: watch dog 3(not available) 24: pmic watch dog 30: jtag ieee reset 31: software

RESET_TYPE Bits

12.6.6 SOFTWARE_RESET (0x1C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																	
																COUNTER																																																
																																RW																																
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																																	

SOFTWARE_RESET [31:0]

Bits	Name	Description
31-0	COUNTER	counter decrease in 24MHz and stop at 0, trigger reset when value reach 2, software can write 0 to cancel reset

SOFTWARE_RESET Bits

13 System Control Module (SYSCTL)

This chapter describes the main functions and usage of the SYSCTL module.

13.1 Feature Summary

The SYSCTL module is the management module of the system power domain. Its main functions are:

- Functional clock configuration
- Processor boot management
- System power domain resource management
- Low power management
- Clock measurement

13.2 Functional Clock Configuration

Functional clock configuration is implemented in the SYSCTL module. Users can configure the functional clock by modifying the CLOCK[x] register. The MUX bit is used to select the clock source, and the clock source list is shown in [Table 19](#). The DIV bit is used to set the frequency division coefficient. The valid configurable value of the register ranges from 0~255, and the actual frequency division coefficient is the register value plus 1. For example, if the DIV bit is set to 3, the selected clock source will be divided by 4 before output.

When the LOC_BUSY bit is 0, it indicates that the functional clock is available under the current configuration.

Both frequency division coefficient and clock source selection support on-the-fly modification.

13.3 CPU Boot Management

CPU0 is the only core, so the system always boots from CPU0.

13.3.1 Boot Management

The boot of CPU0 supports fast program jump after wake-up from low power mode, see [Section 18.5](#) for details. SYSCTL provides the following support in this process:

- The RESET/SLEEP/WAKE bits of the CPU[CPU0][LP] register are used to record event flags
- The CPU[CPU0][GPR0] register is used to store the fast program jump entry
- The CPU[CPU0][GPRx] registers are used to implement program verification for fast jump

13.4 System Power Domain Resource Management

The SYSCTL module manages clocks, power switches and resets of the entire system power domain.

This section describes the on/off control mechanism of various resources in the system power domain in detail. For a brief description of the programming model, you can directly jump to [Subsection 13.4.6](#).

13.4.1 Resource Nodes

In the system power domain, various nodes that can be turned on or off are called resource nodes, including each functional module, functional clock, subsystem power and reset, clock source, etc.

The RESOURCE[x] register is used to configure resource nodes, and the MODE bit is used to configure

the control mode of the resource node. Software can directly modify the MODE bit to control the enabling and disabling of a resource.

For functional modules, set the MODE bit of their RESOURCE[*] register to 1 to enable the module, and set it to 2 to disable the module.

For CPU subsystems, set the MODE bit of their RESOURCE[POW_*] register to 1 to turn on the power switch of the subsystem, and set it to 2 to turn off the power switch.

For CPU/SoC subsystems, set the MODE bit of their RESOURCE[RST_*] register to 2 to keep the subsystem in reset state, and set it to 1 to exit reset.

For functional clocks and clock sources, set the MODE bit of their RESOURCE[CLK_TOP_*] and RESOURCE[CLK_SRC_*] registers to 1 to enable the clock, and set it to 2 to disable the clock.

When the LOC_BUSY bit is 0, it indicates that the resource node is available.

In practical applications, the control mode of resource nodes is usually configured to the default value 0, which is automatic mode. See the next subsection for its specific working mechanism.

13.4.2 Chain Structure of Resource Nodes

There are dependencies between resource nodes. For example, the normal operation of a functional module depends on the correct output of the functional clock, and the functional clock in turn depends on the frequency stability of clock sources such as PLL. This dependency forms a chain structure: the downstream of the chain is each terminal functional module, the midstream is resources directly serving functional modules such as functional clocks, power supplies and resets, and the upstream is mainly various clock sources.

Figure 8 takes the ENET0 and SDXC0 modules in the SoC subsystem as examples to describe the chain relationship of resource nodes.

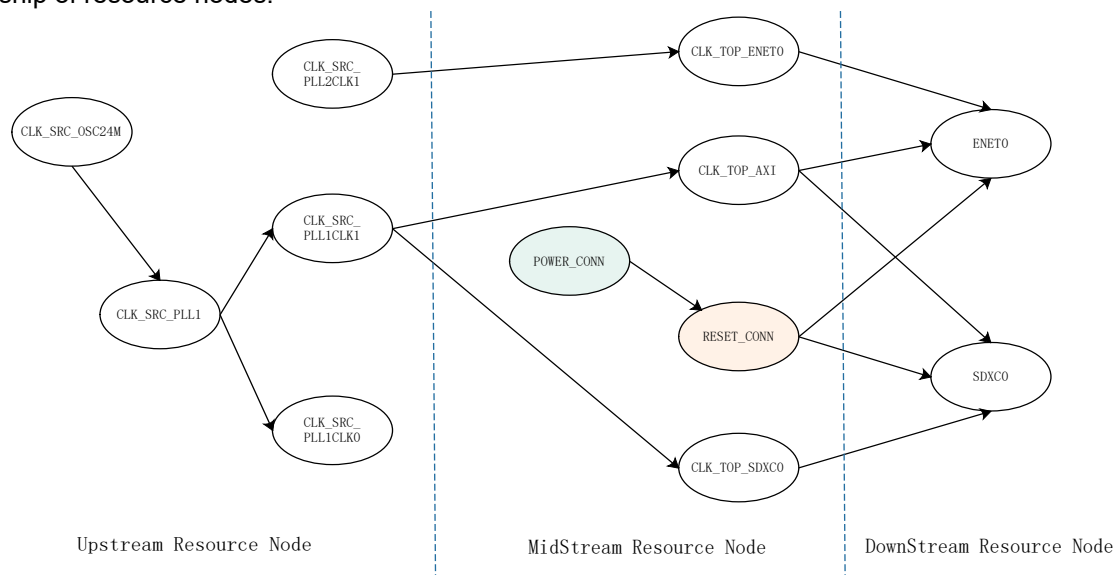


Figure 8: Chain Relationship of Resource Nodes

*For structural principle explanation only, it does not mean that the product necessarily includes ENET0/SDXC0/CONN subsystems

The normal operation of the downstream resource node ENET0 depends on the interface clock provided by CLK_TOP_ENET0, the bus clock provided by CLK_TOP_AXI, and requires that the CONN subsystem it belongs

to is in non-reset state and the power switch is turned on.

The midstream resource node CLK_TOP_ENET0 depends on the clock source PLL2CLK1 under default configuration, and CLK_TOP_AXI depends on the clock source PLL1CLK1 under default configuration. In addition, the reset exit of the CONN subsystem depends on the power switch of the CONN subsystem being turned on.

The upstream resource node PLL1CLK1 clock source depends on PLL1 being enabled and frequency stabilized, and PLL1 in turn depends on XTAL24M providing a valid reference clock.

All resource nodes are uniformly managed by the SYSCTL module, and the dependencies between resource nodes are automatically implemented in hardware, so users do not need to pay attention to specific dependencies. On the resource node chain, when a downstream node needs to be turned on, it will automatically request all midstream and upstream resource nodes it depends on to turn on. If the MODE bit in the RESOURCE[] register of relevant midstream and upstream resource nodes is 0 (automatic mode), the node will automatically turn on after receiving the request, and at the same time request more upstream resource nodes to turn on. Resource nodes work in automatic mode by default.

Take the ENET0 module as an example, when all midstream and upstream resource nodes work in the default automatic mode, modifying the MODE bit of the RESOURCE[ENET0] register to 1 means enabling the ENET0 module. The LOC_BUSY bit of the RESOURCE[ENET0] register will be set to 1, and all relevant resource nodes will be automatically enabled through chain conduction. Until XTAL24M oscillation is stable, PLL1 clock is locked, CLK_TOP_AXI and CLK_TOP_ENET0 clocks are enabled, and the CONN subsystem power is turned on and reset is completed, the LOC_BUSY bit of the RESOURCE[ENET0] register will be set to 0, indicating that the ENET0 module is ready for use.

13.4.3 Automatic Shutdown Mechanism of Resource Nodes

Resource nodes on the chain structure can be automatically turned on according to requests, and can also be automatically turned off under certain conditions.

When a resource node is shut down, it will send a shutdown request to the resource nodes it depends on. A resource node is allowed to shut down only when all downstream nodes connected to it have sent shutdown requests.

Assuming that [Figure 8](#) shows the complete dependency relationship of relevant resource nodes, when all midstream and upstream resource nodes work in the default automatic mode, if the MODE bit of the RESOURCE[ENET0] register is modified to 2, that is, to shut down the ENET0 module, the module clock of ENET0 will be shut down first, and then requests to shut down all resource nodes it depends on. Since CLK_TOP_ENET0 only serves ENET0 as its only downstream resource node, after receiving the shutdown request, it will perform the shutdown operation and continue to send a shutdown request to its dependent upstream resource node PLL2CLK1.

CLK_TOP_AXI will also receive the shutdown request from ENET0, but since SDXC0 connected to its downstream is still in working state, CLK_TOP_AXI is not allowed to shut down, and the same applies to the RESET_CONN resource node.

13.4.4 Resource Node Keep-ON

The automatic shutdown mechanism of resource nodes can automatically transmit shutdown requests to the most upstream of each chain, such as XTAL24M and PLL, when allowed. Re-enabling these resources will consume a certain amount of time. If users want the shutdown request to stop propagating when reaching a certain resource node, they only need to set the MODE bit of the node's RESOURCE[] register to 1 to force it to remain enabled.

Common scenarios include: keeping PLL from being shut down or XTAL24M running all the time, and maintaining the power supply of subsystems.

For subsystems with independent power switches, there are several possible working states in low power mode:

- Clock retention: Keep resource nodes of individual or all modules in the subsystem enabled, so that these modules can start working at any time. This requires setting the MODE bit of the corresponding RESOURCE[*] register to 1
- Register retention: Allow all module resource nodes in the subsystem to be shut down, but do not reset the subsystem, and retain power, so that the states of all registers and logic circuits in the subsystem are saved. This requires setting the MODE bit of the subsystem's RESOURCE[RST_*] and RESOURCE[POW_*] registers to 1
- Memory retention: Allow the subsystem to enter reset state, but retain power, so that data in the subsystem's memory can be retained. This requires setting the MODE bit of the subsystem's RESOURCE[POW_*] register to 1
- Subsystem power off

There is another way to implement subsystem state setting, see [Subsection 13.5.2](#) for details.

13.4.5 Connection between Functional Modules and Processors

When midstream and upstream resource nodes work in automatic mode, they will automatically turn on and off according to requests from their downstream nodes, while downstream functional module resource nodes working in automatic mode can turn on and off according to the processor state.

SYSCTL provides a link group (link group 0) for connecting functional modules with the processor, and group0 is for CPU0 use.

The connection between functional modules and groups is set in the GROUP[*] register. Each bit of the register corresponds to a functional module, and setting the bit to 1 means binding the module to the group. [Table 26](#), [Table 27](#) and [Table 28](#) list the correspondence between GROUP register bits and functional modules.

GROUPx[0] Register	
bit	Resource Node
0	AHB Peripheral Bus
1	AXI Peripheral Bus
2	CONN Subsystem Bus
3	ROM
4	ILM0/DLM0

GROUPx[0] Register	
bit	Resource Node
5	MCHTMR0
6	GPTMR0
7	GPTMR1
8	GPTMR2
9	GPTMR3
10	OWR0
11	OWR1
12	EUI0
13	EUI1
14	I2C0
15	I2C1
16	I2C2
17	I2C3
18	SPI0
19	SPI1
20	SPI2
21	SPI3
22	UART0
23	UART1
24	UART2
25	UART3
26	UART4
27	UART5
28	UART6
29	UART7
30	CRC0
31	TSNS

Table 26: Functional Module Connection Bit Table 0

GROUPx[1] Register	
bit	Resource Node
0	EWDG0
1	EWDG1
2	MBX0
3	GPIO
4	PPI
5	HDMA
6	LOBS
7	ADC0

GROUPx[1] Register	
bit	Resource Node
8	ADC1
9	ACMP0
10	MCAN0
11	MCAN1
12	MCAN2
13	MCAN3
14	PTPC
15	QEIO
16	QEIO1
17	QEO0
18	QEO1
19	SDM
20	PLB
21	PWM0
22	PWM1
23	EMDS
24	XPIO
25	XRAM
26	XDMA
27	ESC
28	ENET0
29	USB0
30	NTM0
31	REF0

Table 27: Functional Module Connection Bit Table 1

GROUPx[2] Register	
bit	Resource Node
0	REF1

Table 28: Functional Module Connection Bit Table 2

To ensure the basic system works properly, SYSCTL has made default configurations for some resource nodes: for example, CPU0 is bound to group0 by default, and resource nodes AHBAPB peripheral bus, AXI system bus and ROM are connected to group0 by default.

The connection between CPU and group is configured through the AFFILIATE[x] register, and the default value of the LINK bit of the AFFILIATE[CPU0] register is 0b1, which means CPU0 has been connected to group0.

When the CPU and functional module resource nodes are connected to the same group, and the resource nodes work in automatic mode, these resource nodes will automatically turn on or off as the CPU state changes. For example, when the CPU connects a resource node to the group it is bound to by modifying the GROUP

register, the functional module of this node will be automatically enabled, and a series of resources dependent on the module will be automatically enabled together. This is also the most convenient functional module enabling method recommended by the system.

Correspondingly, when the CPU enters sleep, all connected resource nodes will be automatically shut down. For detailed behavior management after CPU sleep, please refer to [Section 13.5](#).

After introducing the group and CPU connection function, [Figure 8](#) can be expanded to the form of [Figure 9](#), forming a complete chain structure.

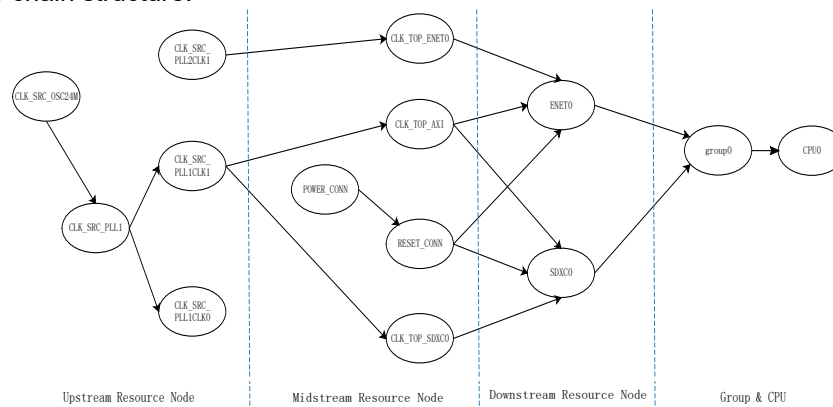


Figure 9: Chain Relationship between Resource Nodes and Group/CPU

*For structural principle explanation only, it does not mean that the product necessarily includes ENET0/SDXC0/CONN subsystems

In the figure, ENET0 and SDXC0 are connected to group0, indicating that CPU0 needs to use these two modules, which also forms the dependency of the resource node chain.

When CPU0 enters sleep, it will send shutdown requests to both ENET0 and SDXC0, and ENET0 and SDXC0 will shut down and transmit the shutdown request to the upstream of the chain.

13.4.6 Usage of Resource Nodes

All functional modules in the system power domain are resource nodes, which are listed in [Table 26](#).

Except for modules such as AHB/APB peripheral bus, AXI system bus and ROM which are enabled by default, most functional modules are in off state by default.

If the software running on CPU0 needs to use a functional module, it only needs to set the corresponding bit of the module in the GROUP0 register to 1, and then wait until the GLB_BUSY bit of the corresponding RESOURCE[*] register of the module becomes 0. All necessary resources will be automatically turned on.

13.5 Low Power Management

13.5.1 Processor Low Power Modes and Wake-up

Executing the WFI instruction is the sign that the processor switches from running mode to low power mode. The system behavior after entering low power mode is set by the MODE bit of the CPU[CPUx][LP] register.

- MODE bit is b00 - WAIT Mode: Only turn off the processor core clock
- MODE bit is b01 - STOP Mode: Turn off the processor core clock, and transmit shutdown requests upstream on the resource node chain, allowing more resource nodes to be shut down

- MODE bit is b10 - RUN Mode: Keep the processor core clock running continuously
- MODE bit is b11 - Invalid setting

Only when the processor enters STOP mode will it trigger SYSCTL to perform shutdown operations on the resource node chain.

Wake-up from STOP mode is generally triggered by interrupts. Which interrupt sources can wake up the system is configured in the CPU[CPUX][WAKEUP_ENABLE] register of SYSCTL. Setting the bit to 1 means the corresponding interrupt source can wake up the CPU.

The bit assignment of the CPU[CPUX][WAKEUP_ENABLE] register is consistent with the interrupt number defined in the product interrupt vector table.

System wake-up aims to make the CPU exit sleep and be able to process interrupts. Therefore, when an interrupt that can wake up the CPU occurs, the SYSCTL module will automatically turn on all nodes on the resource node chain where the CPU is located in sequence.

13.5.2 State Retention of Resource Nodes in Low Power Mode

For subsystems with independent power switches, if their power and reset resource nodes work in automatic mode, processor sleep may turn off their power, resulting in the loss of all register and memory information in the subsystem.

For resource nodes such as XTAL24M and PLL, if they work in automatic mode, processor sleep may turn them off automatically, resulting in a long time waiting for their frequency to stabilize when the system wakes up.

To individually control the behavior of these resource nodes in low power mode, besides directly modifying their RESOURCE[] register to change MODE from automatic mode to enable mode, it is more recommended to use the RETENTION[CPUX] register to control the behavior of some midstream and upstream resource nodes.

Each bit in the RETENTION register corresponds to a resource node. Setting the bit to 1 can control the resource to ignore downstream shutdown requests and remain on.

SYSCTL provides 1 RETENTION register allocated to CPU0. When the software on CPU0 wants some resource nodes to remain on in low power mode, it only needs to modify the RETENTION register and set the corresponding bit to 1.

For a controllable resource node, as long as the RETENTION register requires it to remain on, it will keep working in low power mode.

Note: The RETENTION configuration is independent of the resource node chain and group connection, and can be understood as a forced enable switch added to some midstream and upstream resource nodes.

The specific bit assignment of the RETENTION register is shown in [Table 29](#):

bit	Resource Node to Keep Enabled
0	System power domain power supply
1	System power domain reset
2	CPU0 subsystem power supply
3	CPU0 subsystem reset
4	CPU1 subsystem power supply
5	CPU1 subsystem reset

bit	Resource Node to Keep Enabled
6	OTN subsystem power supply
7	OTN subsystem reset
8	XTAL24M
9	PLL0
10	PLL1
11	PLL2

Table 29: RETENTION Register Bit Table

13.6 Clock Measurement Module

The clock measurement module allows users to select on-chip clocks for measurement, and saves the current results in the result register.

There are 4 clock measurement units SLICE0~3 in SYSCTL. Users can select the clock to be measured through the SELECTION bit of the MONITOR[SLICEx][CONTROL] register. The measurement clock corresponding to the SELECTION value is shown in Table 30.

Select the measurement reference to be XTAL32K or XTAL24M through the REFERENCE bit of the MONITOR[SLICEx][CONTROL] register, and select the accuracy to be 1KHz or 1Hz through the ACCURAY bit of the MONITOR[SLICEx][CONTROL] register.

SELECTION	Measurement Clock Selection
0	clk_32k_batt
1	clk_32k_pmic
2	clk_rc24m
3	clk_xtal24m
4	clk_usb0_phy
20	clk_24m
21	pll0clk0
22	pll0clk1
23	pll1clk0
24	pll1clk1
25	pll1clk2
26	pll2clk0
27	pll2clk1
128	clk_top_cpu0
129	clk_top_mct0
130	clk_top_cpu1
131	clk_top_mct1
132	clk_top_axif
133	clk_top_axis
134	clk_top_axic

SELECTION	Measurement Clock Selection
135	clk_top_axin
136	clk_top_ahb0
137	clk_top_tmr0
138	clk_top_tmr1
139	clk_top_tmr2
140	clk_top_tmr3
141	clk_top_tmr4
142	clk_top_tmr5
143	clk_top_tmr6
144	clk_top_tmr7
145	clk_top_i2c0
146	clk_top_i2c1
147	clk_top_i2c2
148	clk_top_i2c3
149	clk_top_i2c4
150	clk_top_i2c5
151	clk_top_i2c6
152	clk_top_i2c7
153	clk_top_spi0
154	clk_top_spi1
155	clk_top_spi2
156	clk_top_spi3
157	clk_top_spi4
158	clk_top_spi5
159	clk_top_spi6
160	clk_top_spi7
161	clk_top_uart0
162	clk_top_uart1
163	clk_top_uart2
164	clk_top_uart3
165	clk_top_uart4
166	clk_top_uart5
167	clk_top_uart6
168	clk_top_uart7
169	clk_top_uart8
170	clk_top_uart9
171	clk_top_uart10
172	clk_top_uart11
173	clk_top_uart12
174	clk_top_uart13

SELECTION	Measurement Clock Selection
175	clk_top_uart14
176	clk_top_uart15
177	clk_top_can0
178	clk_top_can1
179	clk_top_can2
180	clk_top_can3
181	clk_top_can4
182	clk_top_can5
183	clk_top_can6
184	clk_top_can7
185	clk_top_xpi0
186	clk_top_dram
187	clk_top_ana0
188	clk_top_ana1
189	clk_top_ana2
190	clk_top_ana3
191	clk_top_aud0
192	clk_top_aud1
193	clk_top_enet0
194	clk_top_ptp0
195	clk_top_ref0
196	clk_top_ref1
197	clk_top_ntm0
198	clk_top_tsw0
199	clk_top_tsw1
200	clk_top_tsw2

Table 30: Measurement Clock Selection Table

After configuration is completed, write 1 to the START bit of the MONITOR[SLICEx][CONTROL] register to start clock measurement. After waiting for the VALID flag bit of the MONITOR[SLICEx][CONTROL] register to be set to 1, users can read the clock frequency information from the MONITOR[SLICEx][CURRENT] register.

Note: The clock measurement function is only applicable to clock frequencies below 400MHz.

13.7 SYSCTL Register Description

The SYSCTL register list is as follows:

SYSCTL base address: 0xF4000000

Offset	Register Name	Description	Reset Value
0x0000	RESOURCE[CPU0]	Resource control register for cpu0_core	0x00000000
0x0004	RESOURCE[CPX0]	Resource control register for cpu0_subsys	0x00000000
0x0050	RESOURCE[PMIC]	Resource control register for pmic	0x00000000
0x0054	RESOURCE[POW_CPU0]	Resource control register for pow_cpu0	0x00000000
0x005C	RESOURCE[RST_SOC]	Resource control register for rst_soc	0x00000000
0x0060	RESOURCE[RST_CPU0]	Resource control register for rst_cpu0	0x00000000
0x0080	RESOURCE[CLK_SRC_XTAL]	Resource control register for xtal	0x00000000
0x0084	RESOURCE[CLK_SRC_PLL0]	Resource control register for pll0	0x00000000
0x0088	RESOURCE[CLK_SRC_CLK0_PLL0]	Resource control register for clk0_pll0	0x00000000
0x008C	RESOURCE[CLK_SRC_CLK1_PLL0]	Resource control register for clk1_pll0	0x00000000
0x0090	RESOURCE[CLK_SRC_PLL1]	Resource control register for pll1	0x00000000
0x0094	RESOURCE[CLK_SRC_CLK0_PLL1]	Resource control register for clk0_pll1	0x00000000
0x0098	RESOURCE[CLK_SRC_CLK1_PLL1]	Resource control register for clk1_pll1	0x00000000
0x009C	RESOURCE[CLK_SRC_CLK2_PLL1]	Resource control register for clk2_pll1	0x00000000
0x00A0	RESOURCE[CLK_SRC_PLL2]	Resource control register for pll2	0x00000000
0x00A4	RESOURCE[CLK_SRC_CLK0_PLL2]	Resource control register for clk0_pll2	0x00000000
0x00A8	RESOURCE[CLK_SRC_CLK1_PLL2]	Resource control register for clk1_pll2	0x00000000
0x00AC	RESOURCE[CLK_SRC_PLL0_REF]	Resource control register for pll0 ref clock	0x00000000
0x00B0	RESOURCE[CLK_SRC_PLL1_REF]	Resource control register for pll1 ref clock	0x00000000
0x00B4	RESOURCE[CLK_SRC_PLL2_REF]	Resource control register for pll2 ref clock	0x00000000

Offset	Register Name	Description	Reset Value
0x0100	RESOURCE[CLK_TOP_CPU0]	Resource control register for clk_top_cpu0	0x00000000
0x0104	RESOURCE[CLK_TOP_MCT0]	Resource control register for clk_top_mct0	0x00000000
0x0108	RESOURCE[CLK_TOP_AHB0]	Resource control register for clk_top_ahb0	0x00000000
0x010C	RESOURCE[CLK_TOP_AXIF]	Resource control register for clk_top_axif	0x00000000
0x0110	RESOURCE[CLK_TOP_AXIS]	Resource control register for clk_top_axis	0x00000000
0x0114	RESOURCE[CLK_TOP_AXIC]	Resource control register for clk_top_axic	0x00000000
0x0118	RESOURCE[CLK_TOP_TMR0]	Resource control register for clk_top_tmr0	0x00000000
0x011C	RESOURCE[CLK_TOP_TMR1]	Resource control register for clk_top_tmr1	0x00000000
0x0120	RESOURCE[CLK_TOP_TMR2]	Resource control register for clk_top_tmr2	0x00000000
0x0124	RESOURCE[CLK_TOP_TMR3]	Resource control register for clk_top_tmr3	0x00000000
0x0128	RESOURCE[CLK_TOP_OWR0]	Resource control register for clk_top_owr0	0x00000000
0x012C	RESOURCE[CLK_TOP_OWR1]	Resource control register for clk_top_owr1	0x00000000
0x0130	RESOURCE[CLK_TOP_I2C0]	Resource control register for clk_top_i2c0	0x00000000
0x0134	RESOURCE[CLK_TOP_I2C1]	Resource control register for clk_top_i2c1	0x00000000
0x0138	RESOURCE[CLK_TOP_I2C2]	Resource control register for clk_top_i2c2	0x00000000
0x013C	RESOURCE[CLK_TOP_I2C3]	Resource control register for clk_top_i2c3	0x00000000
0x0140	RESOURCE[CLK_TOP_SPI0]	Resource control register for clk_top_spi0	0x00000000
0x0144	RESOURCE[CLK_TOP_SPI1]	Resource control register for clk_top_spi1	0x00000000
0x0148	RESOURCE[CLK_TOP_SPI2]	Resource control register for clk_top_spi2	0x00000000
0x014C	RESOURCE[CLK_TOP_SPI3]	Resource control register for clk_top_spi3	0x00000000

Offset	Register Name	Description	Reset Value
0x0150	RESOURCE[CLK_TOP_URT0]	Resource control register for clk_top_urt0	0x00000000
0x0154	RESOURCE[CLK_TOP_URT1]	Resource control register for clk_top_urt1	0x00000000
0x0158	RESOURCE[CLK_TOP_URT2]	Resource control register for clk_top_urt2	0x00000000
0x015C	RESOURCE[CLK_TOP_URT3]	Resource control register for clk_top_urt3	0x00000000
0x0160	RESOURCE[CLK_TOP_URT4]	Resource control register for clk_top_urt4	0x00000000
0x0164	RESOURCE[CLK_TOP_URT5]	Resource control register for clk_top_urt5	0x00000000
0x0168	RESOURCE[CLK_TOP_URT6]	Resource control register for clk_top_urt6	0x00000000
0x016C	RESOURCE[CLK_TOP_URT7]	Resource control register for clk_top_urt7	0x00000000
0x0170	RESOURCE[CLK_TOP_CAN0]	Resource control register for clk_top_can0	0x00000000
0x0174	RESOURCE[CLK_TOP_CAN1]	Resource control register for clk_top_can1	0x00000000
0x0178	RESOURCE[CLK_TOP_CAN2]	Resource control register for clk_top_can2	0x00000000
0x017C	RESOURCE[CLK_TOP_CAN3]	Resource control register for clk_top_can3	0x00000000
0x0180	RESOURCE[CLK_TOP_XPI0]	Resource control register for clk_top_xpi0	0x00000000
0x0184	RESOURCE[CLK_TOP_ESC0]	Resource control register for clk_top_esc0	0x00000000
0x0188	RESOURCE[CLK_TOP_ETH0]	Resource control register for clk_top_eth0	0x00000000
0x018C	RESOURCE[CLK_TOP_PTP0]	Resource control register for clk_top_ptp0	0x00000000
0x0190	RESOURCE[CLK_TOP_REF0]	Resource control register for clk_top_ref0	0x00000000
0x0194	RESOURCE[CLK_TOP_REF1]	Resource control register for clk_top_ref1	0x00000000
0x0198	RESOURCE[CLK_TOP_NTM0]	Resource control register for clk_top_ntm0	0x00000000
0x019C	RESOURCE[CLK_TOP_ANA0]	Resource control register for clk_top_ana0	0x00000000

Offset	Register Name	Description	Reset Value
0x01A0	RESOURCE[CLK_TOP_ANA1]	Resource control register for clk_top_ana1	0x00000000
0x01A4	RESOURCE[CLK_TOP_ADC0]	Resource control register for clk_top_adc0	0x00000000
0x01A8	RESOURCE[CLK_TOP_ADC1]	Resource control register for clk_top_adc1	0x00000000
0x0400	RESOURCE[AHBP]	Resource control register for soc ahb bus	0x00000000
0x0404	RESOURCE[AXIS]	Resource control register for soc axi_fast and axi_main bus	0x00000000
0x0408	RESOURCE[AXIC]	Resource control register for con bus	0x00000000
0x040C	RESOURCE[ROM0]	Resource control register for rom	0x00000000
0x0410	RESOURCE[LMM0]	Resource control register for lmm0	0x00000000
0x0414	RESOURCE[MCT0]	Resource control register for mchtmr0	0x00000000
0x0418	RESOURCE[TMR0]	Resource control register for tmr0	0x00000000
0x041C	RESOURCE[TMR1]	Resource control register for tmr1	0x00000000
0x0420	RESOURCE[TMR2]	Resource control register for tmr2	0x00000000
0x0424	RESOURCE[TMR3]	Resource control register for tmr3	0x00000000
0x0428	RESOURCE[OWR0]	Resource control register for owr0	0x00000000
0x042C	RESOURCE[OWR1]	Resource control register for owr1	0x00000000
0x0430	RESOURCE[EUI0]	Resource control register for eui0	0x00000000
0x0434	RESOURCE[EUI1]	Resource control register for eui1	0x00000000
0x0438	RESOURCE[I2C0]	Resource control register for i2c0	0x00000000
0x043C	RESOURCE[I2C1]	Resource control register for i2c1	0x00000000
0x0440	RESOURCE[I2C2]	Resource control register for i2c2	0x00000000

Offset	Register Name	Description	Reset Value
0x0444	RESOURCE[I2C3]	Resource control register for i2c3	0x00000000
0x0448	RESOURCE[SPI0]	Resource control register for spi0	0x00000000
0x044C	RESOURCE[SPI1]	Resource control register for spi1	0x00000000
0x0450	RESOURCE[SPI2]	Resource control register for spi2	0x00000000
0x0454	RESOURCE[SPI3]	Resource control register for spi3	0x00000000
0x0458	RESOURCE[URT0]	Resource control register for uart0	0x00000000
0x045C	RESOURCE[URT1]	Resource control register for uart1	0x00000000
0x0460	RESOURCE[URT2]	Resource control register for uart2	0x00000000
0x0464	RESOURCE[URT3]	Resource control register for uart3	0x00000000
0x0468	RESOURCE[URT4]	Resource control register for uart4	0x00000000
0x046C	RESOURCE[URT5]	Resource control register for uart5	0x00000000
0x0470	RESOURCE[URT6]	Resource control register for uart6	0x00000000
0x0474	RESOURCE[URT7]	Resource control register for uart7	0x00000000
0x0478	RESOURCE[CRC0]	Resource control register for crc	0x00000000
0x047C	RESOURCE[TSNS]	Resource control register for tsns	0x00000000
0x0480	RESOURCE[WDG0]	Resource control register for wdg0	0x00000000
0x0484	RESOURCE[WDG1]	Resource control register for wdg1	0x00000000
0x0488	RESOURCE[MBX0]	Resource control register for mbx0	0x00000000
0x048C	RESOURCE[GPIO]	Resource control register for gpio	0x00000000
0x0490	RESOURCE[PPI0]	Resource control register for ppi	0x00000000
0x0494	RESOURCE[HDMA]	Resource control register for hdma	0x00000000

Offset	Register Name	Description	Reset Value
0x0498	RESOURCE[LOBS]	Resource control register for lobs	0x00000000
0x049C	RESOURCE[ADC0]	Resource control register for adc0	0x00000000
0x04A0	RESOURCE[ADC1]	Resource control register for adc1	0x00000000
0x04A4	RESOURCE[COMP0]	Resource control register for acmp0	0x00000000
0x04A8	RESOURCE[CAN0]	Resource control register for can0	0x00000000
0x04AC	RESOURCE[CAN1]	Resource control register for can1	0x00000000
0x04B0	RESOURCE[CAN2]	Resource control register for can2	0x00000000
0x04B4	RESOURCE[CAN3]	Resource control register for can3	0x00000000
0x04B8	RESOURCE[PTPC]	Resource control register for ptpc	0x00000000
0x04BC	RESOURCE[QEIO]	Resource control register for qei0	0x00000000
0x04C0	RESOURCE[QEIO1]	Resource control register for qei1	0x00000000
0x04C4	RESOURCE[QEO0]	Resource control register for qeo0	0x00000000
0x04C8	RESOURCE[QEO1]	Resource control register for qeo1	0x00000000
0x04CC	RESOURCE[SDM0]	Resource control register for plb	0x00000000
0x04D0	RESOURCE[PLB0]	Resource control register for plb	0x00000000
0x04D4	RESOURCE[PWM0]	Resource control register for pwm0	0x00000000
0x04D8	RESOURCE[PWM1]	Resource control register for pwm1	0x00000000
0x04DC	RESOURCE[EMDS]	Resource control register for motor misc	0x00000000
0x04E0	RESOURCE[XPIO]	Resource control register for xpi	0x00000000
0x04E4	RESOURCE[RAM0]	Resource control register for ram	0x00000000
0x04E8	RESOURCE[XDMA]	Resource control register for xdma	0x00000000
0x04EC	RESOURCE[ESC0]	Resource control register for esc	0x00000000

Offset	Register Name	Description	Reset Value
0x04F0	RESOURCE[ETH0]	Resource control register for ethernet	0x00000000
0x04F4	RESOURCE[USB0]	Resource control register for usb	0x00000000
0x04F8	RESOURCE[NTM0]	Resource control register for ntimer	0x00000000
0x04FC	RESOURCE[REF0]	Resource control register for ref0	0x00000000
0x0500	RESOURCE[REF1]	Resource control register for ref1	0x00000000
0x0800	GROUP0[LINK0][VALUE]	Group setting	0x00000000
0x0804	GROUP0[LINK0][SET]	Group setting	0x00000000
0x0808	GROUP0[LINK0][CLEAR]	Group setting	0x00000000
0x080C	GROUP0[LINK0][TOGGLE]	Group setting	0x00000000
0x0810	GROUP0[LINK1][VALUE]	Group setting	0x00000000
0x0814	GROUP0[LINK1][SET]	Group setting	0x00000000
0x0818	GROUP0[LINK1][CLEAR]	Group setting	0x00000000
0x081C	GROUP0[LINK1][TOGGLE]	Group setting	0x00000000
0x0820	GROUP0[LINK2][VALUE]	Group setting	0x00000000
0x0824	GROUP0[LINK2][SET]	Group setting	0x00000000
0x0828	GROUP0[LINK2][CLEAR]	Group setting	0x00000000
0x082C	GROUP0[LINK2][TOGGLE]	Group setting	0x00000000
0x0830	GROUP0[LINK3][VALUE]	Group setting	0x00000000
0x0834	GROUP0[LINK3][SET]	Group setting	0x00000000
0x0838	GROUP0[LINK3][CLEAR]	Group setting	0x00000000
0x083C	GROUP0[LINK3][TOGGLE]	Group setting	0x00000000
0x0840	GROUP1[LINK0][VALUE]	Group setting	0x00000000
0x0844	GROUP1[LINK0][SET]	Group setting	0x00000000
0x0848	GROUP1[LINK0][CLEAR]	Group setting	0x00000000
0x084C	GROUP1[LINK0][TOGGLE]	Group setting	0x00000000
0x0850	GROUP1[LINK1][VALUE]	Group setting	0x00000000
0x0854	GROUP1[LINK1][SET]	Group setting	0x00000000
0x0858	GROUP1[LINK1][CLEAR]	Group setting	0x00000000
0x085C	GROUP1[LINK1][TOGGLE]	Group setting	0x00000000
0x0860	GROUP1[LINK2][VALUE]	Group setting	0x00000000
0x0864	GROUP1[LINK2][SET]	Group setting	0x00000000
0x0868	GROUP1[LINK2][CLEAR]	Group setting	0x00000000
0x086C	GROUP1[LINK2][TOGGLE]	Group setting	0x00000000
0x0870	GROUP1[LINK3][VALUE]	Group setting	0x00000000
0x0874	GROUP1[LINK3][SET]	Group setting	0x00000000

Offset	Register Name	Description	Reset Value
0x0878	GROUP1[LINK3][CLEAR]	Group setting	0x00000000
0x087C	GROUP1[LINK3][TOGGLE]	Group setting	0x00000000
0x0900	AFFILIATE[CPU0][VALUE]	Affiliate of Group	0x00000000
0x0904	AFFILIATE[CPU0][SET]	Affiliate of Group	0x00000000
0x0908	AFFILIATE[CPU0][CLEAR]	Affiliate of Group	0x00000000
0x090C	AFFILIATE[CPU0][TOGGLE]	Affiliate of Group	0x00000000
0x0920	RETENTION[CPU0][VALUE]	Retention Contol	0x00000000
0x0924	RETENTION[CPU0][SET]	Retention Contol	0x00000000
0x0928	RETENTION[CPU0][CLEAR]	Retention Contol	0x00000000
0x092C	RETENTION[CPU0][TOGGLE]	Retention Contol	0x00000000
0x1000	POWER[CPU0][STATUS]	Power Setting	0x80000000
0x1004	POWER[CPU0][LF_WAIT]	Power Setting	0x000000FF
0x100C	POWER[CPU0][OFF_WAIT]	Power Setting	0x0000000F
0x1400	RESET[SOC][CONTROL]	Reset Setting	0x80000000
0x1404	RESET[SOC][CONFIG]	Reset Setting	0x00402003
0x140C	RESET[SOC][COUNTER]	Reset Setting	0x00000000
0x1410	RESET[CPU0][CONTROL]	Reset Setting	0x80000000
0x1414	RESET[CPU0][CONFIG]	Reset Setting	0x00402003
0x141C	RESET[CPU0][COUNTER]	Reset Setting	0x00000000
0x1800	CLOCK[CLK_TOP_CPU0]	Clock setting	0x00000000
0x1804	CLOCK[CLK_TOP_MCT0]	Clock setting	0x00000000
0x1808	CLOCK[CLK_TOP_AHB0]	Clock setting	0x00000000
0x180C	CLOCK[CLK_TOP_AXIF]	Clock setting	0x00000000
0x1810	CLOCK[CLK_TOP_AXIS]	Clock setting	0x00000000
0x1814	CLOCK[CLK_TOP_AXIC]	Clock setting	0x00000000
0x1818	CLOCK[CLK_TOP_TMR0]	Clock setting	0x00000000
0x181C	CLOCK[CLK_TOP_TMR1]	Clock setting	0x00000000
0x1820	CLOCK[CLK_TOP_TMR2]	Clock setting	0x00000000
0x1824	CLOCK[CLK_TOP_TMR3]	Clock setting	0x00000000
0x1828	CLOCK[CLK_TOP_OWR0]	Clock setting	0x00000000
0x182C	CLOCK[CLK_TOP_OWR1]	Clock setting	0x00000000
0x1830	CLOCK[CLK_TOP_I2C0]	Clock setting	0x00000000
0x1834	CLOCK[CLK_TOP_I2C1]	Clock setting	0x00000000
0x1838	CLOCK[CLK_TOP_I2C2]	Clock setting	0x00000000
0x183C	CLOCK[CLK_TOP_I2C3]	Clock setting	0x00000000
0x1840	CLOCK[CLK_TOP_SPI0]	Clock setting	0x00000000
0x1844	CLOCK[CLK_TOP_SPI1]	Clock setting	0x00000000
0x1848	CLOCK[CLK_TOP_SPI2]	Clock setting	0x00000000
0x184C	CLOCK[CLK_TOP_SPI3]	Clock setting	0x00000000
0x1850	CLOCK[CLK_TOP_URT0]	Clock setting	0x00000000

Offset	Register Name	Description	Reset Value
0x1854	CLOCK[CLK_TOP_URT1]	Clock setting	0x00000000
0x1858	CLOCK[CLK_TOP_URT2]	Clock setting	0x00000000
0x185C	CLOCK[CLK_TOP_URT3]	Clock setting	0x00000000
0x1860	CLOCK[CLK_TOP_URT4]	Clock setting	0x00000000
0x1864	CLOCK[CLK_TOP_URT5]	Clock setting	0x00000000
0x1868	CLOCK[CLK_TOP_URT6]	Clock setting	0x00000000
0x186C	CLOCK[CLK_TOP_URT7]	Clock setting	0x00000000
0x1870	CLOCK[CLK_TOP_CAN0]	Clock setting	0x00000000
0x1874	CLOCK[CLK_TOP_CAN1]	Clock setting	0x00000000
0x1878	CLOCK[CLK_TOP_CAN2]	Clock setting	0x00000000
0x187C	CLOCK[CLK_TOP_CAN3]	Clock setting	0x00000000
0x1880	CLOCK[CLK_TOP_XPI0]	Clock setting	0x00000000
0x1884	CLOCK[CLK_TOP_ESC0]	Clock setting	0x00000000
0x1888	CLOCK[CLK_TOP_ETH0]	Clock setting	0x00000000
0x188C	CLOCK[CLK_TOP_PTP0]	Clock setting	0x00000000
0x1890	CLOCK[CLK_TOP_REF0]	Clock setting	0x00000000
0x1894	CLOCK[CLK_TOP_REF1]	Clock setting	0x00000000
0x1898	CLOCK[CLK_TOP_NTM0]	Clock setting	0x00000000
0x189C	CLOCK[CLK_TOP_ANA0]	Clock setting	0x00000000
0x18A0	CLOCK[CLK_TOP_ANA1]	Clock setting	0x00000000
0x1C00	ADCCLK[CLK_TOP_ADC0]	Clock setting	0x00000000
0x1C04	ADCCLK[CLK_TOP_ADC1]	Clock setting	0x00000000
0x2000	GLOBAL00	Clock senario	0x00000000
0x2400	MONITOR[SLICE0][CONTROL]	Clock measure and monitor control	0x00000000
0x2404	MONITOR[SLICE0][CURRENT]	Clock measure result	0x00000000
0x2408	MONITOR[SLICE0][LOW_LIMIT]	Clock lower limit	0xFFFFFFFF
0x240C	MONITOR[SLICE0][HIGH_LIMIT]	Clock upper limit	0x00000000
0x2420	MONITOR[SLICE1][CONTROL]	Clock measure and monitor control	0x00000000
0x2424	MONITOR[SLICE1][CURRENT]	Clock measure result	0x00000000
0x2428	MONITOR[SLICE1][LOW_LIMIT]	Clock lower limit	0xFFFFFFFF
0x242C	MONITOR[SLICE1][HIGH_LIMIT]	Clock upper limit	0x00000000
0x2440	MONITOR[SLICE2][CONTROL]	Clock measure and monitor control	0x00000000
0x2444	MONITOR[SLICE2][CURRENT]	Clock measure result	0x00000000
0x2448	MONITOR[SLICE2][LOW_LIMIT]	Clock lower limit	0xFFFFFFFF

Offset	Register Name	Description	Reset Value
0x244C	MONITOR[SLICE2][HIGH_LIMIT]	Clock upper limit	0x00000000
0x2460	MONITOR[SLICE3][CONTROL]	Clock measure and monitor control	0x00000000
0x2464	MONITOR[SLICE3][CURRENT]	Clock measure result	0x00000000
0x2468	MONITOR[SLICE3][LOW_LIMIT]	Clock lower limit	0xFFFFFFFF
0x246C	MONITOR[SLICE3][HIGH_LIMIT]	Clock upper limit	0x00000000
0x2800	CPU[CPU0][LP]	CPU0 LP control	0x00001000
0x2804	CPU[CPU0][LOCK]	CPU0 Lock GPR	0x00000000
0x2808	CPU[CPU0][GPR][GPR0]	CPU0 GPR0	0x00000000
0x280C	CPU[CPU0][GPR][GPR1]	CPU0 GPR1	0x00000000
0x2810	CPU[CPU0][GPR][GPR2]	CPU0 GPR2	0x00000000
0x2814	CPU[CPU0][GPR][GPR3]	CPU0 GPR3	0x00000000
0x2818	CPU[CPU0][GPR][GPR4]	CPU0 GPR4	0x00000000
0x281C	CPU[CPU0][GPR][GPR5]	CPU0 GPR5	0x00000000
0x2820	CPU[CPU0][GPR][GPR6]	CPU0 GPR6	0x00000000
0x2824	CPU[CPU0][GPR][GPR7]	CPU0 GPR7	0x00000000
0x2828	CPU[CPU0][GPR][GPR8]	CPU0 GPR8	0x00000000
0x282C	CPU[CPU0][GPR][GPR9]	CPU0 GPR9	0x00000000
0x2830	CPU[CPU0][GPR][GPR10]	CPU0 GPR10	0x00000000
0x2834	CPU[CPU0][GPR][GPR11]	CPU0 GPR11	0x00000000
0x2838	CPU[CPU0][GPR][GPR12]	CPU0 GPR12	0x00000000
0x283C	CPU[CPU0][GPR][GPR13]	CPU0 GPR13	0x00000000
0x2840	CPU[CPU0][WAKEUP_STATUS][STATUS0]	CPU0 wakeup IRQ status	0x00000000
0x2844	CPU[CPU0][WAKEUP_STATUS][STATUS1]	CPU0 wakeup IRQ status	0x00000000
0x2848	CPU[CPU0][WAKEUP_STATUS][STATUS2]	CPU0 wakeup IRQ status	0x00000000
0x284C	CPU[CPU0][WAKEUP_STATUS][STATUS3]	CPU0 wakeup IRQ status	0x00000000
0x2850	CPU[CPU0][WAKEUP_STATUS][STATUS4]	CPU0 wakeup IRQ status	0x00000000
0x2854	CPU[CPU0][WAKEUP_STATUS][STATUS5]	CPU0 wakeup IRQ status	0x00000000
0x2880	CPU[CPU0][WAKEUP_ENABLE][ENABLE0]	CPU0 wakeup IRQ enable	0x00000000
0x2884	CPU[CPU0][WAKEUP_ENABLE][ENABLE1]	CPU0 wakeup IRQ enable	0x00000000
0x2888	CPU[CPU0][WAKEUP_ENABLE][ENABLE2]	CPU0 wakeup IRQ enable	0x00000000
0x288C	CPU[CPU0][WAKEUP_ENABLE][ENABLE3]	CPU0 wakeup IRQ enable	0x00000000
0x2890	CPU[CPU0][WAKEUP_ENABLE][ENABLE4]	CPU0 wakeup IRQ enable	0x00000000
0x2894	CPU[CPU0][WAKEUP_ENABLE][ENABLE5]	CPU0 wakeup IRQ enable	0x00000000

Table 31: SYSCTL Register List

Note: The boot code of the product ROM will configure this module, so the register values read by the software may be inconsistent with the reset values described in this chapter.

13.8 Detailed Register Information

The detailed description of SYSCTL registers is as follows:

13.8.1 RESOURCE (0x0 + 0x4 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GLB_BUSY	LOC_BUSY	RSVD																												MODE	
RO	RO	N/A																												RW	
0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	

RESOURCE [31:0]

Bits	Name	Description
31	GLB_BUSY	global busy 0: no changes pending to any nodes 1: any of nodes is changing status
30	LOC_BUSY	local busy 0: no change is pending for current node 1: current node is changing status
1-0	MODE	resource work mode 0:auto turn on and off as system required(recommended) 1:always on 2:always off 3:reserved

RESOURCE Bits

13.8.2 GROUP0[VALUE] (0x800 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LINK																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GROUP0[VALUE] [31:0]

Bits	Name	Description
31-0	LINK	dependency on peripherals, index count from resource ahbp(0x400), each bit represents a peripheral 0: peripheral is not needed 1: periphera is needed

Bits	Name	Description

GROUP0[VALUE] Bits

13.8.3 GROUP0[SET] (0x804 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LINK																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GROUP0[SET] [31:0]

Bits	Name	Description
31-0	LINK	dependency on peripherals, index count from resource ahbp(0x400), each bit represents a peripheral 0: no effect 1: add periphra into this group, periphra is needed

GROUP0[SET] Bits

13.8.4 GROUP0[CLEAR] (0x808 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LINK																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GROUP0[CLEAR] [31:0]

Bits	Name	Description
31-0	LINK	dependency on peripherals, index count from resource ahbp(0x400), each bit represents a peripheral 0: no effect 1: delete periphra in this group, periphra is not needed

GROUP0[CLEAR] Bits

13.8.5 GROUP0[TOGGLE] (0x80C + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LINK																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GROUP0[TOGGLE] [31:0]

Bits	Name	Description
31-0	LINK	dependency on peripherals, index count from resource ahbp(0x400), each bit represents a peripheral 0: no effect 1: toggle the result that whether periphra is needed before

GROUP0[TOGGLE] Bits

13.8.6 GROUP1[VALUE] (0x840 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LINK																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GROUP1[VALUE] [31:0]

Bits	Name	Description
31-0	LINK	dependency on peripherals, index count from resource ahbp(0x400), each bit represents a peripheral 0: peripheral is not needed 1: periphra is needed

GROUP1[VALUE] Bits

13.8.7 GROUP1[SET] (0x844 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LINK																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GROUP1[SET] [31:0]

Bits	Name	Description
31-0	LINK	dependency on peripherals, index count from resource ahbp(0x400), each bit represents a peripheral 0: no effect 1: add periphra into this group, periphra is needed

GROUP1[SET] Bits

13.8.8 GROUP1[CLEAR] (0x848 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LINK																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GROUP1[CLEAR] [31:0]

Bits	Name	Description
31-0	LINK	dependency on peripherals, index count from resource ahbp(0x400), each bit represents a peripheral 0: no effect 1: delete periphra in this group, periphra is not needed

GROUP1[CLEAR] Bits

13.8.9 GROUP1[TOGGLE] (0x84C + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LINK																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GROUP1[TOGGLE] [31:0]

Bits	Name	Description
31-0	LINK	dependency on peripherals, index count from resource ahbp(0x400), each bit represents a peripheral 0: no effect 1: toggle the result that whether periphra is needed before

GROUP1[TOGGLE] Bits

13.8.10 AFFILIATE[VALUE] (0x900 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																												LINK			
N/A																												RW			
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

AFFILIATE[VALUE] [31:0]

Bits	Name	Description
3-0	LINK	Affiliate groups of cpu0, each bit represents a group bit0: cpu0 depends on group0 bit1: cpu0 depends on group1 bit2: cpu0 depends on group2 bit3: cpu0 depends on group3

AFFILIATE[VALUE] Bits

13.8.11 AFFILIATE[SET] (0x904 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																												LINK			
N/A																												RW			
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

AFFILIATE[SET] [31:0]

Bits	Name	Description
3-0	LINK	Affiliate groups of cpu0, each bit represents a group 0: no effect 1: the group is assigned to CPU0

AFFILIATE[SET] Bits

13.8.12 AFFILIATE[CLEAR] (0x908 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																												LINK			
N/A																												RW			
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

AFFILIATE[CLEAR] [31:0]

Bits	Name	Description
3-0	LINK	Affiliate groups of cpu0, each bit represents a group 0: no effect 1: the group is not assigned to CPU0

AFFILIATE[**CLEAR**] Bits

13.8.13 AFFILIATE[TOGGLE] (0x90C + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																												LINK			
N/A																												RW			
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

AFFILIATE[**TOGGLE**] [31:0]

Bits	Name	Description
3-0	LINK	Affiliate groups of cpu0, each bit represents a group 0: no effect 1: toggle the result that whether the group is assigned to CPU0 before

AFFILIATE[**TOGGLE**] Bits

13.8.14 RETENTION[VALUE] (0x920 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																LINK															
N/A																RW															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RETENTION[**VALUE**] [31:0]

Bits	Name	Description
14-0	LINK	retention setting while CPU0 enter stop mode, each bit represents a resource bit00: soc_mem is kept on while cpu0 stop bit01: soc_ctx is kept on while cpu0 stop bit02: cpu0_mem is kept on while cpu0 stop bit03: cpu0_ctx is kept on while cpu0 stop bit04: xtal_hold is kept on while cpu0 stop bit05: pll0_hold is kept on while cpu0 stop bit06: pll1_hold is kept on while cpu0 stop

RETENTION[VALUE] Bits

13.8.15 RETENTION[SET] (0x924 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																	LINK														
N/A																	RW														
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RETENTION[SET] [31:0]

Bits	Name	Description
14-0	LINK	retention setting while CPU0 enter stop mode, each bit represents a resource 0: no effect 1: keep

RETENTION[SET] Bits

13.8.16 RETENTION[CLEAR] (0x928 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																	LINK														
N/A																	RW														
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RETENTION[CLEAR] [31:0]

Bits	Name	Description
14-0	LINK	retention setting while CPU0 enter stop mode, each bit represents a resource 0: no effect 1: no keep

RETENTION[CLEAR] Bits

13.8.17 RETENTION[TOGGLE] (0x92C + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																LINK															
N/A																RW															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RETENTION[TOGGLE] [31:0]

Bits	Name	Description
14-0	LINK	retention setting while CPU0 enter stop mode, each bit represents a resource 0: no effect 1: toggle the result that whether the resource is kept on while CPU0 stop before

RETENTION[TOGGLE] Bits

13.8.18 POWER[STATUS] (0x1000 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
FLAG	FLAG_WAKE	RSVD														MEM_RET_N	MEM_RET_P	RSVD		LF_DISABLE	RSVD		LF_ACK	RSVD									
RW	RW	N/A														RO	RO	N/A		RO	N/A		RO	N/A									
1	0	x	x	x	x	x	x	x	x	x	x	x	x	0	0	x	x	x	0	x	x	x	0	x	x	x	x	x	x	x	x		

POWER[STATUS] [31:0]

Bits	Name	Description
31	FLAG	flag represents power cycle happened from last clear of this bit 0: power domain did not edurance power cycle since last clear of this bit 1: power domain enduranced power cycle since last clear of this bit
30	FLAG_WAKE	flag represents wakeup power cycle happened from last clear of this bit 0: power domain did not edurance wakeup power cycle since last clear of this bit 1: power domain enduranced wakeup power cycle since last clear of this bit
17	MEM_RET_N	memory info retention control signal 0: memory enter retention mode 1: memory exit retention mode
16	MEM_RET_P	memory info retention control signal 0: memory not enterexitretention mode 1: memory enter retention mode
12	LF_DISABLE	low fanout power switch disable 0: low fanout power switches are turned on 1: low fanout power switches are truned off
8	LF_ACK	low fanout power switch feedback 0: low fanout power switches are turned on 1: low fanout power switches are truned off

POWER[STATUS] Bits

13.8.19 POWER[LF_WAIT] (0x1004 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD												WAIT																			
N/A												RW																			
x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	

POWER[LF_WAIT] [31:0]

Bits	Name	Description
19-0	WAIT	wait time for low fan out power switch turn on, default value is 255 0: 0 clock cycle 1: 1 clock cycles ... clock cycles count on 24MHz

POWER[LF_WAIT] Bits

13.8.20 POWER[OFF_WAIT] (0x100C + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD												WAIT																			
N/A												RW																			
x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

POWER[OFF_WAIT] [31:0]

Bits	Name	Description
19-0	WAIT	wait time for power switch turn off, default value is 15 0: 0 clock cycle 1: 1 clock cycles ... clock cycles count on 24MHz

POWER[OFF_WAIT] Bits

13.8.21 RESET[CONTROL] (0x1400 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLAG	FLAG_WAKE	RSVD																								HOLD	RSVD		RESET		
RW	RW	N/A																								RW	N/A		RW		
1	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	x	x	x	0	

RESET[CONTROL] [31:0]

Bits	Name	Description
31	FLAG	flag represents reset happened from last clear of this bit 0: domain did not edurance reset cycle since last clear of this bit 1: domain endured reset cycle since last clear of this bit

Bits	Name	Description
30	FLAG_WAKE	flag represents wakeup reset happened from last clear of this bit 0: domain did not edurance wakeup reset cycle since last clear of this bit 1: domain endured wakeup reset cycle since last clear of this bit
4	HOLD	perform reset and hold in reset, until ths bit cleared by software 0: reset is released for function 1: reset is assert and hold
0	RESET	perform reset and release imediately 0: reset is released 1 reset is asserted and will release automatically

RESET[CONTROL] Bits

13.8.22 RESET[CONFIG] (0x1404 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								PRE_WAIT								RSTCLK_NUM								POST_WAIT							
N/A								RW								RW								RW							
x	x	x	x	x	x	x	x	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1

RESET[CONFIG] [31:0]

Bits	Name	Description
23-16	PRE_WAIT	wait cycle numbers before assert reset 0: wait 0 cycle 1: wait 1 cycles ... Note, clock cycle is base on 24M
15-8	RSTCLK_NUM	reset clock number(must be even number) 0: 0 cycle 1: 0 cycles 2: 2 cycles 3: 2 cycles ... Note, clock cycle is base on 24M

Bits	Name	Description
7-0	POST_WAIT	time guard band for reset release 0: wait 0 cycle 1: wait 1 cycles ... Note, clock cycle is base on 24M

RESET[CONFIG] Bits

13.8.23 RESET[COUNTER] (0x140C + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD												COUNTER																			
N/A												RW																			
x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RESET[COUNTER] [31:0]

Bits	Name	Description
19-0	COUNTER	self clear trigger counter, reset triggered when counter value is 1, write 0 will cancel reset 0: wait 0 cycle 1: wait 1 cycles ... Note, clock cycle is base on 24M

RESET[COUNTER] Bits

13.8.24 CLOCK (0x1800 + 0x4 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GLB_BUSY	LOC_BUSY	RSVD	PRESERVE	RSVD																	MUX			DIV							
RO	RO	N/A	RW	N/A																	RW			RW							
0	0	x	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0

CLOCK [31:0]

Bits	Name	Description
31	GLB_BUSY	global busy 0: no changes pending to any clock 1: any of nodes is changing status
30	LOC_BUSY	local busy 0: a change is pending for current node 1: current node is changing status
28	PRESERVE	preserve function against global select 0: select global clock setting 1: not select global clock setting
10-8	MUX	current mux in clock component 0:osc0_clk0 1:pll0_clk0 2:pll0_clk1 3:pll1_clk0 4:pll1_clk1 5:pll1_clk2 6:pll2_clk0 7:pll2_clk1
7-0	DIV	clock divider 0: divider by 1 1: divider by 2 2: divider by 3 ... 255: divider by 256

CLOCK Bits

13.8.25 ADCCLK (0x1C00 + 0x4 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GLB_BUSY	LOC_BUSY	RSVD	PRESERVE	RSVD																		MUX	RSVD								
RO	RO	N/A	RW	N/A																		RW	N/A								
0	0	x	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	x	x	x	x	x	x	x	

ADCCLK [31:0]

Bits	Name	Description
31	GLB_BUSY	global busy 0: no changes pending to any clock 1: any of nodes is changing status

Bits	Name	Description
30	LOC_BUSY	local busy 0: a change is pending for current node 1: current node is changing status
28	PRESERVE	preserve function against global select 0: select global clock setting 1: not select global clock setting
8	MUX	current mux 0: ahb0 clock N 1: ana clock

ADCCLK Bits

13.8.26 GLOBAL00 (0x2000)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								MUX							
N/A																								RW							
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	

GLOBAL00 [31:0]

Bits	Name	Description
7-0	MUX	global clock override request bit0: override to preset0 bit1: override to preset1 bit2: override to preset2 bit3: override to preset3 bit4: override to preset4 bit5: override to preset5 bit6: override to preset6 bit7: override to preset7

GLOBAL00 Bits

13.8.27 MONITOR[CONTROL] (0x2400 + 0x20 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALID	RSVD			DIV_BUSY	RSVD		OUTEN	DIV								HIGH	LOW	RSVD	START	RSVD	MODE	ACCURACY	REFERENCE	SELECTION							
RW	N/A			RO	N/A		RW	RW								RW	RW	N/A	RW	N/A	RW	RW	RW	RW	RW						

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	x	x	x	0	x	x	0	0	0	0	0	0	0	0	0	0	0	x	0	x	0	0	0	0	0	0	0	0	0	0	0

MONITOR[CONTROL] [31:0]

Bits	Name	Description
31	VALID	result is ready for read 0: not ready 1: result is ready
27	DIV_BUSY	divider is applying new setting
24	OUTEN	enable clock output
23-16	DIV	output divider
15	HIGH	clock frequency higher than upper limit
14	LOW	clock frequency lower than lower limit
12	START	start measurement
10	MODE	work mode, 0: register value will be compared to measurement 1: upper and lower value will be recordered in register
9	ACCURACY	measurement accuracy, 0: resolution is 1kHz 1: resolution is 1Hz
8	REFERENCE	refrence clock selection, 0: 32k 1: 24M
7-0	SELECTION	clock measurement selection

MONITOR[CONTROL] Bits

13.8.28 MONITOR[CURRENT] (0x2404 + 0x20 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																FREQUENCY															
																RO															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MONITOR[CURRENT] [31:0]

Bits	Name	Description
31-0	FREQUENCY	self updating measure result

MONITOR[CURRENT] Bits

13.8.29 MONITOR[LOW_LIMIT] (0x2408 + 0x20 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FREQUENCY																															
RW																															
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	

MONITOR[LOW_LIMIT] [31:0]

Bits	Name	Description
31-0	FREQUENCY	lower frequency

MONITOR[LOW_LIMIT] Bits

13.8.30 MONITOR[HIGH_LIMIT] (0x240C + 0x20 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																FREQUENCY																
																RW																
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

MONITOR[HIGH_LIMIT] [31:0]

Bits	Name	Description
31-0	FREQUENCY	upper frequency

MONITOR[HIGH_LIMIT] Bits

13.8.31 CPU[LP] (0x2800 + 0x400 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
WAKE_CNT								RSVD								HALT	RSVD		WAKE	EXEC	RSVD	WAKE_FLAG	SLEEP_FLAG	RESET_FLAG	RSVD								MODE
RW								N/A								RW	N/A		RO	RO	N/A	RW	RW	RW	N/A								RW
0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	0	x	x	0	1	x	0	0	0	x	x	x	x	x	x	0	0		

CPU[LP] [31:0]

Bits	Name	Description
31-24	WAKE_CNT	CPU0 wake up counter, counter satuated at 255, write 0x00 to clear

Bits	Name	Description
16	HALT	halt request for CPU0, 0: CPU0 will start to execute after reset or receive wakeup request 1: CPU0 will not start after reset, or wakeup after WFI
13	WAKE	CPU0 is waking up 0: CPU0 wake up not asserted 1: CPU0 wake up asserted
12	EXEC	CPU0 is executing 0: CPU0 is not executing 1: CPU0 is executing
10	WAKE_FLAG	CPU0 wakeup flag, indicate a wakeup event got active, write 1 to clear this bit 0: CPU0 wakeup not happened 1: CPU0 wake up happened
9	SLEEP_FLAG	CPU0 sleep flag, indicate a sleep event got active, write 1 to clear this bit 0: CPU0 sleep not happened 1: CPU0 sleep happened
8	RESET_FLAG	CPU0 reset flag, indicate a reset event got active, write 1 to clear this bit 0: CPU0 reset not happened 1: CPU0 reset happened
1-0	MODE	Low power mode, system behavior after WFI 00: CPU clock stop after WFI 01: System enter low power mode after WFI 10: Keep running after WFI 11: reserved

CPU[LP] Bits

13.8.32 CPU[LOCK] (0x2804 + 0x400 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																GPR														LOCK	RSVD
N/A																RW														RW	N/A
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x

CPU[LOCK] [31:0]

Bits	Name	Description
15-2	GPR	Lock bit for CPU_DATA0 to CPU_DATA13, once set, this bit will not clear until next reset
1	LOCK	Lock bit for CPU_LOCK

CPU[LOCK] Bits

13.8.33 CPU[GPR] (0x2808 + 0x400 * n + 0x4 * m)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPR																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CPU[GPR] [31:0]

Bits	Name	Description
31-0	GPR	register for software to handle resume, can save resume address or status

CPU[GPR] Bits

13.8.34 CPU[WAKEUP_STATUS] (0x2840 + 0x400 * n + 0x4 * m)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STATUS																															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CPU[WAKEUP_STATUS] [31:0]

Bits	Name	Description
31-0	STATUS	IRQ values

CPU[WAKEUP_STATUS] Bits

13.8.35 CPU[WAKEUP_ENABLE] (0x2880 + 0x400 * n + 0x4 * m)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE																															
RW																															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CPU[WAKEUP_ENABLE] [31:0]

Bits	Name	Description
31-0	ENABLE	IRQ wakeup enable

CPU[WAKEUP_ENABLE] Bits

14 PLL Controller PLLCTL

This chapter describes the main functions and characteristics of the phase-locked loop controller PLLCTL.

14.1 Feature Summary

This product includes a phase-locked loop controller PLLCTL that is used to configure the on-chip PLL. The main characteristics are as follows:

- XTAL oscillator configuration
- On-chip PLL operating mode configuration
 - Configure the PLL reference clock
 - Configure the PLL operating frequency
 - Configure PLL spread spectrum function
- Configure output frequency division
- supports hardware automatic control
- supports runtime modification frequency

14.2 Function Description

This chapter describes the main functions of the phase-locked loop controller PLLCTL.

14.2.1 XTAL oscillator

The XTAL oscillator can read the working status of XTAL. The user can query the ENABLE flag bit in the XTAL register. Setting to 1 means that the crystal oscillator as the reference clock source is turned on, and querying the RESPONSE flag bit in the XTAL register. When setting to 1, it means that the reference clock source is in stable state. You can set the time required for XTAL to start, and the reset value of this time is 3 ms. This setting affects the entry and exit time of turning off XTAL's low power mode.

14.2.2 PLL reference clock settings

The phase-locked loop reference clock can be selected through REFSEL:

- 0: Select XTAL
- 1: Select RC24M

This option can be modified while the PLL is operating, which may cause brief fluctuations in the PLL output clock frequency.

14.2.3 PLL operating frequency configuration

PLL voltage controlled oscillator output frequency is as follows:

$$F_{vco} = F_{ref} \times (MFI + (MFN \div MFD))$$

MFD is 24000000, and the number 1 on the MFN represents 0.1Hz.

The MFI and MFN fields of the PLL register support run-time modifications. After modification, you can read the BUSY flag to determine whether the PLL is operating stably under the new settings. The PLL does not stop

clock output when switching to a new setting. MFD does not support runtime modifications. If modifications are made, they will take effect after the PLL is turned off and turned on again.

14.2.4 PLL spread spectrum mode

The spread mode of the PLL can be enabled via SPREAD:

- 0: Prohibition of spectrum spreading
- 1: Enable spectrum spreading

Spread spectrum can be enabled or disabled while the PLL is operating. The spread spectrum of the PLL is downward spread spectrum. When the spread spectrum of the PLL is enabled, the VCO oscillation frequency is swept between the upper and lower limits of the spread spectrum, and

$$F_{ref} \times (STEP \div MFD)$$

increases or decreases every 24M clock cycle.

The upper limit of VCO oscillation frequency is:

$$F_{vco} = F_{ref} \times (MFI + (MFN \div MFD))$$

The lower limit of VCO oscillation frequency is:

$$F_{vco} = F_{ref} \times (MFI + ((MFN - STOP) \div MFD))$$

The PLL works in spread spectrum mode, and STEP and STOP do not support changing settings when working. If modified at this time, the new setting will be enabled again after spectrum spreading is disabled, or the PLL will be turned off and turned on again.

14.2.5 PLL output divider

The clock of the VCO is divided and output to the clock module to generate the system clock. The frequency divider is a fractional frequency divider with a frequency dividing factor of 0.2 steps.

$$F_{out} = F_{vco} \div (1 + 0.2 \times DIV)$$

DIV supports modifying the working frequency while working.

14.3 PLLCTL register

14.3.1 register description

The register list for PLLCTLv2 is as follows:

PLLCTLV2 base address: 0xF40C0000

Offset	Register Name	Description	Reset Value
0x0000	XTAL	OSC configuration	0x0001FFFF
0x0080	PLL[PLL0][MFI]	PLL0 multiple register	0x00000010
0x0084	PLL[PLL0][MFN]	PLL0 fraction numerator register	0x09896800
0x0088	PLL[PLL0][MFD]	PLL0 fraction demoninator register	0x0E4E1C00
0x008C	PLL[PLL0][SS_STEP]	PLL0 spread spectrum step register	0x00000000
0x0090	PLL[PLL0][SS_STOP]	PLL0 spread spectrum stop register	0x00000000
0x0094	PLL[PLL0][CONFIG]	PLL0 configuration register	0x00000000
0x0098	PLL[PLL0][LOCKTIME]	PLL0 lock time register	0x000009C4
0x009C	PLL[PLL0][STEPTIME]	PLL0 step time register	0x000009C4
0x00A0	PLL[PLL0][ADVANCED]	PLL0 advance configuration register	0x00000000
0x00C0	PLL[PLL0][DIV][DIV0]	PLL0 divider output 0 configuration register	0x00000000
0x00C4	PLL[PLL0][DIV][DIV1]	PLL0 divider output 1 configuration register	0x00000000
0x00C8	PLL[PLL0][DIV][DIV2]	PLL0 divider output 2 configuration register	0x00000000
0x0100	PLL[PLL1][MFI]	PLL0 multiple register	0x00000010
0x0104	PLL[PLL1][MFN]	PLL0 fraction numerator register	0x09896800
0x0108	PLL[PLL1][MFD]	PLL0 fraction demoninator register	0x0E4E1C00
0x010C	PLL[PLL1][SS_STEP]	PLL0 spread spectrum step register	0x00000000
0x0110	PLL[PLL1][SS_STOP]	PLL0 spread spectrum stop register	0x00000000
0x0114	PLL[PLL1][CONFIG]	PLL0 configuration register	0x00000000
0x0118	PLL[PLL1][LOCKTIME]	PLL0 lock time register	0x000009C4
0x011C	PLL[PLL1][STEPTIME]	PLL0 step time register	0x000009C4
0x0120	PLL[PLL1][ADVANCED]	PLL0 advance configuration register	0x00000000
0x0140	PLL[PLL1][DIV][DIV0]	PLL0 divider output 0 configuration register	0x00000000
0x0144	PLL[PLL1][DIV][DIV1]	PLL0 divider output 1 configuration register	0x00000000
0x0148	PLL[PLL1][DIV][DIV2]	PLL0 divider output 2 configuration register	0x00000000
0x0180	PLL[PLL2][MFI]	PLL0 multiple register	0x00000010

Offset	Register Name	Description	Reset Value
0x0184	PLL[PLL2][MFN]	PLL0 fraction numerator register	0x09896800
0x0188	PLL[PLL2][MFD]	PLL0 fraction demoninator register	0x0E4E1C00
0x018C	PLL[PLL2][SS_STEP]	PLL0 spread spectrum step register	0x00000000
0x0190	PLL[PLL2][SS_STOP]	PLL0 spread spectrum stop register	0x00000000
0x0194	PLL[PLL2][CONFIG]	PLL0 configuration register	0x00000000
0x0198	PLL[PLL2][LOCKTIME]	PLL0 lock time register	0x000009C4
0x019C	PLL[PLL2][STEPTIME]	PLL0 step time register	0x000009C4
0x01A0	PLL[PLL2][ADVANCED]	PLL0 advance configuration register	0x00000000
0x01C0	PLL[PLL2][DIV][DIV0]	PLL0 divider output 0 configuration register	0x00000000
0x01C4	PLL[PLL2][DIV][DIV1]	PLL0 divider output 1 configuration register	0x00000000
0x01C8	PLL[PLL2][DIV][DIV2]	PLL0 divider output 2 configuration register	0x00000000

Table 32: PLLCTLV2 Register List

The register details of PLLCTLv2 are as follows:

14.3.2 XTAL (0x0)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BUSY	RSVD	RESPONSE	ENABLE	RSVD								RAMP_TIME																			
RO	N/A	RO	RO	N/A								RW																			
0	x	0	0	x	x	x	x	x	x	x	x	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	

XTAL [31:0]

Bits	Name	Description
31	BUSY	Busy flag 0: Oscillator is working or shutdown 1: Oscillator is changing status
29	RESPONSE	Crystal oscillator status 0: Oscillator is not stable 1: Oscillator is stable for use

Bits	Name	Description
28	ENABLE	Crystal oscillator enable status 0: Oscillator is off 1: Oscillator is on
19-0	RAMP_TIME	Rampup time of XTAL oscillator in cycles of RC24M clock 0: 0 cycle 1: 1 cycle 2: 2 cycle 1048575: 1048575 cycles

XTAL Bits

14.3.3 PLL[MFI] (0x80 + 0x80 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BUSY	RSVD	RESPONSE	ENABLE	RSVD												MFI															
RO	N/A	RO	RO	N/A												RW															
0	x	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	1	0	0	0	0

PLL[MFI] [31:0]

Bits	Name	Description
31	BUSY	Busy flag 0: PLL is stable or shutdown 1: PLL is changing status
29	RESPONSE	PLL status 0: PLL is not stable 1: PLL is stable for use
28	ENABLE	PLL enable status 0: PLL is off 1: PLL is on
6-0	MFI	loop back divider of PLL, support from 13 to 42, $f = f_{ref} * (mfi + mfn/mfd)$ 0-15: invalid 16: divide by 16 17: divide by 17 ... 42: divide by 42 43~:invalid

PLL[MFI] Bits

14.3.4 PLL[MFN] (0x84 + 0x80 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD		MFN																													
N/A		RW																													
x	x	0	0	1	0	0	1	1	0	0	0	1	0	0	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0

PLL[MFN] [31:0]

Bits	Name	Description
29-0	MFN	Numerator of fractional part, $f = f_{ref} * (mfi + mfn/mfd)$. This field supports changing while running.

PLL[MFN] Bits

14.3.5 PLL[MFD] (0x88 + 0x80 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD		MFD																													
N/A		RW																													
x	x	0	0	1	1	1	0	0	1	0	0	1	1	1	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0

PLL[MFD] [31:0]

Bits	Name	Description
29-0	MFD	Demoninator of fraction part, $f = f_{ref} * (mfi + mfn/mfd)$. This field should not be changed during PLL enabled. If changed, change will take effect when PLL re-enabled.

PLL[MFD] Bits

14.3.6 PLL[SS_STEP] (0x8C + 0x80 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD		STEP																													
N/A		RW																													
x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PLL[SS_STEP] [31:0]

Bits	Name	Description
29-0	STEP	Step of spread spectrum modulator. This register should not be changed during PLL and spread spectrum enabled. If changed, new value will take effect when PLL disabled or spread spectrum disabled.

PLL[SS_STEP] Bits

14.3.7 PLL[SS_STOP] (0x90 + 0x80 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD		STOP																													
N/A		RW																													
x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

PLL[SS_STOP] [31:0]

Bits	Name	Description
29-0	STOP	Stop point of spread spectrum modulator This register should not be changed during PLL and spread spectrum enabled. If changed, new value will take effect when PLL disabled or spread spectrum disabled.

PLL[SS_STOP] Bits

14.3.8 PLL[CONFIG] (0x94 + 0x80 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																							SPREAD	RSVD							REFSEL
N/A																							RW	N/A							RW
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	x	x	x	x	x	x	x	0	

PLL[CONFIG] [31:0]

Bits	Name	Description
8	SPREAD	Enable spread spectrum function. This field supports changing during PLL running.

Bits	Name	Description
0	REFSEL	Select reference clock, This field support changing while running, but application must take frequency error and jitter into consideration. And if MFN changed before reference switch, application need make sure time is enough for MFN updating. 0: XTAL24M 1: IRC24M

PLL[CONFIG] Bits

14.3.9 PLL[LOCKTIME] (0x98 + 0x80 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																LOCKTIME															
N/A																RW															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	1	0	0	1	1	1	0	0	0	1	0	0

PLL[LOCKTIME] [31:0]

Bits	Name	Description
15-0	LOCKTIME	Lock time of PLL in 24M clock cycles, typical value is 2500. If MFI changed during PLL startup, PLL lock time may be longer than this setting.

PLL[LOCKTIME] Bits

14.3.10 PLL[STEPTIME] (0x9C + 0x80 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																STEPTIME															
N/A																RW															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	1	0	0	1	1	1	0	0	0	1	0	0

PLL[STEPTIME] [31:0]

Bits	Name	Description
15-0	STEPTIME	Step time for MFI on-the-fly change in 24M clock cycles, typical value is 2500.

PLL[STEPTIME] Bits

14.3.11 PLL[ADVANCED] (0xA0 + 0x80 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD			SLOW	RSVD			DITHER	RSVD																							
N/A			RW	N/A			RW	N/A																							
x	x	x	0	x	x	x	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	

PLL[ADVANCED] [31:0]

Bits	Name	Description
28	SLOW	Use slow lock flow, PLL lock expendite is disabled. This mode might be stabler. And software need config LOCKTIME field accordingly. 0: fast lock enabled, lock time is 100us 1: fast lock disabled, lock time is 400us
24	DITHER	Enable dither function

PLL[ADVANCED] Bits

14.3.12 PLL[DIV] (0xC0 + 0x80 * n + 0x4 * m)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BUSY	RSVD	RESPONSE	ENABLE	RSVD																					DIV						
RO	N/A	RO	RO	N/A																					RW						
0	x	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	

PLL[DIV] [31:0]

Bits	Name	Description
31	BUSY	Busy flag 0: divider is working 1: divider is changing status
29	RESPONSE	Divider response status 0: Divider is not stable 1: Divider is stable for use
28	ENABLE	Divider enable status 0: Divider is off 1: Divider is on

Bits	Name	Description
5-0	DIV	Divider factor, divider factor is $DIV/5 + 1$ 0: divide by 1 1: divide by 1.2 2: divide by 1.4 ... 63: divide by 13.6

PLL[DIV] Bits

15 Low Power Management

This chapter describes the management and application of system-level low power consumption.

15.1 Running Mode RUN

The processor is in running mode when running programs normally. In running mode, the system supports ways to reduce power consumption:

- Properly set the GROUPx register of SYSTL to enable only necessary functional modules, and unused modules will automatically shut down to save power consumption
- Properly reduce the clock frequency of the CPU and functional modules by configuring the frequency and division value of the PLL, as well as the division coefficient of the functional clock in SYSTL
- If the system power domain uses DCDC power supply, you can configure the PCFG module to reduce the DCDC_SOC output voltage appropriately as needed

15.2 Waiting Mode WAIT

If the MODE bit of CPU[CPUx][LP] in the SYSTL register is b00, the CPU will enter wait mode after executing the WFI instruction, and the CPU core clock will be turned off, but the CPU can immediately wake up for processing when an interrupt is received.

15.3 Stop Mode STOP

If the MODE bit of CPU[CPUx][LP] in the SYSTL register is b01, the CPU will enter stop mode after executing the WFI instruction. In stop mode, the SYSTL module will shut down some resource nodes in the system power domain according to configuration. Compared with waiting mode,

allows more resources to be shut down and provides flexible configuration options. Users need to make reasonable configuration in the SYSTL module according to power consumption and wake-up time requirements.

considers the simplest application scenario 1: the system only uses CPU0, only enables the necessary functional modules through the GROUP0 register connection of SYSTL, and modifies the RETENTION[CPU0] register to all 0s. First of all, resource nodes that are not in use are automatically closed in run mode. After the CPU executes WFI and enters the stop mode, the clock of CPU0, the power supply of the CPU0 subsystem, each functional module used by CPU0, the functional clock, other subsystem power supply, PLL and 24M crystal oscillator will be turned off in turn, and only the functions of SYSTL are retained to wait for wake-up.

Application Scenario 2: The system only uses CPU0. If the register status of CPU0 is required not to be lost after entering stop mode, that is, the CPU0 subsystem will not be powered down or reset, and the 24M crystal oscillator and PLL are required not to be turned off to achieve a faster wake-up speed, you need to set the RETENTION[CPU0] register of SYSTL to set the corresponding positions of the CPU0 register, 24M crystal oscillator and PLL to 1.

Application Scenario 3: The system only uses CPU0. If a certain functional module such as GPTMR0 is required to keep running after CPU0 enters stop mode, and this module is not within the management range of the RETENTION register, you need to set the RESOURCE[GPTMR0] register of SYSTL, and set the MODE bit to 1 to keep it turned on.

Note that the default value of the RETENTION[CPU0] register reserves power to the CPU0 subsystem and

system power domain.

Interrupts can wake up the system from stopped mode. These interrupts can come from functional modules in the system's power domain that are still working, or from the power management domain. Make sure that the interrupt source you want to wake up is enabled in the CPU[CPUx][WAKEUP_ENABLE][ENABLEx] register of SYSCTL.

In summary, the steps that may be involved in configuring the stop mode are as follows:

- Configure the MODE bit of the SYSCTL register CPU[CPUx][LP] to b01
- Configure the SYSCTL register RETENTION[x] to keep some resource nodes open
- Configure the SYSCTL register RESOURCE[x] to keep more resource nodes open
- Configure SYSCTL register CPU[CPUx][WAKEUP_ENABLE][ENABLEx] to specify the wake-up interrupt source
- Configure the module used to generate wake-up interrupts, set interrupt triggering conditions, and enable interrupts
- Implement WFI command

15.4 Sleep Mode STANDBY

In sleep mode, the entire system power domain is no longer running and may be in a reset state or a power down state. If there are no resource nodes in the RETENTION register in the SYSCTL configuration that need to be kept running, and there are no resource nodes in the RESOURCE register that are forcibly turned on, then when all CPUs enter the stop mode, the entire system power domain will no longer have running requirements. After the SYSCTL module shuts down all resource nodes, it will issue a request to the power management domain to turn off the system power domain. After the request occurs, the system power domain will be reset first, and then the PCFG module will perform the operation according to the setting of the POWER_TRAP register:

If DCDC is used to supply power to the system's power domain, the DCDC will switch state according to [Table 23](#): normal operation, shutdown, or enter a low power mode, where the DCDC can provide a lower output voltage according to configuration in the low power mode.

After entering sleep mode, the system can only be awakened by interrupts from modules in the power management domain. See [Subsection 12.2.4](#) for the wake-up source.

In addition, unused modules in the power management domain can turn off their clocks by setting the SCG_CTRL register of PCFG, thereby further reducing system power consumption in sleep mode. See [Subsection 12.2.3](#) for details.

In summary, the steps that may be involved in configuring the sleep mode are as follows:

- Configure the MODE bit of the SYSCTL register CPU[CPUx][LP] to b01
- Configure SYSCTL register RETENTION[x] register to all 0s
- Configure that the MODE bit in all SYSCTL registers in the RESOURCE[x] register is not 1
- Configure the TRAP bit and RETENTION bit in the POWER_TRAP register of the PCFG module to set the power domain to hold, turn off, or decrease the voltage
- Configure the PCFG module WAKE_MASK register to enable wake-up sources
- Configure interrupt triggering conditions for wake-up source modules in the power management domain and enable interrupts

- Implement WFI command

15.5 Shutdown mode SHUTDOWN

In shutdown mode, only PDGO in the power management domain can work normally, and parts of the power management domain other than PDGO and the system power domain are in a reset or power-down state.

enters shutdown mode by software writing registers:

- writes a non-zero countdown value to the countdown shutdown register DGO_TURNOFF. After the countdown ends, the system enters shutdown mode

The shutdown mode of this product does not require special treatment for the system-level power supply. Power supply pins such as VPMC and DCDC_IN can remain in the power supply state.

The DCDC in the power management domain is turned off, so the system power domain will lose power if it is powered by the built-in DCDC. LDOPMC in the

power management domain is turned off, so all functions in the power management domain except PDGO cannot be used, and only the RESETN and WAKEUP pins are reserved for global reset or wake-up.

In shutdown mode, the system can only wake up from the PDGO module. See [Subsection 11.4.2](#) for details on how to exit shutdown mode.

16 System Memory Mapping

This chapter describes the system memory mapping of this product.

16.1 System Memory Map

The system memory mapping table of this product is as follows:

Start Address	End Address	Size	Name	Description
0x00000000	0x0001FFFF	128 KBytes	CPU0_ILM	N/A
0x00200000	0x0021FFFF	128 KBytes	CPU0_DLM	N/A
0x00300000	0x00303FFF	16 KBytes	FGPIO	N/A
0x00400000	0x00403FFF	16 KBytes	TFA	N/A
0x011C0000	0x011DFFFF	128 KBytes	CPU0_ILM_SLV_ALS	N/A
0x011E0000	0x011FFFFFFF	128 KBytes	CPU0_DLM_SLV_ALS	N/A
0x01200000	0x0123FFFF	256 KBytes	XRAM0	N/A
0x20000000	0x2001FFFF	128 KBytes	ROM	N/A
0x30000000	0x300FFFFFFF	1 MBytes	DM	N/A
0x80000000	0x8FFFFFFF	256 MBytes	XPI0	N/A
0xB0000000	0xBFFFFFFF	256 MBytes	HYBRID_XPI(256KB)	N/A
0xE4000000	0xE43FFFFFFF	4 MBytes	PLIC	N/A
0xE6000000	0xE60FFFFFFF	1 MBytes	MCHTMR	N/A
0xE6400000	0xE67FFFFFFF	4 MBytes	PLICSW	N/A
0xF0000000	0xF0003FFF	16 KBytes	GPTMR0	N/A
0xF0004000	0xF0007FFF	16 KBytes	GPTMR1	N/A
0xF0008000	0xF000BFFF	16 KBytes	GPTMR2	N/A
0xF000C000	0xF000FFFF	16 KBytes	GPTMR3	N/A
0xF0030000	0xF0033FFF	16 KBytes	OWR0	N/A
0xF0034000	0xF0037FFF	16 KBytes	OWR1	N/A
0xF0038000	0xF003BFFF	16 KBytes	EU10	N/A
0xF003C000	0xF003FFFF	16 KBytes	EU11	N/A
0xF0040000	0xF0043FFF	16 KBytes	UART0	N/A
0xF0044000	0xF0047FFF	16 KBytes	UART1	N/A
0xF0048000	0xF004BFFF	16 KBytes	UART2	N/A
0xF004C000	0xF004FFFF	16 KBytes	UART3	N/A
0xF0050000	0xF0053FFF	16 KBytes	UART4	N/A
0xF0054000	0xF0057FFF	16 KBytes	UART5	N/A
0xF0058000	0xF005BFFF	16 KBytes	UART6	N/A
0xF005C000	0xF005FFFF	16 KBytes	UART7	N/A
0xF0060000	0xF0063FFF	16 KBytes	I2C0	N/A
0xF0064000	0xF0067FFF	16 KBytes	I2C1	N/A
0xF0068000	0xF006BFFF	16 KBytes	I2C2	N/A

Start Address	End Address	Size	Name	Description
0xF006C000	0xF006FFFF	16 KBytes	I2C3	N/A
0xF0070000	0xF0073FFF	16 KBytes	SPI0	N/A
0xF0074000	0xF0077FFF	16 KBytes	SPI1	N/A
0xF0078000	0xF007BFFF	16 KBytes	SPI2	N/A
0xF007C000	0xF007FFFF	16 KBytes	SPI3	N/A
0xF0080000	0xF0083FFF	16 KBytes	CRC	N/A
0xF0090000	0xF0093FFF	16 KBytes	TSNS	N/A
0xF00A0000	0xF00A3FFF	16 KBytes	MBX0A	N/A
0xF00A4000	0xF00A7FFF	16 KBytes	MBX0B	N/A
0xF00B0000	0xF00B3FFF	16 KBytes	EWDG0	N/A
0xF00B4000	0xF00B7FFF	16 KBytes	EWDG1	N/A
0xF00C4000	0xF00C7FFF	16 KBytes	DMAMUX	N/A
0xF00C8000	0xF00CBFFF	16 KBytes	HDMA	N/A
0xF00CC000	0xF00CFFFF	16 KBytes	PPI	N/A
0xF00D0000	0xF00D3FFF	16 KBytes	GPIO0	N/A
0xF00D8000	0xF00DBFFF	16 KBytes	GPIOM	N/A
0xF00DC000	0xF00DFFFF	16 KBytes	LOBS	N/A
0xF0100000	0xF0103FFF	16 KBytes	ADC0	N/A
0xF0104000	0xF0107FFF	16 KBytes	ADC1	N/A
0xF0130000	0xF0133FFF	16 KBytes	ACMP0	N/A
0xF0200000	0xF0207FFF	32 KBytes	HRAM	N/A
0xF0300000	0xF0303FFF	16 KBytes	MCAN0	N/A
0xF0304000	0xF0307FFF	16 KBytes	MCAN1	N/A
0xF0308000	0xF030BFFF	16 KBytes	MCAN2	N/A
0xF030C000	0xF030FFFF	16 KBytes	MCAN3	N/A
0xF037C000	0xF037FFFF	16 KBytes	PTPC	N/A
0xF0400000	0xF0403FFF	16 KBytes	QEIO	N/A
0xF0404000	0xF0407FFF	16 KBytes	QEIO	N/A
0xF0410000	0xF0413FFF	16 KBytes	QEO0	N/A
0xF0414000	0xF0417FFF	16 KBytes	QEO1	N/A
0xF0420000	0xF0423FFF	16 KBytes	PWM0	N/A
0xF0424000	0xF0427FFF	16 KBytes	PWM1	N/A
0xF0450000	0xF0453FFF	16 KBytes	SDM0	N/A
0xF0460000	0xF0463FFF	16 KBytes	PLB	N/A
0xF0464000	0xF0467FFF	16 KBytes	SYNT	N/A
0xF047C000	0xF047FFFF	16 KBytes	TRGM0	N/A
0xF1400000	0xF1403FFF	16 KBytes	ENET0	N/A
0xF1410000	0xF1413FFF	16 KBytes	NTMR0	N/A
0xF1420000	0xF1423FFF	16 KBytes	USB0	N/A
0xF1700000	0xF171FFFF	128 KBytes	ESC	N/A

Start Address	End Address	Size	Name	Description
0xF3000000	0xF3003FFF	16 KBytes	XPI0	N/A
0xF3100000	0xF3103FFF	16 KBytes	XDMA	N/A
0xF3158000	0xF315BFFF	16 KBytes	OTP	N/A
0xF4000000	0xF403FFFF	256 KBytes	SYSCTL	N/A
0xF4040000	0xF407FFFF	256 KBytes	IOC	N/A
0xF40C0000	0xF40FFFFFFF	256 KBytes	PLLCTLV2	N/A
0xF4100000	0xF4103FFF	16 KBytes	PPOR	N/A
0xF4104000	0xF4107FFF	16 KBytes	PCFG	N/A
0xF4118000	0xF411BFFF	16 KBytes	PIOC	N/A
0xF411C000	0xF411FFFF	16 KBytes	PGPIO	N/A
0xF4120000	0xF4123FFF	16 KBytes	PTMR	N/A
0xF4124000	0xF4127FFF	16 KBytes	PUART	N/A
0xF4128000	0xF412BFFF	16 KBytes	PEWDG	N/A
0xF4134000	0xF4137FFF	16 KBytes	PDGO	N/A
0xF4138000	0xF413BFFF	16 KBytes	PGPR0	N/A
0xF413C000	0xF413FFFF	16 KBytes	PGPR1	N/A
0xF8000000	0xFFFFFFFF	128 MBytes	PPI	N/A

Table 33: Address space allocation

17 OTP Mapping Table

The OTP mapping table of this product is as follows:

Offset	Name	Position	Function	Type
0	HARD_LOCK	[0:31]	Fuse Value Lock	Security
1	LIFECYCLE_B	[28:31]	Life Cycle	Security
1	TCU_DISABLE	[19:19]	Disable Test	Security
1	DEBUG_DISABLE	[17:17]	Disable CPU Debug	Security
1	JTAG_DISABLE	[16:16]	Disable JTAG Port	Security
1	PUK_REVOKE	[8:15]	Public Key Revocation	Security
1	LIFECYCLE_A	[0:3]	Life Cycle	Security
2	MONO_EPOCH	[16:31]	Monotonic Counter High-Order Value	Security
3	SW_VER	[0:31]	Software Version	Security
8	DIE_TRACE	[0:127]	Chip Identification Data	Identification
12	DEBUG_KEY	[0:127]	Debug Key	Key
21	TSNS_BASE	[0:11]	Temperature Sensor Base Value	General
21	TSNS_SLOPE	[12:23]	Temperature Sensor Slope	General
21	TSNS_BASE_EN	[24:24]	Temperature Sensor Base Enable	General
21	TSNS_SLOPE_EN	[25:25]	Temperature Sensor Slope Enable	General
64	CHIP_ID	[0:31]	Chip Model Code	General
65	MAC0	[0:47]	ETH0 MAC Address	General
66	MAC1	[16:63]	ETH1 MAC Address	General
68	USB_PID	[0:15]	USB Product ID (PID)	General
68	USB_VID	[16:31]	USB Vendor ID (VID)	General
80	PUBLIC_KEY_HASH	[0:255]	Public Key Hash	Identification
88	UUID	[0:127]	Universally Unique Identifier (UUID)	Identification
96	EXIP0_KEY	[0:127]	EXIP Key 0	Key
100	EXIP1_KEY	[0:127]	EXIP Key 1	Key
104	USER_KEY	[0:255]	User Key	Key
112	MASTER_KEY	[0:255]	Symmetric Algorithm Root Key	Key

Table 34: OTP List

On this product, users can program OTP Word 69~79 by themselves to store application data.

18 BootROM

This chapter details the features and applications related to BootROM.

18.1 BootROM Overview

BootROM is the first program executed after the chip is powered on. It supports the following functions:

- Booting from [Serial NOR FLASH](#)
 - Supports executing code in-place on FLASH or copying to internal RAM for execution
 - Supports booting from XPI0 accelerated address space
 - Supports in-place decryption and execution (EXIP) based on AES-CTR mode
- [In-System Programming \(ISP\)](#)
 - Supports UART0 and USB0 (USB-HID protocol)
 - Supports programming Serial NOR FLASH connected to XPI0
 - Supports programming on-chip OTP
 - Boots from UART0 or USB0, supports streaming download of code to system RAM for execution
- [Secure Boot](#)
 - Supports secure boot based on ECDSA-P256 + SHA256
 - Supports secure boot based on SM2+SM3
- [Low Power Wake-up](#)
 - Supports fast wake-up via BootROM from CPU low power shutdown mode
- [Debug Port Permission Management](#)
- [Rich ROM API](#)

18.2 Boot Flow

This chapter describes the boot flow related to BootROM.

BootROM supports the following boot modes:

- Main boot mode (XPI NOR boot mode).
- In-System Programming mode (In-System-Programming)

18.2.1 Boot Flow Diagram

This section shows the boot flow diagram of BootROM.

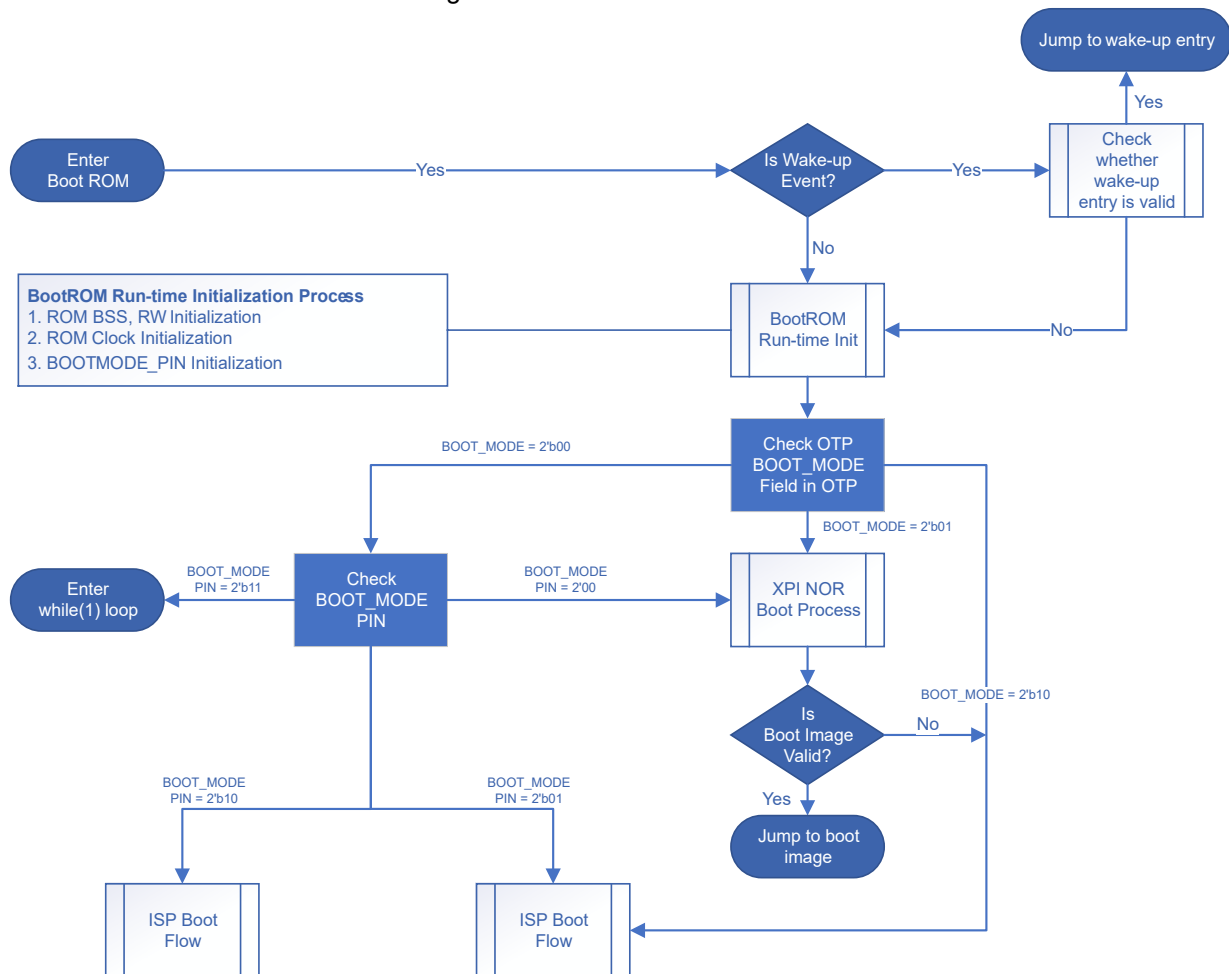


Figure 10: BootROM Boot Flow Diagram

18.2.2 XPI NOR Boot

18.2.2.1 Supported Serial NOR Devices

Boot ROM supports booting from mainstream Serial NOR FLASH, including but not limited to:

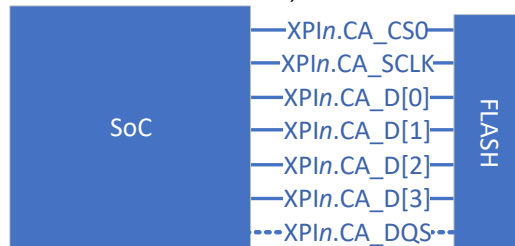
- QSPI NOR FLASH (Macronix, Winbond, GigaDevice, ISSI, etc.)
- HyperFLASH (Cypress, ISSI and other manufacturers)
- OctaBus FLASH (Macronix)
- Xccela FLASH (Micron, GigaDevice, ISSI and other manufacturers)

18.2.2.2 FLASH Connection Methods

BootROM connects with 4-wire or 8-wire FLASH through the following three typical connection methods.

QSPI FLASH Connection via CA Port

SoC connects with 4-wire FLASH through the CA port of XPI. Most 4-wire FLASH on the market do not provide DQS pins. In this case, XPIIn.CA_DQS should be left floating, and this pin cannot be used for other purposes (unless otherwise specified in the [XPI Information](#) section).

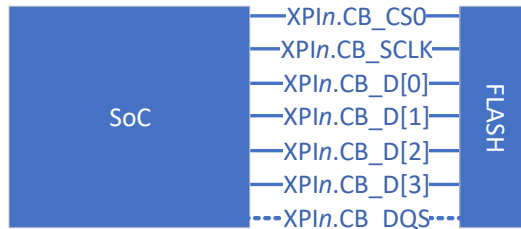


Scenario #1

Figure 11: XPI QSPI NOR FLASH CA Port Connection

QSPI FLASH Connection via CB Port

SoC connects with 4-wire FLASH through the CB port of XPI. Most 4-wire FLASH on the market do not provide DQS pins. In this case, XPIIn.CB_DQS should be left floating, and this pin cannot be used for other purposes (unless otherwise specified in the [XPI Information](#) section).



Scenario #2

Figure 12: XPI QSPI NOR FLASH CB Port Connection

Octal FLASH Connection

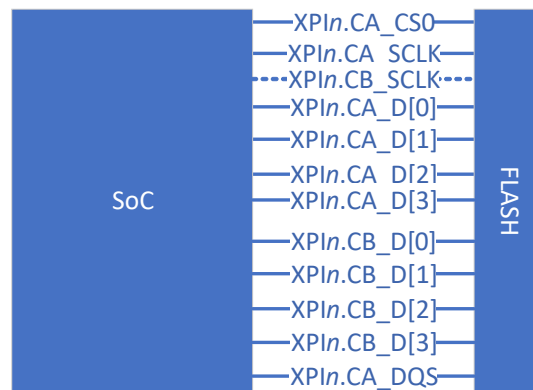
SoC connects with 8-wire FLASH through the CA port of XPI and the data port of CB, where the CB data port is connected to the high 4 bits of FLASH. If the 8-wire FLASH requires differential clock (such as HyperFLASH 1.8V), the clock line of the CB port can be used as a differential clock pin.

Specification of FLASH Connection Method

BootROM determines the corresponding FLASH connection based on the following OTP fields:

- Select default XPI pin grouping based on XPI_PIN_GROUP
- Determine XPI default connection port based on XPI_PORT_SEL

Detailed XPI pin assignment can be found in the [XPI Information](#) section.



Scenario #3

Figure 13: XPI Octal NOR FLASH Connection

Specification of FLASH Configuration Information

BootROM determines the corresponding FLASH configuration information based on the following OTP fields:

- Determine the default source of FLASH configuration information based on XPI_NOR_CFG_SRC: Serial NOR FLASH or OTP
- Determine the default FLASH access command after power-on based on XPI_DEFAULT_READ
- Determine whether to enable encrypted execute-in-place function based on ENCRYPT_XIP

When FLASH configuration information comes from OTP:

- Determine FLASH read operation clock frequency based on XPI_FREQ_OPTION
- Determine FLASH type based on PROBE_TYPE
- Determine XPI IO drive strength based on XPI_DRIVE_STRENGTH
- Determine dummy cycle value during read based on DUMMY_CYCLE
- Determine FLASH IO voltage based on XPI_IO_VOLTAGE

When FLASH configuration information comes from FLASH, determine the FLASH read command to use based on XPI_DEFAULT_READ, and read FLASH configuration information from FLASH address 0x400.

Note: Detailed description of FLASH configuration information can be found in [XPI NOR Configuration Options](#)

18.2.2.3 XPI NOR Boot Flow

When the user selects booting from XPI NOR, the device will execute the XPI NOR boot according to the flow shown in Figure (14) after power-on.

18.2.2.4 XPI NOR Boot Image Layout

The boot image layout of XPI NOR in FLASH is shown in Figure (15):

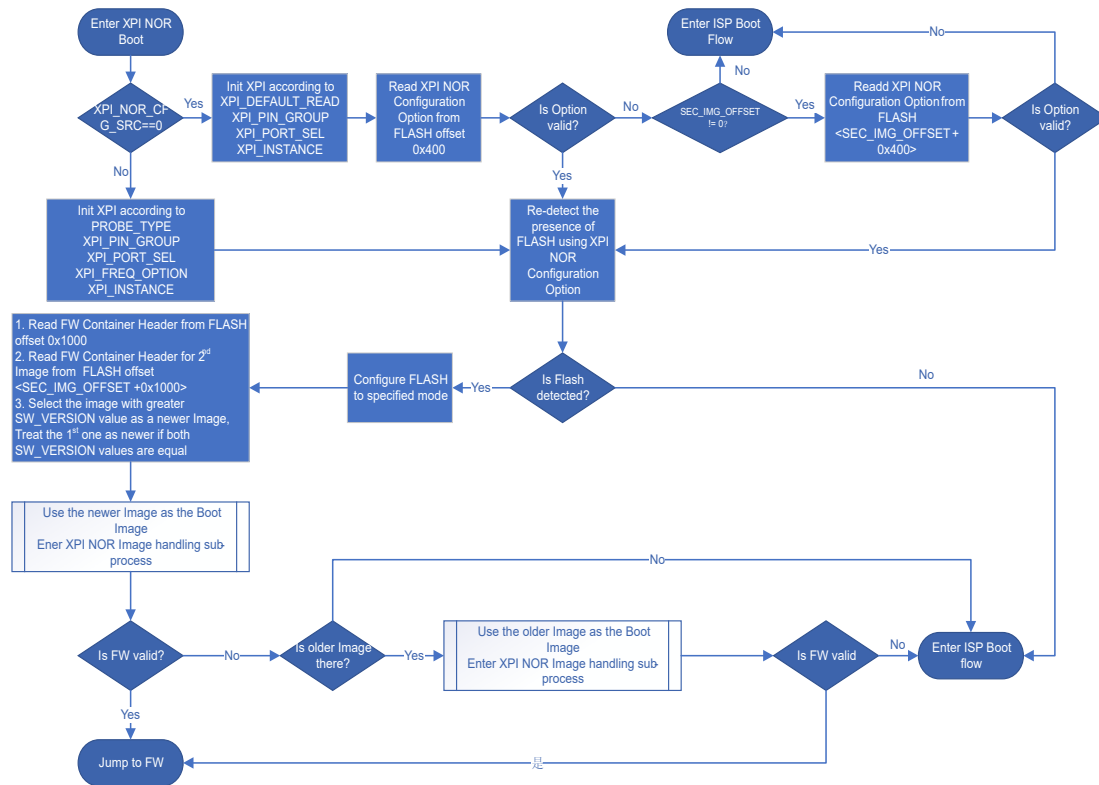


Figure 14: XPI NOR Boot Flow

18.2.2.5 Execute-In-Place with Decryption

BootROM supports execute-in-place with decryption through XPI NOR. When the ENCRYPT_XIP field in OTP is set to 1, BootROM will forcibly enable EXIP during XPI NOR boot and attempt to decrypt the EXIP Blob using EXIP0_KEK (when XPI_INSTANCE value is 0)/EXIP1_KEK (when XPI_INSTANCE value is 1). After the EXIP Blob is successfully decrypted, ROM will configure the corresponding decryption information (decryption area, initial vector, etc.) according to the information in the EXIP BLOB. After EXIP configuration is completed, ROM can perform execute-in-place decryption operations.

Note: The header of the boot image cannot be encrypted, otherwise BootROM cannot boot this image

Execute-in-place decryption is divided into two steps:

1. Decrypt the encrypted EXIP BLOB using $EXIP\_KEK_n$ in OTP. The EXIP BLOB is encrypted using the [RFC3394](#) standard.
2. Configure the EXIP module using the information in the EXIP BLOB to prepare the execute-in-place decryption environment.

18.2.3 In-System Programming (In-System-Programming)

ROM in ISP mode supports the following interfaces:

- Serial port (UART)
- USB (USB-HID)

ROM in ISP mode supports the following functions:

- Query runtime information (query-rte)

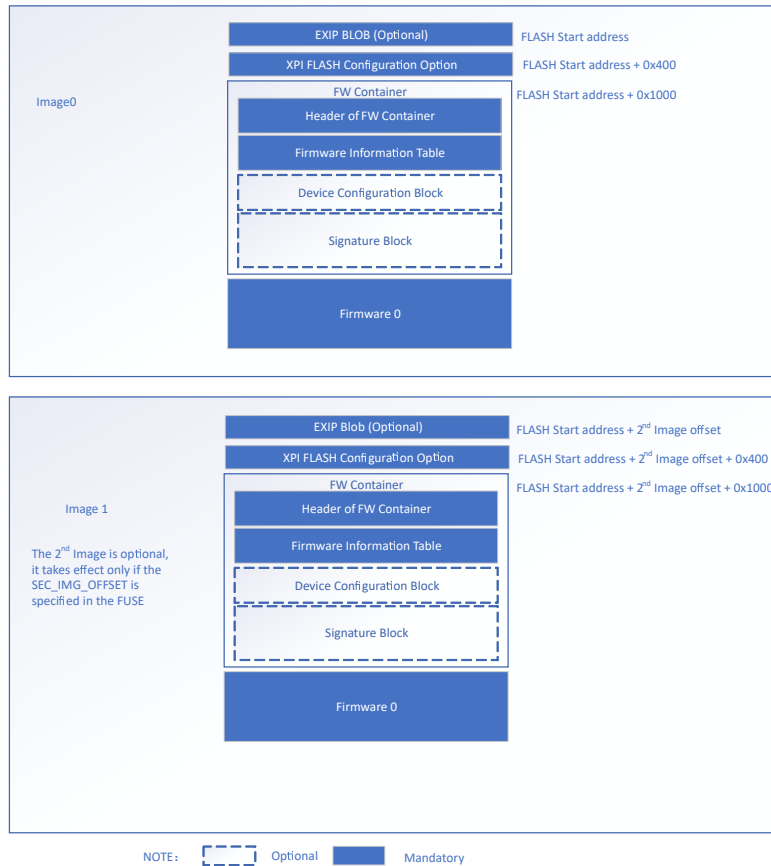


Figure 15: XPI NOR Boot Image Layout

- Configure runtime information (configure-rte)
- Configure storage (configure-memory)
- Write to storage device (write-memory)
- Read from storage device (read-memory)
- Load image (load-image)
- Erase storage device (erase)
- Reset device (reset)

18.2.3.1 Command Protocol

BootROM supports the following commands:

- Query runtime environment (query-rte)
- Configure runtime environment (configure-rte)
- Configure storage device (configure-memory)
- Write to storage device (write-memory)
- Read from storage device (read-memory)
- Load image (load-image)
- Erase storage device (erase)
- Reset SoC (reset)

Communication Process

Under ROM's communication protocol, each command contains 4 phases as shown in the figure below.

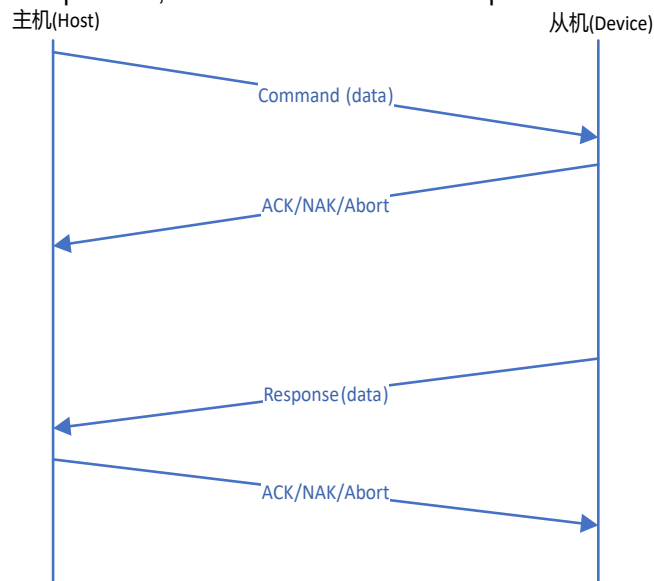


Figure 16: Serial Communication Protocol

1. Phase 1: Host command and data transmission phase. In this phase, if the command only needs to transmit simple parameters, the command data transmission only contains one data packet (such as reset/erase); if the command contains data transmission, the data transmission contains a mixed packet (command + data) and several pure data packets.
2. Phase 2: Slave sends ACK/NAK/Abort phase. For each data packet in the first phase, the slave needs to provide corresponding confirmation. When data processing is normal, reply ACK; when data check error occurs, reply NAK; when data processing error occurs, reply Abort.
3. Phase 3: Slave sends response phase. When the first phase data transmission is completed or the Device replies Abort, the communication enters the third phase. In this phase, if the command only requires the slave to reply with a simple status, the data packet only has one response packet (such as reset/erase); if the slave needs to reply with data, the slave will first reply with a command response frame, and then reply with the remaining data to be sent. After each frame is sent, the slave needs to wait for the host's confirmation frame. When the host sends Abort, the communication ends.
4. Phase 4: Host sends confirmation for slave response. When the host receives the slave's response packet, it needs to reply with ACK/NAK/Abort. When the data packet is normal, reply ACK; when the data packet check error occurs, reply NAK; when the slave sends unexpected data, reply Abort.

Command Data Structure

Each command uniformly contains a 4-byte command header, with detailed definitions as shown in the following table:

Offset (Bytes)	Field	Description

Offset (Bytes)	Field	Description
0	Command Identifier	1 - Query runtime environment (query-rte) 2 - Configure runtime environment (configure-rte) 3 - Configure storage (configure-memory) 4 - Write to storage (write-memory) 5 - Read from storage (read-memory) 6 - Load image (load-image) 7 - Erase (erase) 8 - Reset (reset) 9 - Generate firmware Blob (gen-fwblob)
1	Number of command parameters	-
2	Command type	0 - Command + data 1 - Data only 2 - Response only
3	Reserved	-

Table 35: Command Data Structure

Query Runtime Environment (query-rte)

This command can be used to query the following status:

- ROM parameters
- Active communication interface parameters
- Last boot status
- Memory properties

The detailed command data structure is shown in the following table:

Offset (Bytes)	Width (Bytes)	Field	Description
0	1	Command Identifier	1 - query-rte
1	1	Number of command parameters	1 / 2
2	1	Command type	0 - Command + data
3	1	Reserved	-
4	4	Runtime environment identifier	0 - ROM parameters 1 - Current active peripheral information 3 - Last boot status 4 - Storage properties

Table 36: Query Runtime Environment (query-rte) Command Data Structure

When ROM receives this command, if the parameters are valid, ROM will reply with the corresponding response packet. The parameter part of the response packet varies according to different commands.

ROM Parameters Response Packet

Offset (Bytes)	Width (Bytes)	Field	Description			
0	1	Command Identifier	1 - query-rte			
1	1	Number of command parameters	5			
2	1	Command type	2 - Response data only			
3	1	Reserved	-			
4	4	Command status	0 - Success			
8	4	ROM version	Bit[31:24]	ASCII: V(0x56)		
			Bit[23:16]	Major version: 0x01		
			Bit[15:8]	Minor version: 0x00		
			Bit[7:0]	Patch: 0x00		
12	4	SoC version	Same definition as ROM version number			
16	4	Miscellaneous	Bit[7:0] - Life cycle			
			Bit[31:8] - Reserved			
20	4	ROM function information	Bit[31:27]	Reserved		
			Bit[26:26]	Supports SM4_CCM		
			Bit[25:25]	Supports AES_CCM (256-bit key)		
			Bit[24:24]	Supports AES_CCM (128-bit key)		
			Bit[23:20]	Reserved		
			Bit[19:19]	Supports SM2		
			Bit[18:18]	Supports ECDSA P521		
			Bit[17:17]	Supports ECDSA P384		
			Bit[16:16]	Supports ECDSA P256		
			Bit[15:12]	Reserved		
			Bit[11:11]	Supports device configuration block		
			Bit[10:10]	Supports command container		
			Bit[9:9]	Supports encrypted boot		
			Bit[8:8]	Supports secure boot		
			Bit[7:4]	Reserved		
			Bit[3:3]	Supports In-System Programming (ISP)		
			Bit[2:2]	Supports serial boot		
			Bit[1:1]	Supports recovery boot		
			Bit[0:0]	Supports main boot		

Offset (Bytes)	Width (Bytes)	Field	Description

Table 37: ROM Parameters Response Packet

Active Peripheral Information Packet

Offset (Bytes)	Width (Bytes)	Field	Description
0	1	Command Identifier	1 - query-rte
1	1	Number of command parameters	If parameters are invalid, this value is 0, otherwise it is "word" bytes + 1
2	1	Command type	2 - Response data only
3	1	Reserved	-
4	4	Command status	0 - Success 2 - Invalid parameter
8	4	General information	Bit[7:0] Number of words (in 32-bit units) Bit[31:8] - Reserved
12	4	Peripheral mask	0x01 - UART 0x10 - USB-HID
16	4	Rate	For UART, this field represents baud rate; for USB-HID, this field is meaningless
20	8	Reserved	-
28	4	Maximum packet length	Maximum data packet length (excluding packet header and CRC check bits)

Table 38: Active Peripheral Information Packet

Last ROM Boot Status Packet

Offset (Bytes)	Width (Bytes)	Field	Description
0	1	Command Identifier	1 - query-rte
1	1	Number of command parameters	If parameters are invalid, this value is 0, otherwise it is "word" bytes + 1
2	1	Command type	2 - Response data only

Offset (Bytes)	Width (Bytes)	Field	Description	
3	1	Reserved	-	
4	4	Command status	0 - Success	
8	4	Boot parameters	Boot parameters	
			Bit[31:16]	Reserved
			Bit[15:8]	Number of words for boot status (in 32-bit units)
			Bit[7:0]	Boot mode

Table 39: Last ROM Boot Status

Storage Device Properties Packet

Offset (Bytes)	Width (Bytes)	Field	Description
0	1	Command Identifier	1 - query-rte
1	1	Number of command parameters	If parameters are invalid, this value is 0, otherwise it is "word" bytes + 1
2	1	Command type	2 - Response data only
3	1	Reserved	-
4	4	Command status	0 - Success
8	4	General information	Bit[7:0] Number of words for storage device properties (in 32-bit units) Bit[15:8] Storage size unit: 0 - 1 byte, 1 - 2 bytes, 2 - 4 bytes Others - Reserved
12	8	Storage device identifier	TBD
16	4	Start address	Storage device start address
20	4	End address	Storage device end address
24	4	Storage device capacity	Storage device capacity
28	4	Page size (optional)	Only valid for FLASH devices

Offset (Bytes)	Width (Bytes)	Field	Description
32	4	Sector size (optional)	Only valid for FLASH devices
32	4	Block size (optional)	Only valid for FLASH devices

Table 40: Storage Device Properties Response Packet

Configure Runtime Environment (configure-rte)

This command can configure the parameters of active peripherals. Currently only the baud rate of UART is supported to be configured. The detailed command data structure is as follows:

Offset (Bytes)	Width (Bytes)	Field	Description
0	1	Command Identifier	2 - configure-rte
1	1	Number of command parameters	4
2	1	Command type	0 - Command + data
3	1	Reserved	-
4	4	Runtime environment identifier	1 - Active peripheral information
8	4	General information	Bit[7:0] Number of parameter words (in 32-bit units) Others - Reserved
12	4	Peripheral mask	1 - UART
16	4	Rate	UART baud rate

Table 41: Configure Runtime Environment (configure-rte)

The response frame data structure for this command is as follows:

Offset (Bytes)	Width (Bytes)	Field	Description
0	1	Command Identifier	2 - configure-rte
1	1	Number of command parameters	1

Offset (Bytes)	Width (Bytes)	Field	Description
2	1	Command type	2 - Response data only
3	1	Reserved	-
4	4	Command status	0 - Success 2 - Invalid parameter

Table 42: Configure Runtime Environment Response Packet

Configure Storage (configure-memory)

This command is used to enable support for external storage devices by specifying the storage device configuration information stored at a specified address, such as enabling support for Serial NOR FLASH. The detailed data structure of this command is as follows:

Offset (Bytes)	Width (Bytes)	Field	Description
0	1	Command Identifier	3 - configure-memory
1	1	Number of command parameters	2
2	1	Command type	0 - Command + data
3	1	Reserved	-
4	4	Storage device identifier	0x10000 - NOR FLASH mounted on XPI0
8	4	Storage configuration block address	Address of the storage configuration block (only supports reading from on-chip RAM)

Table 43: Configure Storage (configure-memory)

The data structure of the response data frame for this command is as follows:

Offset (Bytes)	Width (Bytes)	Field	Description
0	1	Command Identifier	3 - configure-memory
1	1	Number of command parameters	1

Offset (Bytes)	Width (Bytes)	Field	Description
2	1	Command type	2 - Response data only
3	1	Reserved	-
4	4	Command status	0 - Success 2 - Invalid parameter

Table 44: configure-memory Response Frame Data Structure

XPI NOR Configuration Options

Offset (Bytes)	Field	Description
0x00	Header	Bit[31:12] - tag: Must be 0xfcf90 Bit[11:4] - Reserved Bit[3:0] - words: Number of configuration option words (excluding Header itself)

Offset (Bytes)	Field	Description
0x04	Option0	Bit[31:28] - Probe type 0 - SFDP SDR, 1 - SFDP DDR 2 - 1-4-4 Read 0xEB, 3 - 1-2-2 Read 0xBB 4 - HyperBus 1V8, 5 - HyperBus 3V0 6 - OctaBus DDR, 8 - Xccela DDR 10 - EcoXiP DDR Bit[27:24] - Number of data pins for sending commands after power-on reset 0 - SPI, 1 - DPI, 2 - QPI, 3 - OPI Bit[23:20] - Number of data pins for sending commands after FLASH configuration 0 - SPI, 1 - DPI, 2 - QPI, 3 - OPI Bit[19:16] - 1-4-4 mode enable sequence 0 - Not required or automatic 1 - QE bit is at bit 6 in Status Register1 2 - QE bit is at bit1 in Status Register2 3 - QE bit is at bit7 in Status Register2 4 - QE bit is at bit1 in Status Register2 and should be programmed by 0x31 Bit[15:8] - dummy cycles 0 - Auto detect/default value Others - User specified value, for DTR read, this value should be <i>dummycycle</i>×2 as specified in the actual datasheet Bit[7:4] - Miscellaneous 0 - None 1 - SPI mode 2 - Internal loopback mode 3 - Force external DQS Bit[3:0] - Frequency option 0 - Do not change existing configuration Others - Corresponding value for SoC

Offset (Bytes)	Field	Description
0x08	Option1	This field is valid when the "words" field value is greater than 1 Bit[31:20] - reserved Bit[19:16] - IO voltage 0 - IO is 3V, 1 - IO is 1.8V Bit[15:12] - Pin group selection 0 - First group, 1 - Second group Bit[11:8] - Connection method selection 0 - PORTA_CS0, 1 - PORB_CS0, 2 - PORTA_CS0 + PORTB_CS0, 3 - PORTA_CS0+PORTA_CS1, 4 - PORTB_CS0+PORTB_CS1 Bit[7:0] - IO drive strength 0 - Default value (maximum drive strength) Others - See details (TBD)

Table 45: XPI NOR Configuration Options (XPI NOR Configuration Option)

Write to Storage Device (write-memory)

This command is used to write data to on-chip SRAM, on-chip OTP, and external XPI NOR. This command can consist of a single data packet or multiple data packets. The data structure of the first packet is defined as follows:

Offset (Bytes)	Width (Bytes)	Field	Description
0	1	Command Identifier	4 - write-memory
1	1	Number of command parameters	3
2	1	Command type	0 - Command + data
3	1	Reserved	-
4	4	Address to write to	Device address to write to
8	4	Length	Length of data to write
12	4	Memory device identifier	0 - On-chip RAM 0x10000 - NOR FLASH mounted on XPI0 0x10001 - NOR FLASH mounted on XPI1 0x20000 - OTP

Offset (Bytes)	Width (Bytes)	Field	Description
16	Variable length	Data	Data to be written to the storage device

Table 46: Configure Storage (write-memory) - Command + Data

The data structure of the second packet to the last packet is as follows:

Offset (Bytes)	Width (Bytes)	Field	Description
0	1	Command Identifier	4 - write-memory
1	1	Number of command parameters	0
2	1	Command type	1 - Data only
3	1	Reserved	-
4	Variable length	Data	Data to be written to the storage device

Table 47: Configure Storage (write-memory) - Data Only

When data can be sent in a single packet, only the first type of data packet is needed. When data needs to be sent in multiple packets, the second type of data packet is required. The length of the Data part in the data packet is obtained from the data packet length received through various communication interfaces.

If an error occurs during the execution of write-memory, the slave will abort the transmission and notify the host of the error code in the response packet.

When write-memory is successfully executed, it reports the success code to the host in the response packet.

The response packet data structure for write-memory is defined as follows:

Offset (Bytes)	Width (Bytes)	Field	Description
0	1	Command Identifier	4 - configure-memory
1	1	Number of command parameters	1
2	1	Command type	2 - Response data only
3	1	Reserved	-
4	4	Command status	0 - Success Others - Error code

Offset (Bytes)	Width (Bytes)	Field	Description

Table 48: write-memory Response Frame Data Structure

Read from Storage (read-memory)

This command is used to read data from on-chip SRAM, on-chip OTP, and external XPI NOR. This command consists of one command packet and multiple response packets.

The data structure of the command packet is as follows:

Offset (Bytes)	Width (Bytes)	Field	Description
0	1	Command Identifier	5 - read-memory
1	1	Number of command parameters	3
2	1	Command type	0 - Command + data
3	1	Reserved	-
4	4	Address to read from	Device address to read from
8	4	Length	Length of data to read
12	4	Memory device identifier	0 - On-chip RAM 0x10000 - NOR FLASH mounted on XPI0 0x10001 - NOR FLASH mounted on XPI1 0x20000 - OTP

Table 49: read-memory Command Data Frame

When the slave receives this command, it first performs parameter checking. If parameter checking fails, it directly replies with Abort to the host in the second phase of the command, then reports the error code in the response packet. After waiting for the host's confirmation in the fourth phase, this command ends. If parameter checking passes, the slave replies with ACK to the host in the second phase, then notifies the host of successful parameter checking in the first response packet, and subsequently sends the requested data to the host sequentially through data packets.

The first type of response frame from the slave is as follows:

Offset (Bytes)	Width (Bytes)	Field	Description
0	1	Command Identifier	5 - read-memory

Offset (Bytes)	Width (Bytes)	Field	Description
1	1	Number of command parameters	1
2	1	Command type	2 - Response data only
3	1	Reserved	-
4	4	Command status	0 - Success Others - Error code

Table 50: read-memory Response Frame Data Structure

The pure data packet frame from the slave is as follows:

Offset (Bytes)	Width (Bytes)	Field	Description
0	1	Command Identifier	5 - read-memory
1	1	Number of command parameters	0
2	1	Command type	1 - Data only
3	1	Reserved	-
4	Variable length	Data	Data read from the storage device

Table 51: read-memory Response Frame Data Structure

Load Image (load-image)

This command is used to load and execute an executable image, or load and parse a command container. This command consists of pure data packets, with one command composed of several data packets.

The data packet structure is as follows:

Offset (Bytes)	Width (Bytes)	Field	Description
0	1	Command Identifier	6 - load-image
1	1	Number of command parameters	0
2	1	Command type	1 - Data only

Offset (Bytes)	Width (Bytes)	Field	Description
3	1	Reserved	-
4	Variable length	Data	Image data

Table 52: load-image Data Frame Structure

After data transmission is completed, the slave replies with a response frame, whose data structure is as follows:

Offset (Bytes)	Width (Bytes)	Field	Description
0	1	Command Identifier	6 - load-image
1	1	Number of command parameters	1
2	1	Command type	2 - Response data only
3	1	Reserved	-
4	4	Command status	0 - Success Others - Error code

Table 53: load-image Response Frame Data Structure

Erase Storage Device (erase)

This command can be used to erase part or all sectors of external XPI NOR. Its data structure is as follows:

Offset (Bytes)	Width (Bytes)	Field	Description
0	1	Command Identifier	7 - erase
1	1	Number of command parameters	2 - chip erase 4 - sector erase
2	1	Command type	1 - Data only
3	1	Reserved	-
4	4	Erase type	0 - sector erase 1 - chip erase
8	4	Storage device identifier	0x10000 - NOR FLASH mounted on XPI0 0x10001 - NOR FLASH mounted on XPI1

Offset (Bytes)	Width (Bytes)	Field	Description
12	4	Erase address	Only valid when erase type is sector erase
16	4	Erase length	Only valid when erase type is sector erase

Table 54: erase Data Frame Structure

The response frame for the command is as follows:

Offset (Bytes)	Width (Bytes)	Field	Description
0	1	Command Identifier	7 - erase
1	1	Number of command parameters	1
2	1	Command type	2 - Response data only
3	1	Reserved	-
4	4	Command status	0 - Success Others - Error code

Table 55: erase Response Frame Data Structure

Reset (reset)

This command can reset the SoC. Its data structure is as follows:

Offset (Bytes)	Width (Bytes)	Field	Description
0	1	Command Identifier	8 - reset
1	1	Number of command parameters	1
2	1	Command type	1 - Data only
3	1	Reserved	-
4	4	Reset type	This field is reserved in this SoC

Table 56: reset Data Frame Structure

The response frame for this command is as follows:

Offset (Bytes)	Width (Bytes)	Field	Description
0	1	Command Identifier	8 - reset
1	1	Number of command parameters	1
2	1	Command type	2 - Response data only
4	4	Command status	0 - Success Others - Error code

Table 57: reset Response Frame Data Structure

18.2.4 Serial Port Communication

The default serial port communication parameters supported by ROM are 115200, 8-N-1. The baud rate can be switched via the configure-rte command.

The protocol data packets for serial port communication are divided into two types:

- Payload packet
- Acknowledgment packet

18.2.4.1 Payload Packet

The payload frame consists of four parts: packet header, packet type, payload, and CRC check. The detailed data packet structure is as follows:

Offset (Bytes)	Width (Bytes)	Field	Description
0	1	Start byte	Must be 0x5A
1	1	Packet type	0xA5 - Payload packet
2	2	Payload length	Length of payload. Cannot exceed 512 bytes
4	<Payload_Len>	Data payload	Data payload, used to carry data in the command protocol
4 + <Payload_Len>	2	CRC check value	CRC check value for the first <4+Payload_Len> bytes of the entire data packet

Table 58: Payload Packet Data Structure

The CRC calculation range is from the start byte to the last byte of the data payload.

The CRC algorithm is a variant of XMODEM. Its C language implementation is as follows:

```
uint16_t crc16_update(const uint8_t *src, uint32_t lengthInBytes)
{
    uint32_t crc = 0;
    uint32_t j;
```

```

    for (j = 0; j < lengthInBytes; ++j)
    {
        uint32_t i;
        uint32_t byte = src[j];
        crc ^= byte << 8;
        for (i = 0; i < 8; ++i)
        {
            uint32_t temp = crc << 1;
            if (crc & 0x8000)
            {
                temp ^= 0x1021;
            }
            crc = temp;
        }
    }
    return (uint16_t)crc;
}

```

18.2.4.2 Acknowledgment Packet (ACK Packet)

The acknowledgment packet is used to implement ACK, NAK, or Abort for payload packets. The detailed data packet structure is as follows:

Offset (Bytes)	Width (Bytes)	Field	Description
0	1	Start byte	Must be 0x5A
1	1	Packet type	0xA1 - ACK 0xA2 - NAK 0xA3 - Abort

Table 59: Acknowledgment Packet

The sender of each payload packet must wait for the receiver's acknowledgment packet before proceeding with subsequent data transmission.

The payload part of the data packet can carry various commands or data described in the previous section.

18.2.5 USB-HID Communication

BootROM supports communication via USB-HID to implement functions such as FLASH programming, EFUSE programming, and loading boot images.

18.2.5.1 USB-HID Descriptor Information

The USB descriptor configuration in BootROM is shown in the following table:

Descriptor Identifier	Content/Value
VID	0x34b7
PID	0x0008
Vendor descriptor	HPMICRO Semiconductor Co., Ltd
Product descriptor	Boot ROM Open (factory setting) Boot ROM Closed (mass production setting) Boot ROM Return (return-to-factory setting)
HID packet length	When connected to High-Speed HUB: 516 bytes When connected to Full-Speed HUB: 64 bytes
HID interrupt transfer interval	When connected to High-Speed HUB: 125us When connected to Full-Speed HUB: 1ms

Table 60: USB Descriptors

18.2.5.2 USB Endpoint Information

BootROM uses the following three endpoints:

- Control (Endpoint 0) - Used to get device descriptors
- OUT (Endpoint 1) - Used for Host to send data to Device
- IN (Endpoint 2) - Used for Device to send data to Host

18.2.5.3 USB-HID Communication Protocol

The USB-HID communication protocol is similar to UART, both containing payload packets and acknowledgment packets.

USB-HID Payload Packet

The USB-HID payload packet contains four fields: packet header, packet type, payload length, and data payload. The detailed data structure is as follows:

Offset (Bytes)	Width (Bytes)	Field	Description
0	1	Start byte	0x1 for Host to Device, 0x2 for Device to Host
1	1	Packet type	0xA5 - Payload packet
2	2	Payload length	Payload length. Cannot exceed 512 bytes
4	<Payload_Len>	Data payload	Data payload, used to carry data in the command protocol

Table 61: USB-HID Payload Packet Data Structure

USB-HID Acknowledgment Packet

The USB-HID acknowledgment packet contains three fields: packet header, packet type, and payload length. The detailed acknowledgment packet data structure is as follows:

Offset (Bytes)	Width (Bytes)	Field	Description
0	1	Start byte	0x1 for Host to Device, 0x2 for Device to Host
1	1	Packet type	0xA1 - ACK 0xA2 - NAK 0xA3 - Abort
2	2	Payload length	Fixed at 0

Table 62: USB-HID Acknowledgment Packet Data Structure

18.3 Boot Image

The boot image supported by this device consists of two parts: Firmware Container (FW Container) and Firmware. The FW Container is composed of FW Container Header, FW Info Table, Device Configuration Block, and Signature Block. The layout of the boot image is shown in the figure below.

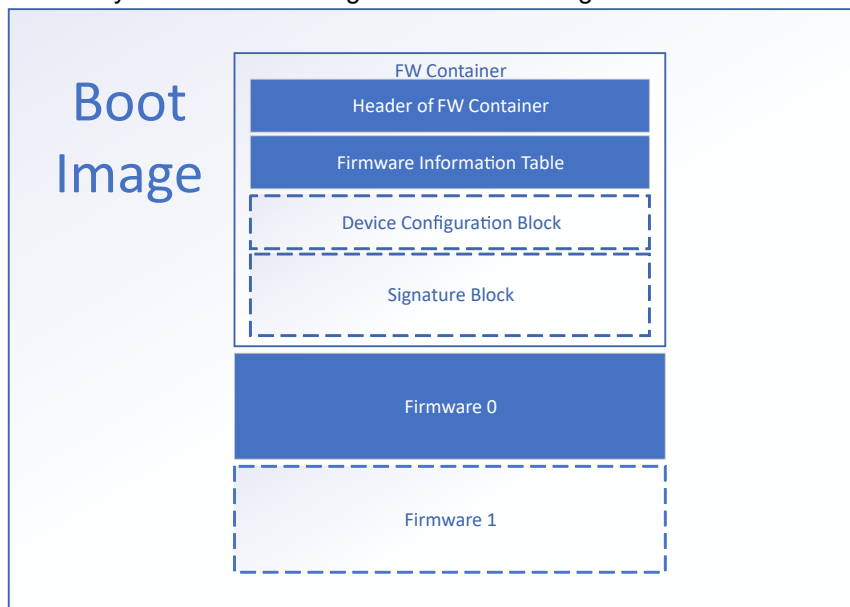


Figure 17: Boot Image Layout

18.3.1 Firmware Container Header

The detailed data structure of the firmware container header is as follows:

Table 63: Firmware Container Header

Offset (Bytes)	Field	Width (Bytes)	Description
0x00	tag	1	Boot image tag, must be 0xBF

Table 63: Firmware Container Header

Offset (Bytes)	Field	Width (Bytes)	Description														
0x01	version	1	Boot image format version bit[7:4] - Major version bit[3:0] - Minor version This value must be 0x10 on this SoC														
0x02	length	2	Length of firmware header Includes firmware container header, firmware info table, device configuration block, and signature block														
0x04	flags	4	Corresponding flags <table><tr><td>Bit[3:0]</td><td>SRK set index</td></tr><tr><td>Bit[7:4]</td><td>SRK index</td></tr><tr><td>Bit[11:8]</td><td>SRK_REVOKE_MASK Each bit represents revoking the SRK at the corresponding position. When this byte is non-zero, BootROM will attempt to revoke the corresponding SRK according to this field's value and program the PUK_REVOKE in OTP with the same value</td></tr><tr><td>Bit[19:16]</td><td>Signature type 1 - ECDSA P256 4 - SM3</td></tr><tr><td>Bit[23:20]</td><td>Firmware BLOB location 0 - BLOB in signature block 1 - BLOB in storage medium firmware location</td></tr><tr><td>Bit[27:24]</td><td>Lifecycle 0 - No change to lifecycle Others - BootROM will switch lifecycle to this field's value</td></tr><tr><td>Bit[31:28]</td><td>Reserved</td></tr></table>	Bit[3:0]	SRK set index	Bit[7:4]	SRK index	Bit[11:8]	SRK_REVOKE_MASK Each bit represents revoking the SRK at the corresponding position. When this byte is non-zero, BootROM will attempt to revoke the corresponding SRK according to this field's value and program the PUK_REVOKE in OTP with the same value	Bit[19:16]	Signature type 1 - ECDSA P256 4 - SM3	Bit[23:20]	Firmware BLOB location 0 - BLOB in signature block 1 - BLOB in storage medium firmware location	Bit[27:24]	Lifecycle 0 - No change to lifecycle Others - BootROM will switch lifecycle to this field's value	Bit[31:28]	Reserved
Bit[3:0]	SRK set index																
Bit[7:4]	SRK index																
Bit[11:8]	SRK_REVOKE_MASK Each bit represents revoking the SRK at the corresponding position. When this byte is non-zero, BootROM will attempt to revoke the corresponding SRK according to this field's value and program the PUK_REVOKE in OTP with the same value																
Bit[19:16]	Signature type 1 - ECDSA P256 4 - SM3																
Bit[23:20]	Firmware BLOB location 0 - BLOB in signature block 1 - BLOB in storage medium firmware location																
Bit[27:24]	Lifecycle 0 - No change to lifecycle Others - BootROM will switch lifecycle to this field's value																
Bit[31:28]	Reserved																
0x08	sw_version	2	Software version number, defines the version of current Image. When there are multiple Images in storage medium, ROM determines which one is newer based on this field														

Table 63: Firmware Container Header

Offset (Bytes)	Field	Width (Bytes)	Description
0x0A	fuse_version	1	This field is used for Image anti-rollback protection. The value of this field must not be less than the value of SW_VER field in OTP Note: The number of set bits in SW_VER represents the version number (e.g., 0xffff represents version 15)
0x0B	number_of_fw	1	Specifies the number of firmware contained in current Image. Current SoC supports up to two
0x0C	device_config_block_offset	2	Offset of device configuration block relative to firmware container header. Device configuration block is optional. If this block is not present in current Image, this field must be 0
0x0E	signature_block_offset	2	Offset of signature block relative to firmware container header. If signature block does not exist, this field must be 0

Table 63: Firmware Container Header

18.3.2 Firmware Info Table

The firmware info table mainly provides firmware execution address, length, offset relative to firmware container header in storage medium, firmware hash value, etc. Its detailed data structure is as follows:

Offset (Bytes)	Field	Width (Bytes)	Description
0x00	offset	4	Firmware offset relative to firmware container header
0x04	size	4	Firmware size

Offset (Bytes)	Field	Width (Bytes)	Description	
0x08	flags	4	Corresponding flags	
			Bit[3:0]	fw_type - Firmware type 0 - Executable image 1 - Command container
			Bit[7:4]	core_id - CPU ID number 0 - CPU0 1 - CPU1
			Bit[11:8]	hash_type - Hash type 0 - None 1 - SHA256 4 - SM3
			Bit[15:12]	is_encrypted - Whether encrypted 0 - Not encrypted 1 - Firmware encrypted
			Others	Reserved
0x0C	reserved	4	Reserved for future use	
0x10	load_addr	4	Firmware load address	
0x14	reserved	4	Reserved for future use	
0x18	entry_point	4	Firmware entry point	
0x1C	reserved	4	Reserved for future use	
0x20	hash	64	Firmware hash value, currently only 32 bytes are used. When firmware type is executable file, this field is the hash of entire FW. When firmware type is command container, this field is the hash of command container header packet. When firmware is encrypted, the first 128-bits of this field are used as initial vector	
0x60	reserved	32	Reserved for future use	

Table 64: Firmware Container Header

18.3.3 Device Configuration Info Block

The device configuration block can support the following 4 types of configuration information:

- XPI NOR configuration information
- Wake-up entry check information
- OTP UNLOCK information

The device configuration info block consists of the following parts:

- Device Configuration Block Header
- Device configuration information
 - Device configuration info header

– Device configuration info content

The data structure of device configuration info header is as follows:

Bit[31:16]	Bit[15:8]	Bit[7:0]
Configuration info block length	Version number, currently 0x10	Configuration info type 0xC0 - General device configuration info

Table 65: Configuration Info Block Header

Each configuration info has its own header, defined as follows:

Bit[31:16]	Bit[15:8]	Bit[7:0]
Configuration info block length	Parameters For XPI NOR, XPI RAM represents XPI instance number	Configuration info type 0xC1 - XPI NOR configuration info 0xC2 - XPI RAM configuration info 0xC3 - SDRAM configuration info 0xC4 - General register info 0xC5 - Wake-up verification configuration info 0xC6 - OTP unlock info

Table 66: Configuration Info Header

Following the configuration info header is the corresponding configuration information. The detailed definitions of each configuration info are shown in subsequent sections.

18.3.3.1 XPI NOR Configuration Info

XPI NOR configuration info is divided into two categories: one is simplified XPI NOR configuration info, and the other is complete XPI NOR configuration block.

XPI NOR Configuration Options

Offset (Bytes)	Field	Description
0x00	Header	Bit[31:12] - tag: Must be 0xfcf90 Bit[11:4] - Reserved Bit[3:0] - words: Number of configuration option words (excluding Header itself)

Offset (Bytes)	Field	Description
0x04	Option0	Bit[31:28] - Probe type 0 - SFDP SDR, 1 - SFDP DDR 2 - 1-4-4 Read 0xEB, 3 - 1-2-2 Read 0xBB 4 - HyperBus 1V8, 5 - HyperBus 3V0 6 - OctaBus DDR, 8 - Xccela DDR 10 - EcoXiP DDR Bit[27:24] - Number of data pins for sending commands after power-on reset 0 - SPI, 1 - DPI, 2 - QPI, 3 - OPI Bit[23:20] - Number of data pins for sending commands after FLASH configuration 0 - SPI, 1 - DPI, 2 - QPI, 3 - OPI Bit[19:16] - 1-4-4 mode enable sequence 0 - Not required or automatic 1 - QE bit is at bit 6 in Status Register1 2 - QE bit is at bit1 in Status Register2 3 - QE bit is at bit7 in Status Register2 4 - QE bit is at bit1 in Status Register2 and should be programmed by 0x31 Bit[15:8] - dummy cycles 0 - Auto detect/default value Others - User specified value, for DTR read, this value needs to be <i>dummycycle</i>×2 as specified in actual datasheet Bit[7:4] - Miscellaneous 0 - None 1 - SPI mode 2 - Internal loopback mode 3 - Force external DQS Bit[3:0] - Frequency option 0 - Do not change existing configuration Others - Corresponding value for SoC

Offset (Bytes)	Field	Description
0x08	Option1	This field is valid when "words" field value is greater than 1 Bit[31:20] - reserved Bit[19:16] - IO voltage 0 - IO is 3V, 1 - IO is 1.8V Bit[15:12] - Pin group selection 0 - First group, 1 - Second group Bit[11:8] - Connection method selection 0 - PORTA_CS0, 1 - PORB_CS0, 2 - PORTA_CS0 +PORTB_CS0, 3 - PORTA_CS0+PORTA_CS1, 4 - PORTB_CS0+PORTB_CS1 Bit[7:0] - IO drive strength 0 - Default value (maximum drive strength) Others - See details (TBD)
0x0C	Option2	This field is valid when "words" is greater than 2 Bit[31:16] - reserved Bit[15:12] - Erase sector command option 0 - Erase 4KB, 1 - Erase 32KB, 2 - Erase 64KB, 3 - Erase 256KB Bit[11:8] - Sector size option 0 - 4KB, 1 - 32KB, 2 - 64KB, 3 - 256KB Bit[7:0] - FLASH size option 0 - 4MB, 1 - 8MB, 2 - 16MB

Table 67: XPI NOR Configuration Options

XPI NOR Configuration Block

The data structure of XPI NOR configuration block is shown in the following table:

Offset (Bytes)	Field	Width (Bytes)	Description
0x000	tag	4	Configuration block tag, fixed as 0x524f4e58
0x004	reserved	4	-
0x008	rxclk_src	1	Sampling clock source 0 - Internal loopback 1 - DQS internal loopback 3-External DQS input
0x009	clk_freq	1	Clock frequency option, see details in 18.8.2.1
0x00A	drive_strength	1	Drive strength 0 - Default value (maximum drive strength) Others - See details (TBD)

Offset (Bytes)	Field	Width (Bytes)	Description
0x00B	column_addr_width	1	Column address width Only valid for HyperFLASH, should be set to 0 for other FLASH
0x00C	rxclk_src_for_init	1	rxclk_src value during FLASH initialization
0x00D	config_in_progress	1	Indicates whether FLASH is currently in initialization process 1 - Initializing, Others - Not initializing
0x00E	reserved	2	-
0x010	port_info	16	- Connection port information This field is divided into 4 groups, representing CA_CS0, CA_CS1, CB_CS0, CB_CS1 respectively. Each group's 4-byte definition is as follows: Byte 0 - Enable: Enable bit, indicates this connection is enabled Byte 1 - Group: Pin group: 0 represents first group, 1 represents second group Byte 2, Byte 3 - Reserved for future use
0x020	device_info	0x50	Detailed definition see 69
0x070	instruction_set	0x90	Standard XPI NOR instruction set, divided into 9 groups, each group of 16 bytes represents a complete instruction sequence, group information as follows: 0 - FLASH Read sequence 1 - FLASH Write sequence 2 - FLASH Read Status sequence 3 - FLASH Read Status XPI sequence (QPI/OPI mode) 4 - FLASH Write Enable sequence 5 - FLASH Write Enable XPI sequence (QPI/OPI mode) 6 - FLASH Erase Sector sequence 7 - FLASH Erase Block sequence 8 - FLASH Erase Chip sequence Note: For HyperFLASH, the above definitions do not apply. BootROM supports dedicated FLASH operations

Table 68: XPI NOR Configuration Block

The data structure of XPI NOR device info table is as follows:

Offset (Bytes)	Field	Width (Bytes)	Description
0x00	size_in_kbytes	4	FLASH size (in KB)
0x04	page_size	2	Page size

Offset (Bytes)	Field	Width (Bytes)	Description
0x06	sector_size_kbytes	2	Sector size (in KB)
0x08	block_size_kbytes	2	Block size (in KB)
0x0A	busy_offset	1	Busy bit offset (in FLASH Status register)
0x0B	busy_polarity	1	Busy bit polarity: 0 - 1 represents busy 1 - 0 represents busy
0x0C	data_pads	1	Number of data pins 0 - 1 wire, 1 - 2 wire, 2 - 4 wire, 3 - 8 wire
0x0D	en_ddr_mode	1	Enable DDR/DTR mode
0x0E	clk_freq_for_dev_cfg	1	Clock frequency used for FLASH configuration Clock frequency definition see 18.8.2.1
0x0F	working_mode_por	1	FLASH working mode after power-on reset 0 - Extended SPI 1 - XPI(DPI/QPI/OPI) 2 - HyperBUS
0x10	working_mode	1	FLASH working mode after initialization 0 - Extended SPI 1 - XPI(DPI/QPI/OPI) 2 - HyperBUS
0x11	en_diff_clk	1	Enable differential clock Only 1.8V HyperFLASH needs to set this option to 1, other FLASH should set to 0
0x12	data_valid_time	1	-
0x13	en_half_clk_for_ non_read_cmd	1	Enable half clock frequency reduction for non-read commands
0x14	clk_freq_for_ non_read_cmd		Set clock frequency for non-read commands Clock frequency definition see 18.8.2.1
0x15	reserved	3	-
0x18	cs_hold_time	1	Set CS Hold clock cycle count
0x19	cs_setup_time	1	Set CS Setup clock cycle count
0x1A	cs_interval	1	Interval clock cycles between two commands
0x1B	en_dev_mode_cfg	1	Enable device mode configuration
0x1C	flash_state_ctx	4	FLASH state context Context definition see (TBD)
0x20	mode_cfg_list	4	Contains two groups, each group 2 bytes, each group's data structure as follows: Byte 0 - Configuration command type Byte 1 - Parameter size
0x24	mode_cfg_param	8	Contains 2 groups, each group 4 bytes, corresponding one-to-one with elements in the above list

Offset (Bytes)	Field	Width (Bytes)	Description
0x2C	reserved	4	-
0x30	cfg_instr_seq	32	Contains 2 groups, each group of 16 bytes is a command sequence, corresponding one-to-one with elements in the above list Detailed command sequence definition see (TBD)

Table 69: XPI NOR Configuration Block

18.3.3.2 Wake-up Verification Info

Wake-up verification info provides parameters used to verify the integrity and legality of wake-up programs in wake-up mode, including wake-up program entry address, wake-up program code start address, length, and corresponding SHA256 hash value. Under secure boot enabled conditions, all information is signed and verified by ROM.

Detailed data structure is as follows:

Offset (Bytes)	Field	Width (Bytes)	Description
0x00	Start	4	Wake-up code start address
0x04	wake-up entry	4	Wake-up code entry address Note: This address must be within the address range specified by start and length
0x08	Length	4	Wake-up code length
0x0C	Reserved	4	-
0x10	wake-up entry HASH	32	Hash value of code within start and length specified range

Table 70: Wake-up Verification Info

18.3.3.3 OTP Unlock Info

The data structure for OTP unlock operation is as follows:

Bit[31:24]	Bit[23:16]	Bit[15:8]	Bit[7:0]
Tag, must be 0x55	Reserved	Reserved	Unlock mask When Lifecycle is Closed or higher, BootROM will disable write permissions for corresponding OTP fields by default. When BootROM detects the following bits are set to 1 in OTP unlock info, it will unlock the corresponding write permissions Bit 0 - HARD_LOCK Bit 1 - LIFECYCLE, PUBKEY_REVOKE, etc. Bit 2 - Boot configuration info part Bit 3 - OTP KEY0 (for Encrypt boot) Bit 4 - OTP KEY1 (for Encrypt boot)

Table 71: OTP Unlock Info

18.3.4 Signature Block

The complete signature block provides the following information:

- Signature Block Header
- Super Root Key Table (SRK Table) - This part contains root key table header SRK Table Header and 4 root key table entries (SRK0 - SRK3)
- Signature
- Certificate - This part is optional
- Firmware Blob (FW Blob)

The layout of each part in the signature block in target storage is shown in the figure:

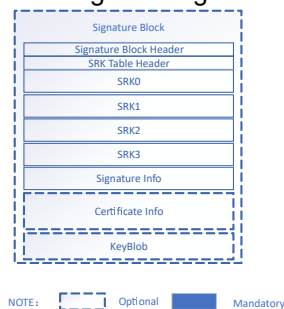


Figure 18: Layout of Parts in Signature Block

18.3.4.1 Signature Block Header

The signature block header contains the following information:

- Signature block length
- Signature block version number
- Signature identifier
- Root Key Table (SRK Table) relative offset
- Certificate relative offset
- Signature relative offset
- FW Blob relative offset (if FW Blob is within Container)

The detailed data structure is shown in the following table:

Offset (Bytes)	Field	Width (Bytes)	Description
0x00	Header	4	Signature block Header
			Bit[31:16] Length - Entire signature block length
			Bit [15:8] Version - Signature block version, fixed as 0x10 on this SoC
			Bit [7:0] Tag - Signature block tag, fixed as 0xD0
0x04	Flags	4	Signature block flags Bit0 Key identifier 1 - OEM Key Others - Reserved bits

Offset (Bytes)	Field	Width (Bytes)	Description								
0x08	Member_Offset	8	Relative offsets of various parts in signature <table><tr><td>Bit[15:0]</td><td>srk_table_offset SRK table offset relative to signature block start address</td></tr><tr><td>Bit[31:16]</td><td>signature_offset Signature offset relative to signature block start address</td></tr><tr><td>Bit[47:32]</td><td>certificate_offset(optional) Certificate offset relative to signature block start address Only valid when secondary certificate is used</td></tr><tr><td>Bit[63:48]</td><td>blob_offset(optional) Firmware BLOB offset relative to signature block start address Only valid when encrypted firmware exists and firmware BLOB is within signature block</td></tr></table>	Bit[15:0]	srk_table_offset SRK table offset relative to signature block start address	Bit[31:16]	signature_offset Signature offset relative to signature block start address	Bit[47:32]	certificate_offset(optional) Certificate offset relative to signature block start address Only valid when secondary certificate is used	Bit[63:48]	blob_offset(optional) Firmware BLOB offset relative to signature block start address Only valid when encrypted firmware exists and firmware BLOB is within signature block
Bit[15:0]	srk_table_offset SRK table offset relative to signature block start address										
Bit[31:16]	signature_offset Signature offset relative to signature block start address										
Bit[47:32]	certificate_offset(optional) Certificate offset relative to signature block start address Only valid when secondary certificate is used										
Bit[63:48]	blob_offset(optional) Firmware BLOB offset relative to signature block start address Only valid when encrypted firmware exists and firmware BLOB is within signature block										

Table 72: Signature Block Header

Note: To ensure CPU access efficiency to various parts in signature block, the start addresses of SRK table, signature, certificate, and Blob must be 8-byte aligned

18.3.5 Root Key Table (SRK Table)

SRK Table consists of Root Key Table Header (SRK Table Header) and Root Key Table Items (SRK Items).

18.3.5.1 Root Key Table Header

The root key table header consists of 4 bytes:

Bit[31:16]	Bit[15:8]	Bit[7:0]
SRK table length (little-endian storage)	SRK table version, currently 0x10	SRK table tag, fixed as 0xD1

Table 73: Root Key Table Header

18.3.5.2 Root Key Table Item

The root key table item contains public key related information. The detailed data structure is as follows:

Offset (Bytes)	Field	Width (Bytes)	Description

Offset (Bytes)	Field	Width (Bytes)	Description
0x00	Header	4	Bit[31:16] Root key table item length (little-endian storage) Bit[15:8] Key type, 0x21 - Public key Tag - Must be 0xE1
0x04	Flags	4	Bit[31:16] - Reserved Bit[15:8] Curve type: 1 - ECDSA P256, 4 - SM2 Bit[7:0] Hash type: 1 - SHA256, 4 - SM3
0x08	Key Info	4	Bit[7:0] - <X Length> : Big number X length Bit[23:16] - <Y Length> : Big number Y length Others - Reserved bits
0x0C	Big Number X	<X Length>	Big number X
0x0C + <X Length>	Big Number Y	<Y Length>	Big number Y

Table 74: Root Key Table Item

18.3.6 Signature Info

The data structure of signature info is as follows:

Offset (Bytes)	Field	Width (Bytes)	Description
0x00	Header	4	Bit[31:16] - <Sig_Len>: Signature info length (little-endian) Bit[15:8] - Signature info version, currently fixed as 0x10 Bit[7:0] - Tag, fixed as 0xD2
0x04 0x08	reserved Signature Data	4 <Sig_Len>	Reserved for future use Signature data

Table 75: Signature Info

18.3.7 Certificate

The data structure of certificate is as follows:

Offset (Bytes)	Field	Width (Bytes)	Description
0x00	Header	4	Bit[31:16] - <Sig_Len>: Certificate length (little-endian) Bit[15:8] - Certificate version, currently fixed as 0x10 Bit[7:0] - Tag, fixed as 0xD3
0x04	Offset	4	Signature info offset relative to certificate header

Offset (Bytes)	Field	Width (Bytes)	Description
0x08	SRK Table Item	<SRK_Table_Item_Len>	Key table item, detailed information see (18.3.5.2)
0x08 + <SRK_Table_Item_Len>	Signature Info	<Signature_Info_Len>	Signature info, detailed information see (18.3.6)

Table 76: Certificate Header

18.3.8 Command Container

Command container is a packaged encapsulation of several commands supported by ROM. This format supports two forms: encrypted (AES-CBC/SM4-CBC) or plaintext. Users can choose the corresponding packaging method according to their system security requirements.

Command container data packets are packaged in blocks. The block length is variable, indicated by the next_block_info field at the end of the previous block. The first block is fixed as command container header with fixed length. The last block does not contain next_block_info.

Offset (Bytes)	Field	Width (Bytes)	Description
0x00	Payload	<Data_Bytes>	Data payload part
<Data_Bytes>	next_block_info	16	Next block information Byte 0-1 : Next block size Byte 2-3 : Whether it is the last block Byte 4-15: Reserved
<Data_Bytes> + 16	next_block_hash	16	Next block hash value. Hash value length is determined by hash_type in 64

Table 77: Command Container

18.3.8.1 Command Container Header

Offset (Bytes)	Field	Width (Bytes)	Description
0x00	Tag	4	Command container tag: 0x33764343(ASCII:CCv1)
4	Version	4	Command container format version, currently 0x56010000(V1.0.0)
8	Flags	4	Reserved for future use
12	Reserved	48	Reserved for future use
64	Description	48	Simple description of command container, e.g., (xpi nor boot image)

Table 78: Command Container Header

18.3.8.2 Commands Supported by Command Container

Command container supports the following commands:



- configure-memory
- write-memory
- erase
- reset

The data block structure of Configure-memory is as follows:

Offset (Bytes)	Field	Width (Bytes)	Description
0x00	Command Tag	1	Command identifier - 0x03
0x01	Reserved	1	Reserved
0x02	Command Misc.	1	Command miscellaneous - fixed as 0
0x03	Reserved	1	Reserved
0x4	Memory Identifier	4	Storage device identifier 0x10000 - XPI0 0x10001 - XPI1
0x08	Reserved	4	Reserved
12	Configuration Block Data Bytes	4	Configuration block data bytes
16	Configuration Block Data	<Configuration Block Data Bytes>	Configuration block data When data length is less than 16 bytes, it needs to be padded with 0 to 16-byte alignment

Table 79: Configure Storage Device Data Block

Write-memory data block structure is as follows:

Offset (Bytes)	Field	Width (Bytes)	Description
0x00	Command Tag	1	Command identifier - 0x04
0x01	Reserved	1	Reserved
0x02	Command Misc.	1	Command miscellaneous - fixed as 0
0x03	Reserved	1	Reserved
0x4	Memory Identifier	4	Storage device identifier 0x10000 - XPI0 0x10001 - XPI1 0x20000 - OTP
0x08	Start	4	Start address
12	Data Bytes	4	Number of bytes to write

Offset (Bytes)	Field	Width (Bytes)	Description
16	Data Block	<Data Bytes>	Data to write (maximum 256 bytes) When data length is less than 16 bytes, it needs to be padded with 0 to 16-byte alignment

Table 80: Write to Storage Device Data Block

Erase data block structure is as follows:

Offset (Bytes)	Field	Width (Bytes)	Description
0x00	Command Tag	1	Command identifier - 0x07
0x01	Reserved	1	Reserved
0x02	Command Misc.	1	Command miscellaneous: 0x0 - Sector Erase 0x1 - Chip Erase
0x03	Reserved	1	Reserved
0x4	Memory Identifier	4	Storage device identifier 0x10000 - XPI0 0x10001 - XPI1
0x08	Start	4	Start address
12	Length	4	Number of bytes to erase (ignored for Chip Erase)

Table 81: Erase Storage Device Data Block

Reset data block structure is as follows:

Offset (Bytes)	Field	Width (Bytes)	Description
0x00	Command Tag	1	Command identifier - 0x08
0x01	Reserved	1	Reserved
0x02	Command Misc.	1	Command miscellaneous, fixed as 0
0x03	Reserved	1	Reserved
0x4	Reset Type	4	Reset type, reserved in this SoC
0x08	Reserved	8	Reserved bits

Table 82: Reset Data Block

18.3.8.3 Encrypted Command Container

Command container supports encryption in AES-CBC/SM4-CBC mode. The IV is the first 128 bits of the current block's hash value. The command container encryption key is packaged in the FW Blob. The IV for the command container header is the first 128 bits of the hash value in the firmware info table.

18.4 Secure Boot

18.4.1 Secure Boot Introduction

This device supports secure boot.

Secure boot supports digital certificate signature and authentication derived from Public Key Infrastructure (PKI).

The secure boot of this device supports the following two certificate signature and authentication modes:

1. Direct signature authentication based on primary certificates generated by CA, supporting up to 4 primary certificates.
2. Signature authentication based on secondary certificates issued by primary certificates, adopting a chain of trust for hierarchical authentication, supporting up to 4 primary certificates and 1 secondary certificate.

This device supports ECDSA P256 and SM2 signature authentication.

To simplify the secure boot process and reduce unnecessary format conversion and parsing, BootROM uses fixed encoding for the following parts:

- Public key encoding - See details in [Root Key Table](#) section
- Digital signature encoding - See details in [Signature Information](#) section
- Certificate - See details in [Certificate](#) section

18.4.2 Secure Boot Flow

To reduce the overall secure boot duration, the design model for secure boot is as follows:

1. Only verify the signature of the FW Container header. The data to be signed/verified is shown in the following figure:

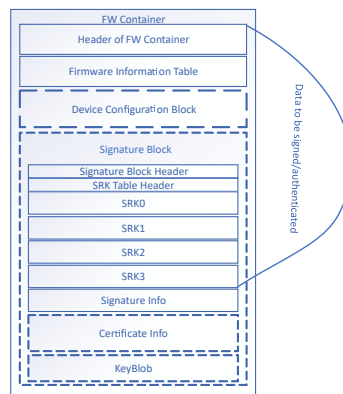


Figure 19: Firmware Header Part to be Signed

2. Use hash value comparison for the FW, where the FW's hash value is included in the firmware information table, as shown in the figure above. Since the hash value itself has been signature-verified, this design ensures signature uniqueness.
3. After the FW header signature verification passes, BootROM parses the firmware table and processes the verification of each FW according to the verification method specified in the table. This device supports the following verification methods:
 - Plaintext hash value verification based on SHA256 or SM3

4. When both the FW Container header verification and all firmware verifications pass, the entire boot image security authentication is successful. BootROM supports on-demand booting or processing of the corresponding firmware. If any step in the above process fails verification, the boot image is considered invalid.

18.4.3 Firmware Container Signature Verification

For secure boot supporting primary certificates, the flow is as follows:

1. BootROM parses the FW Container, obtains the signature block, extracts the SRK table from the signature block, and calculates the hash value of the SRK table
2. BootROM obtains the SRK table hash value from OTP (determined by SRK_SET in the Container header), and compares it with the hash value calculated in the previous step. If they match, the SRK is valid
3. Calculate the hash value of the data from the FW Container header to the end of the SRK table as the signed data
4. Use the SRK specified in the FW Container header to verify the signature part. If verification passes, the FW Container header is legitimate and has not been tampered with.

For secure boot supporting secondary certificates, the flow is as follows:

1. BootROM parses the FW Container, obtains the signature block, extracts the SRK table from the signature block, and calculates the hash value of the SRK table
2. BootROM obtains the SRK table hash value from the SRK_HASH field in OTP (determined by SRK_SET in the Container header), and compares it with the hash value calculated in the previous step. If they match, the SRK is valid
3. Calculate the hash value of the certificate in the certificate block as the verification data for the secondary certificate
4. Use the SRK specified in the FW Container header to verify the signature part. If verification passes, the secondary certificate is legitimate; otherwise, secure boot fails
5. Calculate the hash value of the data from the FW Container header to the end of the SRK table as the signed data
6. Use the secondary certificate to verify the signature of the data from the previous step. If verification passes, the FW Container header is legitimate and has not been tampered with

18.4.4 Firmware Signature Verification

After the FW Container signature verification passes, if the device configuration block exists, BootROM completes device configuration according to the parameters in the device configuration block, then proceeds to the firmware information processing stage. ROM supports the following firmware signature verification methods:

- Plaintext hash verification: BootROM calculates the FW hash value based on the address and length specified in the firmware information table, and compares the result with the result in the firmware information table.
 - When the public key type for FW Container verification is ECDSA P256, the firmware hash is based on SHA256 algorithm
 - When the public key type for FW Container verification is SM2, the firmware hash is based on SM3 algorithm or SHA256 algorithm

18.4.5 Public Key Revocation

As product usage time increases, the currently used public key may face risks such as expiration or private key leakage. When this occurs, the currently used public key needs to be revoked. BootROM does not provide automatic public key revocation. User programs need to revoke the corresponding public key as needed. BootROM provides two methods for public key revocation:

- Revocation via FW Container.
Set the bit corresponding to the current key in the SRK_REVOKE field of the [Firmware Container Header](#) to 1, sign the new image with the new private key, and burn the new image into storage medium or load it via serial boot mode. When the new image is loaded successfully, BootROM will automatically revoke the specified public key.
- Revocation via user program.
Use the device configuration information block in the image, and unlock the write permission for PUK_REVOKE. Sign the new image with the new private key, and burn the new image into storage medium or load it via serial boot mode. When the new image is successfully booted, the user can write to PUK_REVOKE via the OTP driver to achieve public key revocation.

18.5 Low Power Wake-up

In specific low power modes, the CPU core powers down, with only essential modules remaining powered. Upon specific events (such as pressing a wake-up button), the CPU core can be woken up and repowered to execute BootROM code, which then provides subsequent wake-up support. ROM supports the following low power wake-up options:

- Fast Jump: Directly jump to the user-specified wake-up address for execution, performing only basic address range validity checks
- Strong Verification Jump: Perform integrity checks (SHA256) on the user-specified range
- Disable Low Power Wake-up: Force cold boot, ignoring low power wake-up options

18.5.1 Fast Jump

Users need to fill a valid wake-up address in SYSCTL.CPU0_GPR0 and configure the correct wake-up source. If the address is invalid, BootROM will perform a complete boot process.

Example code is as follows:

```
HPM_SYSCTL->CPU0_LP |= SYSCTL_CPU0_LP_STOP_MASK;
HPM_SYSCTL->RETENTION[0].VALUE = 0;
for (uint32_t i = 0; i < 4; i++) {
    HPM_SYSCTL->CPU0_WAKEUP_ENABLE[i] = ~0;
}
HPM_PMIC_WDG->WREN = 0x5AA5UL;
HPM_PMIC_WDG->CTRL = WDT_CTRL_INTTIME(2) | WDT_CTRL_RSTEN(0) | WDT_CTRL_INTEN(1) \
| WDT_CTRL_CLKSEL(1) | WDT_CTRL_EN(1);
HPM_SYSCTL->CPU0_GPR0 = (uint32_t)&lowpower_wakeup_test;
__asm("wfi");
```

18.5.2 Strong Verification Jump

When the FORCE_WAKEUP_ENTRY_CHK OTP bit is set, BootROM forcibly performs integrity checks on the wake-up code. Only after the integrity check passes will BootROM execute the wake-up code; otherwise, BootROM will perform a complete boot process. In addition to the fast jump configuration, users also need to fill the program's start address and length into SYSCTL.CPU0_PARAM[0], SYSCTL.CPU0_PARAM[1], and fill the SHA256 hash value of the entire wake-up code into SYSCTL.CPU0_DATA[0] - SYSCTL.CPU0_DATA[7]. The address entry in SYSCTL.CPU_GPR0 must be within the address range specified by CPU0_GPR1 and CPU0_GPR2.

When CPU0 is woken up, ROM will calculate the SHA256 hash value based on the address range in SYSCTL.CPU0_PARAM[0] and SYSCTL.CPU0_PARAM[1], and compare it with the value in SYSCTL.CPU0_DATA[0] - SYSCTL.CPU0_DATA[7]. Only when the results match will BootROM jump to the wake-up address entry.

To ensure security, users can lock the above registers via the SYSCTL.CPU0_LOCK register.

Activation methods:

- Configure in user code as described above
- Configure in the [Device Configuration Information Block](#), see details in the [Wake-up Verification Information section](#). This method is recommended when users need BootROM to ensure the wake-up code is also secure boot authenticated. In this mode, BootROM will lock the above registers before jumping to the user program.

18.5.3 Disable Low Power Wake-up

When the FORCE_COLD_BOOT OTP bit is set, BootROM ignores wake-up requests and performs a complete secure boot process.

18.6 Debug Interface Permission Management

Access to the SoC's debug interface is jointly managed by SoC hardware and ROM.

- When the device is in CREATE mode, the Debug interface is enabled by default, and BootROM execution can be interrupted and taken over at any time.
- After the device leaves the factory, the Debug interface is disabled by default, and Debug interface control is taken over by BootROM. When the SoC leaves BootROM, BootROM will enable access to the Debug interface.
 - In ISP/Serial boot mode, when BootROM detects a DEBUG request, it will enable Debug interface access and enter a while(1) loop.
 - When a normal boot image is present, BootROM will enable Debug interface access before jumping to the APP and continue executing the user application.

18.7 ROM API

This section details all API interfaces supported by this SoC.

18.7.1 Introduction

BootROM exports commonly used FLASH operation API interfaces and security-related API interfaces. User programs can directly call these interfaces for fast FLASH and security-related programming to save development and verification time. APIs are categorized by module. The following are the main API categories supported by the current SoC:

- run_bootloader interface
- XPI NOR driver interface
- XPI RAM interface
- XPI low-level driver interface
- OTP
- Secure Boot API interface
- EXiP API interface

In this SoC, the APIs provided by BootROM are organized based on the API root list located at 0x2001_FF00. The data structure of the API root is as follows:

```
typedef struct {
    const uint32_t version;                /*!< offset: 0x00 */
    const char *copyright;                 /*!< offset: 0x04 */
    hpm_stat_t (*const run_bootloader)(void *arg); /*!< offset: 0x08 */
    const otp_driver_interface_t *otp_driver_if; /*!< offset: 0x0c */
    const xpi_driver_interface_t *xpi_driver_if; /*!< offset: 0x10 */
    const xpi_nor_driver_interface_t *xpi_nor_driver_if; /*!< offset: 0x14 */
    const xpi_ram_driver_interface_t *xpi_ram_driver_if; /*!< offset: 0x18 */
    const uint32_t reserved0;              /*!< offset: 0x1c */
    const sec_boot_api_interface_t *sec_boot_api_if; /*!< offset: 0x20 */
    const uint32_t reserved1[2];
    const exip_driver_interface_t *exip_api_if; /*!< offset: 0x2c */
} bootloader_api_table_t;
```

```
#define ROM_API_TABLE_ROOT (const bootloader_api_table_t*)0x2001FF00U
```

18.7.2 run_bootloader API

User programs can call this API to jump to a specified boot mode (such as entering ISP mode or selecting to boot another image). Its prototype is defined as:

```
typedef struct {
    uint32_t index:8;
    uint32_t peripheral:8;
    uint32_t src:8;
    uint32_t tag:8;
} api_boot_arg_t;
```

```
#define API_BOOT_TAG (0xEBU)
```

```
#define API_BOOT_SRC_FUSE (0U)
#define API_BOOT_SRC_PRIMARY (1U)
#define API_BOOT_SRC_SERIAL_BOOT (2U)
#define API_BOOT_SRC_ISP (3U)
#define API_BOOT_SRC_MAX (API_BOOT_SRC_ISP)
#define API_BOOT_PERIPH_AUTO (0U)
#define API_BOOT_PERIPH_UART (1U)
#define API_BOOT_PERIPH_USBHID (2U)
#define API_BOOT_PERIPH_MASK (API_BOOT_PERIPH_USBHID)

void run_bootloader(void *arg);
```

18.7.2.1 Example:

If a user wants to enter in-system programming mode in an application, it can be done as follows:

```
api_boot_arg_t boot_arg = {.index = 0,
    .peripheral = API_BOOT_PERIPH_AUTO,
    .src = API_BOOT_SRC_ISP,
    .tag = API_BOOT_TAG};
ROM_API_TABLE_ROOT->run_bootloader(&boot_arg);
```

18.7.3 OTP API

The OTP API provides read and write operations for OTP. Its prototype is defined as follows:

```
typedef struct {
    uint32_t version;
    void (*init)(void);
    uint32_t reserved;
    uint32_t (*read_from_shadow)(uint32_t addr);
    uint32_t (*read_from_ip)(uint32_t addr);
    hpm_stat_t (*program)(uint32_t addr, const uint32_t *src, uint32_t num_of_words);
    hpm_stat_t (*reload)(efuse_region_t region);
    hpm_stat_t (*lock_otp)(uint32_t addr, otp_lock_option_t lock_option);
    hpm_stat_t (*lock_shadow)(uint32_t addr, otp_lock_option_t lock_option);
    hpm_stat_t (*set_configurable_region)(uint32_t start, uint32_t num_of_words);
    hpm_stat_t (*write_shadow_register)(uint32_t addr, uint32_t data);
} otp_driver_interface_t;
```

18.7.3.1 init

This API is used to initialize the OTP driver. Ensure this API has been called before calling OTP read/write operations.

18.7.3.2 read_from_shadow

This API is used to read the specified OTP word from the OTP shadow register.

18.7.3.3 read_from_ip

This API is used to read the specified OTP word from OTP using IP read method.

18.7.3.4 program

This API is used to write data to the specified OTP address range. The parameters of this API are defined as follows:

- addr - Specified write address
- src - Pointer to the data to be written
- num_of_words - Number of words to write (32-bit units)

18.7.3.5 reload

This API is used to reload the value of the shadow register for the specified area. The parameter definition is as follows:

```
typedef enum {
    efuse_region0_mask = 1U,    /*!< Address range: [0, 7] */
    efuse_region1_mask = 2U,    /*!< Address range: [8, 15] */
    efuse_region2_mask = 4U,    /*!< Address range: [16, 127] */
    efuse_region3_mask = 8U,    /*!< Address range: user defined */
} otp_region_t;
```

18.7.3.6 lock_otp

This API can lock the corresponding permissions of the specified OTP word. Its parameters are defined as follows:

```
typedef enum {
    efuse_no_lock = 0,
    efuse_read_only = 1,
    efuse_permanent_no_lock = 2,
    efuse_disable_access = 3,
    efuse_lock_option_max = efuse_disable_access,
} otp_lock_option_t;
```

18.7.3.7 set_configurable_region

This API is used to set the user-specified configurable region.

18.7.3.8 write_shadow_register

This API can directly write the specified value to the specified shadow register.

18.7.3.9 Example

If a user wants to read the 26th word of OTP, it can be obtained in two ways: Method 1: Get through shadow register:

```
uint32_t fuse_word = ROM_API_TABLE_ROOT->otp_driver_if->read_from_shadow(26);
```

Method 2: Get through IP read:

```
uint32_t fuse_word = ROM_API_TABLE_ROOT->otp_driver_if->read_from_ip(26);
```

18.7.4 XPI Low-level Driver API

The XPI low-level API provides configuration of XPI module-specific functions and data transmission. Its API interface prototype is defined as follows:

```
typedef struct {
    uint32_t version;
    hpm_stat_t (*get_default_config)(xpi_config_t *xpi_config);
    hpm_stat_t (*get_default_device_config)(xpi_device_config_t *dev_config);
    hpm_stat_t (*init)(XPI_Type *base, xpi_config_t *xpi_config);
    hpm_stat_t (*config_ahb_buffer)(XPI_Type *base, xpi_ahb_buffer_cfg_t *ahb_buf_cfg);
    hpm_stat_t (*config_device)(XPI_Type *base, xpi_device_config_t *dev_cfg,
                                xpi_port_t port);
    hpm_stat_t (*update_instr_table)(XPI_Type *base, const uint32_t *inst_base,
                                     uint32_t seq_idx, uint32_t num);
    hpm_stat_t (*transfer_blocking)(XPI_Type *base, xpi_xfer_ctx_t *xfer);
    void (*software_reset)(XPI_Type *base);
    bool (*is_idle)(XPI_Type *base);
    void (*update_dllcr)(XPI_Type *base, uint32_t serial_root_clk_freq,
                        uint32_t data_valid_time, xpi_port_t port, uint32_t dly_target);

    hpm_stat_t (*get_abs_apb_xfer_addr)(XPI_Type *base, xpi_xfer_port_t port,
                                       uint32_t in_addr, uint32_t *out_addr);
    hpm_stat_t (*iomux_init)(XPI_Type *base, const xpi_io_config_t *io_cfg);
    hpm_stat_t (*clock_init)(XPI_Type *base, const xpi_clk_config_t *clk_cfg);
} xpi_driver_interface_t;
```

18.7.4.1 get_default_config

This API is used to get the default configuration parameters of XPI. xpi_config_t definition details can be found in the SDK.

Example:

```
xpi_config_t xpi_config;
hpm_stat_t status = ROM_API_TABLE_ROOT->xpi_driver_if->get_default_config(&xpi_config);
```

18.7.4.2 get_default_device_config

This API is used to get the default device configuration parameters. xpi_device_config_t definition can be found in the SDK.

Example:

```
xpi_device_config_t device_config;
status = ROM_API_TABLE_ROOT->xpi_driver_if->get_default_device_config(&device_config);
```

18.7.4.3 init

Configure the XPI controller according to xpi_config parameters.

Example:

```
status = ROM_API_TABLE_ROOT->xpi_driver_if->init(HPM_XPI0, &xpi_config);
```

18.7.4.4 config_ahb_buffer

Configure AHB Buffer information. The definition of xpi_ahb_buffer_cfg_t is as follows:

```
typedef struct
{
    struct
    {
        uint8_t priority;           /* Offset: 0x00 */
        uint8_t master_idx;        /* Offset: 0x01 */
        uint8_t buf_size_in_dword; /* Offset: 0x02 */
        bool enable_prefetch;      /* Offset: 0x03 */
    } entry[XPI_AHB_BUF_CR_COUNT];
} xpi_ahb_buffer_cfg_t;
```

Example:

```
xpi_ahb_buffer_cfg_t ahb_cfg;
for(uint32_t i=0; i<XPI_AHB_BUF_CR_COUNT; i++) {
    ahb_cfg.entry[i].enable_prefetch = false;
}
ahb_cfg.entry[0].priority = 0;
ahb_cfg.entry[0].master_idx = 0;
ahb_cfg.entry[0].buf_size_in_dword = 64;
ahb_cfg.entry[0].enable_prefetch = true;
status = ROM_API_TABLE_ROOT->xpi_driver_if->config_ahb_buffer(HPM_XPI0, &ahb_cfg);
```

18.7.4.5 config_device

Configure device information to the XPI controller.

Example:

```
status = ROM_API_TABLE_ROOT->xpi_driver_if->config_device(HPM_XPI0, &device_config);
```

18.7.4.6 update_instr_table

Update the instruction table to the XPI controller.

Example:

```
uint32_t lut[4] = {XPI_INSTR_SEQ(XPI_PHASE_CMD_SDR, XPI_1PAD, 0x03,
                                XPI_PHASE_RADDR_SDR, XPI_1PAD, 0x18),
                  XPI_INSTR_SEQ(XPI_PHASE_READ_SDR, XPI_1PAD, 0x04,
                                XPI_PHASE_STOP, XPI_1PAD, 0x00),
                  0, 0};
status = ROM_API_TABLE_ROOT->xpi_driver_if->update_instr_table(HPM_XPI0,
                                                                &lut[0], 0, 1);
```

18.7.4.7 transfer_blocking

Complete blocking transfer (read/write/send command). Example:

```
uint32_t buffer[128];
xpi_xfer_ctx_t xfer = {0};
xfer.cmd_type = xpi_apb_xfer_type_read;
xfer.addr = 0x1000;
xfer.seq_idx = 0;
xfer.seq_num = 1;
xfer.port = xpi_xfer_port_auto;
xfer.buf = &buffer[0];
xfer.xfer_size = sizeof(buffer);
status = ROM_API_TABLE_ROOT->xpi_driver_if->transfer_blocking(HPM_XPI0, &xfer);
```

18.7.4.8 software_reset

Perform a soft reset on the XPI module. Example:

```
ROM_API_TABLE_ROOT->xpi_driver_if->software_reset(HPM_XPI0);
```

18.7.4.9 is_idle

Check if the XPI module is idle. Example:

```
bool is_idle = ROM_API_TABLE_ROOT->xpi_driver_if->is_idle(HPM_XPI0);
```

18.7.4.10 update_dllcr

Update delay line settings. Example:

```
uint32_t serial_root_clk_freq = 133000000UL;
uint8_t data_valid_time = 100;
xpi_port_t port = xpi_port_a1;
uint8_t delay_target = 8;
```

```
ROM_API_TABLE_ROOT->xpi_driver_if->update_dllcr(HPM_XPI0, serial_root_clk_freq,
                                                data_valid_time, port, delay_target);
```

18.7.5 XPI NOR API

The XPI NOR API provides support for most FLASH devices on the market, allowing convenient FLASH read/write operations. Its API interface prototype is defined as follows:

```
typedef struct {
    uint32_t version;
    hpm_stat_t (*get_config)(XPI_Type *base,
                            xpi_nor_config_t *nor_cfg,
                            xpi_nor_config_option_t *cfg_option);
    hpm_stat_t (*init)(XPI_Type *base, xpi_nor_config_t *nor_config);
    hpm_stat_t (*enable_write)(XPI_Type *base,
                              xpi_xfer_port_t port,
                              const xpi_nor_config_t *nor_config, uint32_t addr);
    hpm_stat_t (*get_status)(XPI_Type *base, xpi_xfer_port_t port,
                            const xpi_nor_config_t *nor_config, uint32_t addr,
                            uint16_t *out_status);
    hpm_stat_t (*wait_busy)(XPI_Type *base, xpi_xfer_port_t port,
                            const xpi_nor_config_t *nor_config, uint32_t addr);
    hpm_stat_t (*erase)(XPI_Type *base, xpi_xfer_port_t port,
                       const xpi_nor_config_t *nor_config, uint32_t start,
                       uint32_t length);
    hpm_stat_t (*erase_chip)(XPI_Type *base, xpi_xfer_port_t port,
                             const xpi_nor_config_t *nor_config);
    hpm_stat_t (*erase_sector)(XPI_Type *base, xpi_xfer_port_t port,
                              const xpi_nor_config_t *nor_config, uint32_t addr);
    hpm_stat_t (*erase_block)(XPI_Type *base, xpi_xfer_port_t port,
                              const xpi_nor_config_t *nor_config, uint32_t addr);
    hpm_stat_t (*program)(XPI_Type *base, xpi_xfer_port_t port,
                         const xpi_nor_config_t *nor_config, const uint32_t *src,
                         uint32_t dst_addr, uint32_t length);
    hpm_stat_t (*read)(XPI_Type *base, xpi_xfer_port_t port,
                      const xpi_nor_config_t *nor_config, uint32_t *dst,
                      uint32_t start, uint32_t length);
    hpm_stat_t (*page_program_nonblocking)(XPI_Type *base, xpi_xfer_port_t port,
                                           const xpi_nor_config_t *nor_config, const uint32_t *src,
                                           uint32_t dst_addr, uint32_t length);
    hpm_stat_t (*erase_sector_nonblocking)(XPI_Type *base, xpi_xfer_port_t port,
                                           const xpi_nor_config_t *nor_config, uint32_t addr);
    hpm_stat_t (*erase_block_nonblocking)(XPI_Type *base, xpi_xfer_port_t port,
                                           const xpi_nor_config_t *nor_config, uint32_t addr);
```

```

hpm_stat_t (*erase_chip_nonblocking)(XPI_Type *base, xpi_xfer_port_t port,
                                     const xpi_nor_config_t *nor_config);
hpm_stat_t (*restore_spi_protocol)(XPI_Type *base, xpi_xfer_port_t port,
                                   xpi_nor_config_t *config, flash_run_context_t *run_ctx);
hpm_stat_t (*write_persistent)(const uint32_t data);
hpm_stat_t (*read_persistent)(uint32_t *data);
hpm_stat_t (*auto_config)(XPI_Type *base, xpi_nor_config_t *nor_cfg,
                          xpi_nor_config_option_t *cfg_option);
hpm_stat_t (*get_property)(XPI_Type *base, xpi_nor_config_t *nor_cfg,
                          uint32_t property_id, uint32_t *value);
hpm_stat_t (*set_property)(XPI_Type *base, xpi_nor_config_t *nor_cfg,
                          uint32_t property_id, uint32_t value);
} xpi_nor_driver_interface_t;

```

18.7.5.1 get_config

Get FLASH configuration based on FLASH configuration options. This API supports most 4-wire and 8-wire serial NOR FLASH on the market. Detailed definitions can be found in [XPI NOR Configuration Information](#).

Example:

```

xpi_nor_config_option_t flash_option = {0};
xpi_nor_config_t flash_config;
flash_option.header.U = 0xfc90001;
flash_option.option0.U = 0x00000007;

hpm_stat_t status = ROM_API_TABLE_ROOT->xpi_nor_driver_if->get_config(HPM_XPI0,
                              &flash_config, &flash_option);

```

18.7.5.2 init

Initialize XPI based on FLASH configuration. Example:

```

status = ROM_API_TABLE_ROOT->xpi_nor_driver_if->init(HPM_XPI0, &flash_config);

```

18.7.5.3 enable_write

Enable FLASH write operation. Example:

```

status = ROM_API_TABLE_ROOT->xpi_nor_driver_if->enable_write(HPM_XPI0,
                  xpi_xfer_port_auto, &flash_config);

```

18.7.5.4 get_status

Get current FLASH status. Example:

```

uint16_t flash_status;
status = ROM_API_TABLE_ROOT->xpi_nor_driver_if->get_status(HPM_XPI0,

```

```
xpi_xfer_port_auto , &flash_config , 0 , &flash_status );
```

18.7.5.5 wait_busy

FLASH busy wait operation. The API will block and wait while the FLASH is busy. Example:

```
status = ROM_API_TABLE_ROOT->xpi_nor_driver_if->wait_busy(HPM_XPI0,
    xpi_xfer_port_auto , &flash_config , 0);
```

18.7.5.6 erase

Blocking erase of the specified FLASH range. This command sequentially performs FLASH erase and busy wait operations. Example:

```
status = ROM_API_TABLE_ROOT->xpi_nor_driver_if->erase(HPM_XPI0,
    xpi_xfer_port_auto , &flash_config , 0, 0x2000);
```

18.7.5.7 erase_chip

Blocking erase of the entire FLASH. This command sequentially performs FLASH erase and busy wait operations. Example:

```
status = ROM_API_TABLE_ROOT->xpi_nor_driver_if->erase_chip(HPM_XPI0,
    xpi_xfer_port_auto , &flash_config );
```

18.7.5.8 erase_sector

Blocking erase of the specified FLASH sector. This command sequentially performs FLASH erase and busy wait operations. Example:

```
status = ROM_API_TABLE_ROOT->xpi_nor_driver_if->erase_sector(HPM_XPI0,
    xpi_xfer_port_auto , &flash_config , 0x4000);
```

18.7.5.9 erase_block

Blocking erase of the specified FLASH block. This command sequentially performs FLASH erase and busy wait operations. Example:

```
status = ROM_API_TABLE_ROOT->xpi_nor_driver_if->erase_block(HPM_XPI0,
    xpi_xfer_port_auto , &flash_config , 0x4000);
```

18.7.5.10 program

Blocking program data to the specified FLASH area. This command sequentially performs FLASH write operation and busy wait operation. Example:

```
uint32_t buffer[64];
uint8_t *buf_8 = (uint8_t*)&buffer[0];
```

```
for(uint32_t i=0; i<256; i++) {
    buf_8[i] = (uint8_t)i & 0xFF;
}
status = ROM_API_TABLE_ROOT->xpi_nor_driver_if->program(HPM_XPI0,
    xpi_xfer_port_auto, &flash_config, buffer, 0, sizeof(buffer));
```

18.7.5.11 read

Blocking read of the specified FLASH area. Example:

```
uint32_t buffer[64];
status = ROM_API_TABLE_ROOT->xpi_nor_driver_if->read(HPM_XPI0,
    xpi_xfer_port_auto, &flash_config, buffer, 0, sizeof(buffer));
```

18.7.5.12 page_program_nonblocking

Non-blocking program data to the specified FLASH page. This API is only responsible for sending the PAGE Program command; the user program needs to complete the subsequent busy wait operation. Example:

```
uint32_t buffer[64];
uint8_t *buf_8 = (uint8_t*)&buffer[0];
for(uint32_t i=0; i<256; i++) {
    buf_8[i] = (uint8_t)i & 0xFF;
}
status = ROM_API_TABLE_ROOT->xpi_nor_driver_if->page_program_nonblocking(HPM_XPI0,
    xpi_xfer_port_auto, &flash_config, buffer, 0, sizeof(buffer));
```

18.7.5.13 erase_sector_nonblocking

Non-blocking erase of the specified FLASH sector. This API is only responsible for sending the Sector Erase command; the user program needs to complete the subsequent busy wait operation. Example:

```
status = ROM_API_TABLE_ROOT->xpi_nor_driver_if->erase_sector_nonblocking(HPM_XPI0,
    xpi_xfer_port_auto, &flash_config, 0x4000);
```

18.7.5.14 erase_block_nonblocking

Non-blocking erase of the specified FLASH block. This API is only responsible for sending the Block Erase command; the user program needs to complete the subsequent busy wait operation. Example:

```
status = ROM_API_TABLE_ROOT->xpi_nor_driver_if->erase_block_nonblocking(HPM_XPI0,
    xpi_xfer_port_auto, &flash_config, 0x4000);
```

18.7.5.15 erase_chip_nonblocking

Non-blocking erase of the entire FLASH space. This API is only responsible for sending the Chip Erase command; the user program needs to complete the subsequent busy wait operation. Example:

```
status = ROM_API_TABLE_ROOT->xpi_nor_driver_if->erase_chip_nonblocking(HPM_XPI0,
    xpi_xfer_port_auto, &flash_config);
```

18.7.6 XPI RAM API

The XPI RAM API provides support for common HyperRAM, Octal RAM, and QPI RAM on the market. Its API interface prototype is defined as follows:

```
typedef struct {
    uint32_t version;
    hpm_stat_t (*get_config)(XPI_Type *base,
        xpi_ram_config_t *ram_cfg,
        xpi_ram_config_option_t *cfg_option);
    hpm_stat_t (*init)(XPI_Type *base, xpi_ram_config_t *ram_cfg);
} xpi_ram_driver_interface_t;
```

18.7.6.1 get_config

Get RAM configuration parameters based on RAM configuration options. Example:

```
xpi_ram_config_t ram_config;
xpi_ram_config_option_t option = {0};

option.header.tag = XPI_RAM_CFG_OPTION_TAG;
option.header.words = 2;
option.option0.freq_opt = 8;
option.option0.probe_type = xpi_ram_type_apmemory_x8; //APS6408L-OBM
option.option0.misc = 1; /* 1.8V */
option.option1.port = xpi_port_a2;

hpm_stat_t status = ROM_API_TABLE_ROOT->xpi_ram_driver_if->get_config(HPM_XPI1,
    &ram_config, &option);
```

18.7.6.2 init

Configure XPI based on RAM configuration information. Example:

```
status = ROM_API_TABLE_ROOT->xpi_ram_driver_if->init(HPM_XPI1, &ram_config);
```

18.7.7 Secure Boot API

The Secure Boot API provides security authentication support for data compatible with BootROM boot images. Its API interface prototype is defined as follows:

```
typedef struct {
    uint32_t version;
    secure_status_t (*init)(rom_core_ctx_t *ctx, uint32_t core_id);
```



```
secure_status_t (*auth_img_hdr)(rom_core_ctx_t *ctx ,
                                const boot_image_hdr_t *img_hdr ,
                                const uint8_t *blob ,
                                const uint8_t *srk_hash);
secure_status_t (*auth_fw)(rom_core_ctx_t *ctx ,
                           const firmware_info_t *fw_info);
secure_status_t (*hdl_img)(rom_core_ctx_t *ctx ,
                           const image_buf_t *img_hdr_buf ,
                           const boot_interface_t *boot_if ,
                           uint32_t img_offset);
} sec_boot_api_interface_t;
```

Detailed structure definitions can be found in the SDK examples.

18.8 SoC Specific Information

18.8.1 OTP Mapping Table

OTP Word Index	Bit Off-set	Field Width (bits)	Field	Description
1	0	4	LIFE_CYCLE_A	Life Cycle
	8	8	PUBK_REVOKE	Public Key REVOKE Lower 4 bits valid, bit_n represents revoking the n th public key
	28	4	LIFE_CYCLE_B	Duplicate Life Cycle Actual life cycle value is LIFE_CYCLE_A LIFE_CYCLE_B
3	0	32	SW_VER	Minimum software version number
24	0	4	XPI_FREQ_OPTION	XPI Frequency Option See details in XPI Frequency Options
	4	1	XPI_INSTANCE	XPI Instance 0 - XPI0
	5	1	XPI_PIN_GROUP_BIT0	XPI Pin Group 0 - First group 1 - Second group 2 - Third group
	6	2	XPI_PORT_SEL	XPI Port Selection 0 - XPI_CA (supports CS0 only) 1 - XPI_CB (supports CS0 only)

OTP Word Index	Bit Off-set	Field Width (bits)	Field	Description
	8	4	PROBE_TYPE	Serial NOR Probe Type 0 - Based on SFDP (SDR mode) 1 - Based on SFDP (DDR/DTR mode) 2 - 1-4-4 Read (Command 0xEB, 24-bit address) 3 - 1-2-2 Read (Command 0xBB, 24-bit address) 4 - HyperFLASH (1.8V) 5 - HyperFLASH (3.0V) 6 - OctaBus DTR mode 8 - Xccella DDR mode Others - Reserved
	12	1	ENCRYPT_XIP	Encrypted Execute-In-Place Enable 0 - Disable encrypted execute-in-place 1 - Enable encrypted execute-in-place
	13	1	XPI_NOR_CFG_SRC	Select XPI NOR Configuration Option Location 0 - Find XPI NOR CFG OPTION at FLASH offset 0x400 1 - Find XPI NOR CFG OPTION in OTP
	14	2	XPI_DEFAULT_READ	XPI Default Read Mode 0 - 1-1-1 Read, command 0x03, 24-bit address 1 - HyperBUS Read 2 - Xccella OPI DDR Read 3 - OctaBUS OPI DDR Read
	16	4	BOOT_MODE	Boot Mode Selection 0 - Boot mode determined by BOOT_MODE pins • 0 - Boot from XPI NOR • 1 - In-System Programming mode • 2 - In-System Programming mode • Others - Reserved 1 - Boot from XPI NOR 2 - In-System Programming mode Others - Reserved
	20	4	DRIVE_STRENGTH	IO Drive Strength Selection 0 - BootROM default configuration Others - See DS field definition in IOC's PAD_CTL
	24	7	DUMMY_CYCLE	Serial NOR FLASH Dummy Cycle Value 0 - Dummy cycle value automatically detected by ROM/use ROM preset value Others - Use dummy cycle value defined by this field

OTP Word Index	Bit Off-set	Field Width (bits)	Field	Description
25	0	16	SEC_IMG_OFFSET	Second boot image address offset 0 - Second boot image does not exist Others - $SEC\_IMG\_OFFSET \times 256KB$
	16	16	MAX_IMG_LEN	Maximum boot image length 0 - $SEC\_IMG\_OFFSET \times 256KB$ Others - $MAX\_IMG\_LEN \times 256KB$
26	0	2	FORCE_COLD_BOOT	Force Cold Boot 0 - Support low power wake-up Others - Force cold boot
	2	2	FORCE_WAKEUP_ENTRY_CHK	Force Wake-up Entry Integrity Check 0 - No need to check wake-up entry integrity 1 - Force wake-up entry integrity check (based on SHA256)
	4	1	FAIL_SAFE_BOOT	Low Speed Boot Mode 0 - Boot mode depends on HIGHSPEED_BOOT 1 - Low speed boot mode (CPU CORE frequency 250MHz)
	16	1	HIGHSPEED_BOOT	When FAIL_SAFE_BOOT is 0 0 - Normal boot mode (CPU CORE frequency 500MHz) 1 - High speed boot mode (CPU CORE frequency 600MHz)
	20	1	XPI_PIN_GROUP_BIT1	Combined with XPI_PIN_GROUP_BIT0 to form XPI pin group selection
27	0	32	Reserved	-
80	0	32	SRK_HASH[31:0]	SRK_HASH stored in OTP in little-endian format If SRK_HASH in byte form is (b0, b1, b2, b3, ..., b30, b31), stored in OTP as: 0xb3b2b1b0, 0xb7b6b5b4 ..., 0xb31b30b29b28
81	0	32	SRK_HASH[63:32]	
82	0	32	SRK_HASH[95:64]	
83	0	32	SRK_HASH[127:96]	
84	0	32	SRK_HASH[159:128]	
85	0	32	SRK_HASH[191:160]	
86	0	32	SRK_HASH[223:192]	
87	0	32	SRK_HASH[255:224]	
96	0	32	EXIP0_KEK[31:0]	KEK stored in OTP in big-endian format. If KEK in byte form is (b0, b1, b2, ..., b14, b15) stored in OTP as: 0xb15b14b13b12, ..., 0xb3b2b1b0
97	0	32	EXIP0_KEK[63:32]	
98	0	32	EXIP0_KEK[95:64]	
99	0	32	EXIP0_KEK[127:96]	

Table 83: BootROM Related OTP Mapping Table

18.8.2 BootROM Supported Peripheral Information

18.8.2.1 XPI Information

XPI0 Pin Groups

Pin Group	Pin	Function	Description
1	PB30	XPI0.CA_CS0	When FLASH device supports SPI mode, XPI accesses FLASH by default through bold pins
	PB24	XPI0.CA_CS1	
	PB29	XPI0.CA_SCLK	
	PB27	XPI0.CA_D[0]	
	PB28	XPI0.CA_D[1]	
	PB26	XPI0.CA_D[2]	
	PB31	XPI0.CA_D[3]	
	PB25	XPI0.CA_DQS	
	PC00	XPI0.CB_CS0	
	PC09	XPI0.CB_CS1	
	PC04	XPI0.CB_SCLK / XPI0.CA_SCLK_B	
	PC10	XPI0.CB_D[0] / XPI0.CA_D[4]	
	PC01	XPI0.CB_D[1] / XPI0.CA_D[5]	
	PC02	XPI0.CB_D[2] / XPI0.CA_D[6]	
	PC03	XPI0.CB_D[3] / XPI0.CA_D[7]	
	PC11	XPI0.CB_DQS	
2	PX05	XPI0.CA_CS0	-
	PX02	XPI0.CA_SCLK	
	PX01	XPI0.CA_D[0]	
	PX06	XPI0.CA_D[1]	
	PX07	XPI0.CA_D[2]	
	PX03	XPI0.CA_D[3]	
	PX04	XPI0.CA_DQS	
3	PC05	XPI0.CA_CS0	-
	PC08	XPI0.CA_SCLK	
	PC15	XPI0.CA_D[0]	
	PC06	XPI0.CA_D[1]	
	PC12	XPI0.CA_D[2]	
	PC07	XPI0.CA_D[3]	
	PC16	XPI0.CA_DQS	

Table 84: XPI0 Pin Information

Note:

- When XPI0 CA pin group is connected to FLASH that does not support DQS pin, CA_DQS pin can be used for other purposes
- XPI pins in the second group are only valid for SiP packaged chips

XPI Frequency Options

XPI NOR frequencies supported by BootROM:

- 1 - 30.3MHz (Note: In DDR mode, this value is 28.7MHz)
- 2 - 50MHz
- 3 - 66MHz
- 4 - 80MHz (Note: In DDR mode, this value is 83MHz)
- 5 - 100MHz
- 6 - 125MHz
- 7 - 133MHz
- 8 - 166MHz

Note: The above frequencies are all based on the SoC's default PRESET clock divider. If the corresponding clock source (such as PLL1CLK1) frequency changes, the above frequencies will change. Users need to confirm in advance that changes to the PLL clock affect the XPI frequency.

18.8.2.2 UART Pins

Chip Pin	Function	Function Description
PA00	UART0.TXD	UART0 pins
PA01	UART0.RXD	

Table 85: UART0 Pins

18.8.2.3 USB Pins

Chip Pin	Function	Function Description
USB0.DP	USB_DP	USB0 pins
USB0.DM	USB_DM	

Table 86: USB0 Pins

18.8.2.4 Boot Mode (BOOT_MODE) Pins

Chip Pin	Function	Function Description
PA02	BOOT_MODE[0]	Boot mode pins
PA03	BOOT_MODE[1]	

Table 87: BOOT_MODE Pins

18.8.3 BootROM Pre-occupied Register Information

Register	Function	Description/Notes
SYSCTL.CPU0.GPR0	CPU0 Wake-up Entry	CPU0 wake-up program entry

SYSCTL.CPU0.PARAM0	CPU0 Wake-up Code Start Address	CPU0 wake-up program start address (for integrity check)
SYSCTL.CPU0.PARAM1	CPU0 Wake-up Code Length	CPU0 wake-up program length (for integrity check)
SYSCTL.CPU0_DATA0 - SYSCTL.CPU0_DATA7	CPU0 Wake-up Code Hash Value	After FORCE_WAKEUP_ENTRY_CHK is set to 1, BootROM needs to calculate SHA256 based on the address range specified by SYSCTL.CPU0_PARAM0 and SYSCTL.CPU0_PARAM1, and compare with the value in PMIC.GPR4-PMIC.GPR11. Only when the results match, jump is allowed, otherwise perform complete boot process
SYSCTL.CPU0_LOCK	Lock register for the above registers	-
SYSCTL.CPU1_GPR0	CPU1 Boot Program Entry	-
SYSCTL.CPU1_GPR1	CPU1 Boot Program Entry Valid Flag	Only when this register value is 0xc1bef1a9, the CPU1 boot program entry is valid
PMIC.GPR0	run_bootloader parameter	See details in ROM API Introduction
PMIC.GPR1	XPI NOR FLASH Status Context	Save XPI NOR running context, detailed definition see
PMIC.GPR2	ROM Specific Marker	-

Table 88: BootROM Occupied Registers

18.8.4 General Register Configuration Supported Register Range

The following modules support configuration in General Register Information:

- SYSCTL
- IOC
- PPI
- XPI0

18.8.5 BootROM Memory Mapping Table

Address Range	Function
0x2000_0000 - 0x2001_FFFF	BootROM Code and Data
0x0023_7000 - 0x0023_FFFF	BSS, RW, Stack, etc.

Table 89: BootROM Memory Mapping

18.8.6 BootROM Lifecycle

Code	Lifecycle	Description
4'b0000	CREATE	In this mode, BootROM does not control DEBUG interface
4'b0001	NONSEC	In this mode, DEBUG interface is managed by BootROM, BootROM supports booting both unsigned and signed boot images
4'b0011	SECURE	In this mode, DEBUG interface is managed by BootROM, BootROM only supports booting signed boot images
4'b0111	RETURN	In this mode, chip security-related information is disabled, BootROM can boot unsigned boot images
4'b1011	NONRET	This mode behaves the same as CLOSED/DEPLOY mode, the only difference is that OEM vendors can only switch to SCRIBE mode in this mode
4'b1111	SCRIBE	Invalid mode, BootROM cannot perform normal boot operations in this mode

Table 90: BootROM Lifecycle Codes

19 Pin Configuration and Function PINMUX

19.1 IO function allocation

The pin configuration and functions of this product are as follows:

Package			PIN NAME	Digital Function	Analog Function	Power Group
BGA _196	BGA _121	ELQ FP_1 00				
G2	D1	11	PA00	GPIO_A_00(ALT0) GPTMR1_COMP_0(ALT1) UART0_TXD(ALT2) MCAN0_TXD(ALT7) PWM0_P_0(ALT16) TRGM_P_00(ALT17) SYSCTL_CLK_OBS_0(ALT 24)	-	B00
G3	C1	9	PA01	GPIO_A_01(ALT0) GPTMR1_CAPT_0(ALT1) UART0_RXD(ALT2) MCAN0_RXD(ALT7) PWM0_P_1(ALT16) TRGM_P_01(ALT17) SYSCTL_CLK_OBS_2(ALT 24)	-	B00
G4	C2	7	PA02	GPIO_A_02(ALT0) GPTMR1_COMP_1(ALT1) UART0_DE(ALT2) UART0_RTS(ALT3) I2C2_SCL(ALT4) MCAN0_STBY(ALT7) PWM0_P_2(ALT16) TRGM_P_02(ALT17) SYSCTL_CLK_OBS_1(ALT 24)	-	B00

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

19 PIN CONFIGURATION AND FUNCTION

PINMUX

Package			PIN NAME	Digital Function	Analog Function	Power Group
BGA _196	BGA _121	ELQ FP_1 00				
G5	B1	6	PA03	GPIO_A_03(ALT0) GPTMR1_CAPT_1(ALT1) UART0_CTS(ALT3) I2C2_SDA(ALT4) MCAN1_STBY(ALT7) PWM0_P_3(ALT16) TRGM_P_03(ALT17) SYSCTL_CLK_OBS_3(ALT24)	-	B00
F2	B2	3	PA04	GPIO_A_04(ALT0) UART1_CTS(ALT3) SPI1_SCLK(ALT5) MCAN1_RXD(ALT7) PWM0_P_4(ALT16) TRGM_P_04(ALT17) JTAG_TDO(ALT24)	-	B00
F3	B3	2	PA05	GPIO_A_05(ALT0) GPTMR1_COMP_2(ALT1) UART1_DE(ALT2) UART1_RTS(ALT3) SPI1_CS_0(ALT5) MCAN1_TXD(ALT7) PWM0_P_5(ALT16) TRGM_P_05(ALT17) JTAG_TDI(ALT24)	-	B00
F4	B4	1	PA06	GPIO_A_06(ALT0) GPTMR0_CAPT_0(ALT1) UART1_RXD(ALT2) I2C3_SDA(ALT4) SPI1_MISO(ALT5) PWM0_P_6(ALT16) TRGM_P_06(ALT17) JTAG_TCK(ALT24)	-	B00

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

19 PIN CONFIGURATION AND FUNCTION

PINMUX

Package			PIN NAME	Digital Function	Analog Function	Power Group
BGA _196	BGA _121	ELQ FP_1 00				
F5	A2	100	PA07	GPIO_A_07(ALT0) GPTMR0_COMP_0(ALT1) UART1_TXD(ALT2) I2C3_SCL(ALT4) SPI1_MOSI(ALT5) PWM0_P_7(ALT16) TRGM_P_07(ALT17) JTAG_TMS(ALT24)	-	B00
G1	A3	8	PA08	GPIO_A_08(ALT0) GPTMR0_COMP_1(ALT1) UART2_TXD(ALT2) I2C0_SCL(ALT4) MCAN2_TXD(ALT7) PWM1_P_0(ALT16) TRGM_P_08(ALT17) JTAG_TRST(ALT24)	-	B00
F1	A4	5	PA09	GPIO_A_09(ALT0) GPTMR0_CAPT_1(ALT1) UART2_RXD(ALT2) I2C0_SDA(ALT4) MCAN2_RXD(ALT7) ESC0_REFCK(ALT11) PWM1_P_1(ALT16) TRGM_P_09(ALT17)	-	B00
L1	K1	28	PA16	GPIO_A_16(ALT0) GPTMR3_COMP_0(ALT1) UART4_TXD(ALT2) ESC0_P0_RXDV(ALT11) QEO1_A(ALT21) SDM0_DAT_3(ALT23) EUI0_CK(ALT26)	-	B00
L2	J1	27	PA17	GPIO_A_17(ALT0) GPTMR3_CAPT_0(ALT1) UART4_RXD(ALT2) ESC0_P0_RXD_0(ALT11) QEO1_B(ALT21) SDM0_CLK_3(ALT23) EUI0_SH(ALT26)	-	B00

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

19 PIN CONFIGURATION AND FUNCTION

PINMUX

Package			PIN NAME	Digital Function	Analog Function	Power Group
BGA _196	BGA _121	ELQ FP_1 00				
K1	J2	26	PA18	GPIO_A_18(ALT0) GPTMR3_COMP_1(ALT1) UART4_DE(ALT2) UART4_RTS(ALT3) ESC0_P0_RXD_1(ALT11) QEO1_Z(ALT21) SDM0_DAT_2(ALT23) EUI0_DI(ALT26)	-	B00
K2	H1	25	PA19	GPIO_A_19(ALT0) GPTMR3_CAPT_1(ALT1) UART4_CTS(ALT3) SPI3_CS_3(ALT5) ESC0_P0_RXD_2(ALT11) SDM0_CLK_2(ALT23) EUI0_DO(ALT26)	-	B00
J1	H2	21	PA20	GPIO_A_20(ALT0) UART5_CTS(ALT3) SPI2_SCLK(ALT5) ESC0_P0_RXD_3(ALT11) SDM0_DAT_1(ALT23)	-	B00
J2	H3	20	PA21	GPIO_A_21(ALT0) GPTMR3_COMP_2(ALT1) UART5_DE(ALT2) UART5_RTS(ALT3) SPI2_CS_0(ALT5) ESC0_P0_RXCK(ALT11) QEO0_Z(ALT21) SDM0_CLK_1(ALT23) OWR0_DAT(ALT26)	-	B00
H1	G1	10	PA22	GPIO_A_22(ALT0) GPTMR2_CAPT_0(ALT1) UART5_RXD(ALT2) SPI2_MISO(ALT5) ETH0_TXER(ALT18) QEO0_B(ALT21) SDM0_DAT_0(ALT23) OWR1_DAT(ALT26)	-	B00

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

19 PIN CONFIGURATION AND FUNCTION

PINMUX

Package			PIN NAME	Digital Function	Analog Function	Power Group
BGA _196	BGA _121	ELQ FP_1 00				
H2	G2	15	PA23	GPIO_A_23(ALT0) GPTMR2_COMP_0(ALT1) UART5_TXD(ALT2) SPI2_MOSI(ALT5) ESC0_P0_RXER(ALT11) ETH0_RXER(ALT18) QEO0_A(ALT21) SDM0_CLK_0(ALT23)	-	B00
K3	G3	24	PA24	GPIO_A_24(ALT0) GPTMR2_COMP_1(ALT1) UART6_TXD(ALT2) SPI3_CS_2(ALT5) ESC0_P0_TXCK(ALT11)	-	B00
K4	F1	22	PA25	GPIO_A_25(ALT0) GPTMR2_CAPT_1(ALT1) UART6_RXD(ALT2) SPI3_CS_1(ALT5) ESC0_P0_TXD_0(ALT11) ESC0_CTR_0(ALT18)	-	B00
J3	F2	18	PA26	GPIO_A_26(ALT0) GPTMR2_COMP_2(ALT1) UART6_DE(ALT2) UART6_RTS(ALT3) SPI3_SCLK(ALT5) ESC0_P0_TXD_1(ALT11) QE11_H1(ALT20) EWDG0_RST(ALT24)	-	B00
J4	E1	17	PA27	GPIO_A_27(ALT0) UART6_CTS(ALT3) SPI3_CS_0(ALT5) ESC0_P0_TXD_2(ALT11) QE11_F(ALT20)	-	B00
J5	E2	16	PA28	GPIO_A_28(ALT0) UART7_CTS(ALT3) SPI3_MISO(ALT5) ESC0_P0_TXD_3(ALT11) ESC0_CTR_1(ALT18) QE11_H0(ALT20)	-	B00

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

19 PIN CONFIGURATION AND FUNCTION

PINMUX

Package			PIN NAME	Digital Function	Analog Function	Power Group
BGA _196	BGA _121	ELQ FP_1 00				
H3	E3	14	PA29	GPIO_A_29(ALT0) GPTMR3_COMP_3(ALT1) UART7_DE(ALT2) UART7_RTS(ALT3) SPI3_MOSI(ALT5) ESC0_P0_TXEN(ALT11) QE11_Z(ALT20)	-	B00
H4	D2	13	PA30	GPIO_A_30(ALT0) UART7_RXD(ALT2) SPI3_DAT2(ALT5) ESC0_MDIO(ALT11) ETH0_MDIO(ALT18) QE11_A(ALT20)	-	B00
H5	D3	12	PA31	GPIO_A_31(ALT0) GPTMR2_COMP_3(ALT1) UART7_TXD(ALT2) SPI3_DAT3(ALT5) ESC0_MDC(ALT11) ETH0_MDC(ALT18) QE11_B(ALT20)	-	B00
E1	-	-	PB00	GPIO_B_00(ALT0) MCAN0_TXD(ALT7) ESC0_P0_RXDV(ALT11) PWM0_P_0(ALT16) TRGM_P_00(ALT17) ETH0_RXDV(ALT18) QEO1_A(ALT21)	-	B00
E2	-	-	PB01	GPIO_B_01(ALT0) MCAN0_RXD(ALT7) ESC0_P0_RXD_0(ALT11) PWM0_P_1(ALT16) TRGM_P_01(ALT17) ETH0_RXD_0(ALT18) QEO1_B(ALT21)	-	B00

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

19 PIN CONFIGURATION AND FUNCTION

PINMUX

Package			PIN NAME	Digital Function	Analog Function	Power Group
BGA _196	BGA _121	ELQ FP_1 00				
D1	-	-	PB02	GPIO_B_02(ALT0) I2C2_SCL(ALT4) MCAN0_STBY(ALT7) ESC0_P0_RXD_1(ALT11) PWM0_P_2(ALT16) TRGM_P_02(ALT17) ETH0_RXD_1(ALT18) QEIO_H1(ALT20) QEO1_Z(ALT21)	-	B00
D2	-	-	PB03	GPIO_B_03(ALT0) I2C2_SDA(ALT4) MCAN1_STBY(ALT7) ESC0_P0_RXD_2(ALT11) PWM0_P_3(ALT16) TRGM_P_03(ALT17) ETH0_RXD_2(ALT18) QEIO_F(ALT20)	-	B00
C1	-	-	PB04	GPIO_B_04(ALT0) MCAN1_RXD(ALT7) ESC0_P0_RXD_3(ALT11) PWM0_P_4(ALT16) TRGM_P_04(ALT17) ETH0_RXD_3(ALT18) QEIO_H0(ALT20)	-	B00
C2	-	-	PB05	GPIO_B_05(ALT0) MCAN1_TXD(ALT7) ESC0_P0_RXCK(ALT11) PWM0_P_5(ALT16) TRGM_P_05(ALT17) ETH0_RXCK(ALT18) QEIO_Z(ALT20) QEO0_Z(ALT21)	-	B00

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

19 PIN CONFIGURATION AND FUNCTION

PINMUX

Package			PIN NAME	Digital Function	Analog Function	Power Group
BGA _196	BGA _121	ELQ FP_1 00				
E3	-	-	PB06	GPIO_B_06(ALT0) I2C3_SDA(ALT4) ESC0_P0_TXCK(ALT11) PWM0_P_6(ALT16) TRGM_P_06(ALT17) ETH0_TXCK(ALT18) QEIO_B(ALT20) QEO0_B(ALT21)	-	B00
E4	-	-	PB07	GPIO_B_07(ALT0) I2C3_SCL(ALT4) ESC0_P0_TXD_0(ALT11) PWM0_P_7(ALT16) TRGM_P_07(ALT17) ETH0_TXD_0(ALT18) QEIO_A(ALT20) QEO0_A(ALT21)	-	B00
D3	-	-	PB08	GPIO_B_08(ALT0) I2C0_SCL(ALT4) MCAN2_TXD(ALT7) ESC0_P0_TXD_1(ALT11) PWM1_P_0(ALT16) TRGM_P_08(ALT17) ETH0_TXD_1(ALT18)	-	B00
D4	-	-	PB09	GPIO_B_09(ALT0) I2C0_SDA(ALT4) MCAN2_RXD(ALT7) ESC0_P0_TXD_2(ALT11) PWM1_P_1(ALT16) TRGM_P_09(ALT17) ETH0_TXD_2(ALT18)	-	B00
C3	-	-	PB10	GPIO_B_10(ALT0) MCAN2_STBY(ALT7) ESC0_P0_TXD_3(ALT11) PWM1_P_2(ALT16) TRGM_P_10(ALT17) ETH0_TXD_3(ALT18)	-	B00

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

19 PIN CONFIGURATION AND FUNCTION

PINMUX

Package			PIN NAME	Digital Function	Analog Function	Power Group
BGA _196	BGA _121	ELQ FP_1 00				
C4	-	-	PB11	GPIO_B_11(ALT0) ESC0_P0_TXEN(ALT11) PWM1_P_3(ALT16) TRGM_P_11(ALT17) ETH0_TXEN(ALT18)	-	B00
A2	-	-	PB24	GPIO_B_24(ALT0) ESC0_CTR_4(ALT11) XPI0_CA_CS1(ALT14) SDM0_CLK_3(ALT23) ETH0_EVTI_0(ALT25) ESC0_EVTI_0(ALT27)	-	B00
B1	-	-	PB25	GPIO_B_25(ALT0) ESC0_CTR_5(ALT11) XPI0_CA_DQS(ALT14) SDM0_DAT_3(ALT23) ETH0_EVTO_0(ALT25) ESC0_EVTO_0(ALT27)	-	B00
A3	C4	97	PB26	GPIO_B_26(ALT0) ESC0_CTR_6(ALT11) XPI0_CA_D_2(ALT14) QE11_H1(ALT20) SDM0_CLK_2(ALT23) ETH0_EVTI_1(ALT25) EUI1_CK(ALT26) ESC0_EVTI_1(ALT27)	-	B00
B2	C5	96	PB27	GPIO_B_27(ALT0) ESC0_CTR_7(ALT11) XPI0_CA_D_0(ALT14) QE11_F(ALT20) SDM0_DAT_2(ALT23) EUI1_SH(ALT26)	-	B00
A4	B5	95	PB28	GPIO_B_28(ALT0) ESC0_CTR_8(ALT11) XPI0_CA_D_1(ALT14) QE11_H0(ALT20) SDM0_CLK_1(ALT23) EUI1_DI(ALT26)	-	B00

Package			PIN NAME	Digital Function	Analog Function	Power Group
BGA _196	BGA _121	ELQ FP_1 00				
B3	B6	94	PB29	GPIO_B_29(ALT0) XPI0_CA_SCLK(ALT14) QE11_Z(ALT20) SDM0_DAT_1(ALT23) ETH0_EVTO_1(ALT25) EUI1_DO(ALT26) ESC0_EVTO_1(ALT27)	-	B00
A5	A5	93	PB30	GPIO_B_30(ALT0) ESC0_CTR_7(ALT11) XPI0_CA_CS0(ALT14) QE11_A(ALT20) SDM0_CLK_0(ALT23)	-	B00
B4	A6	92	PB31	GPIO_B_31(ALT0) ESC0_CTR_8(ALT11) XPI0_CA_D_3(ALT14) QE11_B(ALT20) SDM0_DAT_0(ALT23)	-	B00
B5	-	-	PC00	GPIO_C_00(ALT0) GPTMR1_COMP_0(ALT1) UART0_TXD(ALT2) MCAN0_TXD(ALT7) ESC0_GPI_63(ALT10) ESC0_GPO_00(ALT11) PPI0_DQ_24(ALT13) XPI0_CB_CS0(ALT14) PWM0_P_0(ALT16) TRGM_P_00(ALT17) ESC0_GPI_15(ALT18) OWR0_DAT(ALT26)	-	B00

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

19 PIN CONFIGURATION AND FUNCTION

PINMUX

Package			PIN NAME	Digital Function	Analog Function	Power Group
BGA _196	BGA _121	ELQ FP_1 00				
A6	-	-	PC01	GPIO_C_01(ALT0) GPTMR1_CAPT_0(ALT1) UART0_RXD(ALT2) MCAN0_RXD(ALT7) ESC0_GPI_62(ALT10) ESC0_GPO_01(ALT11) PPI0_DQ_25(ALT13) XPI0_CB_D_1(ALT14) PWM0_P_1(ALT16) TRGM_P_01(ALT17) ESC0_GPI_14(ALT18) OWR1_DAT(ALT26)	-	B00
B6	-	-	PC02	GPIO_C_02(ALT0) GPTMR1_COMP_1(ALT1) UART0_DE(ALT2) UART0_RTS(ALT3) I2C2_SCL(ALT4) MCAN0_STBY(ALT7) ESC0_GPI_61(ALT10) ESC0_GPO_02(ALT11) PPI0_DQ_22(ALT13) XPI0_CB_D_2(ALT14) PWM0_P_2(ALT16) TRGM_P_02(ALT17) ESC0_GPI_13(ALT18) QEIO_H1(ALT20)	-	B00

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

19 PIN CONFIGURATION AND FUNCTION

PINMUX

Package			PIN NAME	Digital Function	Analog Function	Power Group
BGA _196	BGA _121	ELQ FP_1 00				
A7	-	-	PC03	GPIO_C_03(ALT0) GPTMR1_CAPT_1(ALT1) UART0_CTS(ALT3) I2C2_SDA(ALT4) SPI1_CS_3(ALT5) MCAN1_STBY(ALT7) ESC0_GPI_60(ALT10) ESC0_GPO_03(ALT11) PPI0_DQ_26(ALT13) XPI0_CB_D_3(ALT14) PWM0_P_3(ALT16) TRGM_P_03(ALT17) ESC0_GPI_12(ALT18) QEIO_F(ALT20)	-	B00
B7	-	-	PC04	GPIO_C_04(ALT0) UART1_CTS(ALT3) SPI0_SCLK(ALT5) MCAN1_RXD(ALT7) ESC0_GPI_59(ALT10) ESC0_GPO_04(ALT11) PPI0_DQ_27(ALT13) XPI0_CB_SCLK(ALT14) PWM0_P_4(ALT16) TRGM_P_04(ALT17) ESC0_GPI_11(ALT18) QEIO_H0(ALT20)	-	B00

Package			PIN NAME	Digital Function	Analog Function	Power Group
BGA _196	BGA _121	ELQ FP_1 00				
A8	-	-	PC05	GPIO_C_05(ALT0) GPTMR1_COMP_2(ALT1) UART1_DE(ALT2) UART1_RTS(ALT3) SPI0_CS_0(ALT5) MCAN1_TXD(ALT7) ESC0_GPI_58(ALT10) ESC0_GPO_05(ALT11) PPI0_DQ_28(ALT13) XPI0_CA_CS0(ALT14) PWM0_P_5(ALT16) TRGM_P_05(ALT17) ESC0_GPI_10(ALT18) QEIO_Z(ALT20) QEO0_Z(ALT21)	-	B00
B8	-	-	PC06	GPIO_C_06(ALT0) GPTMR0_CAPT_0(ALT1) UART1_RXD(ALT2) I2C3_SDA(ALT4) SPI0_MISO(ALT5) ESC0_GPI_57(ALT10) ESC0_GPO_06(ALT11) PPI0_DQ_29(ALT13) XPI0_CA_D_1(ALT14) PWM0_P_6(ALT16) TRGM_P_06(ALT17) ESC0_GPI_09(ALT18) QEIO_B(ALT20) QEO0_B(ALT21)	-	B00

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

19 PIN CONFIGURATION AND FUNCTION

PINMUX

Package			PIN NAME	Digital Function	Analog Function	Power Group
BGA _196	BGA _121	ELQ FP_1 00				
A9	-	-	PC07	GPIO_C_07(ALT0) GPTMR0_COMP_0(ALT1) UART1_TXD(ALT2) I2C3_SCL(ALT4) SPI0_MOSI(ALT5) ESC0_GPI_56(ALT10) ESC0_GPO_07(ALT11) PPI0_DQ_30(ALT13) XPI0_CA_D_3(ALT14) PWM0_P_7(ALT16) TRGM_P_07(ALT17) ESC0_GPI_08(ALT18) QEIO_A(ALT20) QEO0_A(ALT21)	-	B00
B9	C7	91	PC08	GPIO_C_08(ALT0) GPTMR0_COMP_1(ALT1) UART2_TXD(ALT2) I2C0_SCL(ALT4) SPI1_CS_2(ALT5) MCAN2_TXD(ALT7) ESC0_GPI_55(ALT10) ESC0_GPO_08(ALT11) PPI0_DQ_31(ALT13) XPI0_CA_SCLK(ALT14) PWM1_P_0(ALT16) TRGM_P_08(ALT17) ESC0_GPI_07(ALT18)	-	B00

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

19 PIN CONFIGURATION AND FUNCTION

PINMUX

Package			PIN NAME	Digital Function	Analog Function	Power Group
BGA _196	BGA _121	ELQ FP_1 00				
C5	B7	90	PC09	GPIO_C_09(ALT0) GPTMR0_CAPT_1(ALT1) UART2_RXD(ALT2) I2C0_SDA(ALT4) SPI1_CS_1(ALT5) MCAN2_RXD(ALT7) ESC0_GPI_54(ALT10) ESC0_GPO_09(ALT11) PPI0_DQ_23(ALT13) XPI0_CB_CS1(ALT14) PWM1_P_1(ALT16) TRGM_P_09(ALT17) ESC0_GPI_06(ALT18) QE11_H1(ALT20) ESC0_CTR_0(ALT27)	-	B00
D5	A7	89	PC10	GPIO_C_10(ALT0) GPTMR0_COMP_2(ALT1) UART2_DE(ALT2) UART2_RTS(ALT3) SPI1_SCLK(ALT5) MCAN2_STBY(ALT7) ESC0_GPI_53(ALT10) ESC0_GPO_10(ALT11) PPI0_DQ_20(ALT13) XPI0_CB_D_0(ALT14) PWM1_P_2(ALT16) TRGM_P_10(ALT17) ESC0_GPI_05(ALT18) QE11_F(ALT20) ESC0_CTR_1(ALT27)	-	B00

Package			PIN NAME	Digital Function	Analog Function	Power Group
BGA _196	BGA _121	ELQ FP_1 00				
C6	C8	88	PC11	GPIO_C_11(ALT0) UART2_CTS(ALT3) SPI1_CS_0(ALT5) ESC0_GPI_52(ALT10) ESC0_GPO_11(ALT11) PPI0_DQ_21(ALT13) XPI0_CB_DQS(ALT14) PWM1_P_3(ALT16) TRGM_P_11(ALT17) ESC0_GPI_04(ALT18) QE11_H0(ALT20) ESC0_CTR_2(ALT27)	-	B00
D6	B8	87	PC12	GPIO_C_12(ALT0) UART3_CTS(ALT3) I2C1_SDA(ALT4) SPI1_MISO(ALT5) ESC0_GPI_51(ALT10) ESC0_GPO_12(ALT11) PPI0_DM_2(ALT13) XPI0_CA_D_2(ALT14) PWM1_P_4(ALT16) TRGM_P_12(ALT17) ESC0_GPI_03(ALT18) QE11_Z(ALT20) QEO1_Z(ALT21) ESC0_CTR_3(ALT27)	-	B00

Package			PIN NAME	Digital Function	Analog Function	Power Group
BGA _196	BGA _121	ELQ FP_1 00				
C7	A8	86	PC13	GPIO_C_13(ALT0) GPTMR1_COMP_3(ALT1) UART3_DE(ALT2) UART3_RTS(ALT3) I2C1_SCL(ALT4) SPI1_MOSI(ALT5) MCAN3_STBY(ALT7) ESC0_GPI_50(ALT10) ESC0_GPO_13(ALT11) PPI0_DQ_18(ALT13) XPI0_CA_CS1(ALT14) PWM1_P_5(ALT16) TRGM_P_13(ALT17) ESC0_GPI_02(ALT18) QE11_A(ALT20) QEO1_A(ALT21) ESC0_CTR_4(ALT27)	-	B00
D7	B9	85	PC14	GPIO_C_14(ALT0) UART3_RXD(ALT2) SPI1_DAT2(ALT5) MCAN3_RXD(ALT7) ESC0_GPI_49(ALT10) ESC0_GPO_14(ALT11) PPI0_DQ_19(ALT13) PWM1_P_6(ALT16) TRGM_P_14(ALT17) ESC0_GPI_01(ALT18) QE11_B(ALT20) QEO1_B(ALT21) ESC0_CTR_5(ALT27)	-	B00

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

19 PIN CONFIGURATION AND FUNCTION

PINMUX

Package			PIN NAME	Digital Function	Analog Function	Power Group
BGA _196	BGA _121	ELQ FP_1 00				
C8	A9	84	PC15	GPIO_C_15(ALT0) GPTMR0_COMP_3(ALT1) UART3_TXD(ALT2) SPI1_DAT3(ALT5) MCAN3_TXD(ALT7) ESC0_GPI_48(ALT10) ESC0_GPO_15(ALT11) PPI0_DM_3(ALT13) XPI0_CA_D_0(ALT14) PWM1_P_7(ALT16) TRGM_P_15(ALT17) ESC0_GPI_00(ALT18) ESC0_CTR_6(ALT27)	-	B00
D8	A10	83	PC16	GPIO_C_16(ALT0) GPTMR3_COMP_0(ALT1) UART4_TXD(ALT2) ESC0_GPI_47(ALT10) ESC0_GPO_16(ALT11) PPI0_DQ_16(ALT13) XPI0_CA_DQS(ALT14) ESC0_CTR_7(ALT27)	-	B00
C9	-	-	PC17	GPIO_C_17(ALT0) GPTMR3_CAPT_0(ALT1) UART4_RXD(ALT2) ESC0_GPI_46(ALT10) ESC0_GPO_17(ALT11) PPI0_DQ_17(ALT13) ESC0_CTR_8(ALT27)	-	B00
A10	-	-	PC18	GPIO_C_18(ALT0) GPTMR3_COMP_1(ALT1) UART4_DE(ALT2) UART4_RTS(ALT3) ESC0_GPI_45(ALT10) ESC0_GPO_18(ALT11) PPI0_DQ_04(ALT13) ESC0_CTR_0(ALT18)	-	B00

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

19 PIN CONFIGURATION AND FUNCTION

PINMUX

Package			PIN NAME	Digital Function	Analog Function	Power Group
BGA _196	BGA _121	ELQ FP_1 00				
B10	-	-	PC19	GPIO_C_19(ALT0) GPTMR3_CAPT_1(ALT1) UART4_CTS(ALT3) SPI2_CS_3(ALT5) ESC0_GPI_44(ALT10) ESC0_GPO_19(ALT11) PPI0_DQ_05(ALT13) ESC0_CTR_1(ALT18)	-	B00
A11	-	-	PC20	GPIO_C_20(ALT0) UART5_CTS(ALT3) SPI3_SCLK(ALT5) ESC0_GPI_43(ALT10) ESC0_GPO_20(ALT11) PPI0_DQ_06(ALT13) ESC0_CTR_2(ALT18)	-	B00
B11	-	-	PC21	GPIO_C_21(ALT0) GPTMR3_COMP_2(ALT1) UART5_DE(ALT2) UART5_RTS(ALT3) SPI3_CS_0(ALT5) ESC0_GPI_42(ALT10) ESC0_GPO_21(ALT11) PPI0_DQ_07(ALT13) ESC0_CTR_3(ALT18) QEO1_Z(ALT21)	-	B00
A12	-	-	PC22	GPIO_C_22(ALT0) GPTMR2_CAPT_0(ALT1) UART5_RXD(ALT2) SPI3_MISO(ALT5) ESC0_GPI_41(ALT10) ESC0_GPO_22(ALT11) PPI0_CTR_2(ALT13) QEO1_B(ALT21)	-	B00

Package			PIN NAME	Digital Function	Analog Function	Power Group
BGA _196	BGA _121	ELQ FP_1 00				
A13	-	-	PC23	GPIO_C_23(ALT0) GPTMR2_COMP_0(ALT1) UART5_TXD(ALT2) SPI3_MOSI(ALT5) ESC0_GPI_40(ALT10) ESC0_GPO_23(ALT11) PPI0_DQ_08(ALT13) QEO1_A(ALT21)	-	B00
B14	-	-	PC24	GPIO_C_24(ALT0) GPTMR2_COMP_1(ALT1) UART6_TXD(ALT2) SPI2_CS_2(ALT5) ESC0_GPI_39(ALT10) ESC0_GPO_24(ALT11) PPI0_DQ_09(ALT13)	-	B00
C14	-	-	PC25	GPIO_C_25(ALT0) GPTMR2_CAPT_1(ALT1) UART6_RXD(ALT2) SPI2_CS_1(ALT5) ESC0_GPI_38(ALT10) ESC0_GPO_25(ALT11) PPI0_DQ_11(ALT13)	-	B00
D14	-	-	PC26	GPIO_C_26(ALT0) GPTMR2_COMP_2(ALT1) UART6_DE(ALT2) UART6_RTS(ALT3) SPI2_SCLK(ALT5) ESC0_GPI_37(ALT10) ESC0_GPO_26(ALT11) PPI0_DQ_12(ALT13) QE11_H1(ALT20)	-	B00

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

19 PIN CONFIGURATION AND FUNCTION

PINMUX

Package			PIN NAME	Digital Function	Analog Function	Power Group
BGA _196	BGA _121	ELQ FP_1 00				
D13	-	-	PC27	GPIO_C_27(ALT0) UART6_CTS(ALT3) SPI2_CS_0(ALT5) ESC0_GPI_36(ALT10) ESC0_GPO_27(ALT11) PPI0_CTR_0(ALT13) ESC0_CTR_4(ALT18) QE11_F(ALT20)	-	B00
E14	-	-	PC28	GPIO_C_28(ALT0) UART7_CTS(ALT3) SPI2_MISO(ALT5) ESC0_GPI_35(ALT10) ESC0_GPO_28(ALT11) PPI0_DM_0(ALT13) ESC0_CTR_5(ALT18) QE11_H0(ALT20)	-	B00
E13	-	-	PC29	GPIO_C_29(ALT0) GPTMR3_COMP_3(ALT1) UART7_DE(ALT2) UART7_RTS(ALT3) SPI2_MOSI(ALT5) ESC0_GPI_34(ALT10) ESC0_GPO_29(ALT11) PPI0_CLK(ALT13) ESC0_CTR_6(ALT18) QE11_Z(ALT20) SOC_REF0(ALT24)	-	B00
F14	-	-	PC30	GPIO_C_30(ALT0) UART7_RXD(ALT2) SPI2_DAT2(ALT5) ESC0_GPI_33(ALT10) ESC0_GPO_30(ALT11) PPI0_CS_0(ALT13) ESC0_CTR_7(ALT18) QE11_A(ALT20) SOC_REF1(ALT24)	-	B00

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

19 PIN CONFIGURATION AND FUNCTION

PINMUX

Package			PIN NAME	Digital Function	Analog Function	Power Group
BGA _196	BGA _121	ELQ FP_1 00				
F13	-	-	PC31	GPIO_C_31(ALT0) GPTMR2_COMP_3(ALT1) UART7_TXD(ALT2) SPI2_DAT3(ALT5) ESC0_GPI_32(ALT10) ESC0_GPO_31(ALT11) PPI0_CS_1(ALT13) ESC0_CTR_8(ALT18) QE11_B(ALT20)	-	B00
D9	B10	81	PD00	GPIO_D_00(ALT0) MCAN0_TXD(ALT7) ESC0_GPI_31(ALT10) ESC0_GPO_32(ALT11) PPI0_DQ_02(ALT13) PWM0_P_0(ALT16) TRGM_P_00(ALT17) ESC0_REFCK(ALT18) ESC0_CTR_0(ALT27)	-	B00
C10	B11	80	PD01	GPIO_D_01(ALT0) MCAN0_RXD(ALT7) ESC0_GPI_30(ALT10) ESC0_GPO_33(ALT11) PPI0_DQ_03(ALT13) PWM0_P_1(ALT16) TRGM_P_01(ALT17) ESC0_REFCK(ALT18) ESC0_CTR_1(ALT27)	-	B00

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

19 PIN CONFIGURATION AND FUNCTION

PINMUX

Package			PIN NAME	Digital Function	Analog Function	Power Group
BGA _196	BGA _121	ELQ FP_1 00				
D10	C10	79	PD02	GPIO_D_02(ALT0) I2C2_SCL(ALT4) MCAN0_STBY(ALT7) ESC0_GPI_29(ALT10) ESC0_GPO_34(ALT11) PPI0_DQ_00(ALT13) PWM0_P_2(ALT16) TRGM_P_02(ALT17) ESC0_SCL(ALT18) QEIO_H1(ALT20) EWDG1_RST(ALT24) ESC0_CTR_2(ALT27)	-	B00
C11	C11	78	PD03	GPIO_D_03(ALT0) I2C2_SDA(ALT4) MCAN1_STBY(ALT7) ESC0_GPI_28(ALT10) ESC0_GPO_35(ALT11) PPI0_DQ_01(ALT13) PWM0_P_3(ALT16) TRGM_P_03(ALT17) ESC0_SDA(ALT18) QEIO_F(ALT20) ESC0_CTR_3(ALT27)	-	B00
D11	D9	76	PD04	GPIO_D_04(ALT0) MCAN1_RXD(ALT7) ESC0_GPI_27(ALT10) ESC0_GPO_36(ALT11) PPI0_DQ_10(ALT13) PWM0_P_4(ALT16) TRGM_P_04(ALT17) QEIO_H0(ALT20) ESC0_CTR_4(ALT27)	-	B00

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

19 PIN CONFIGURATION AND FUNCTION

PINMUX

Package			PIN NAME	Digital Function	Analog Function	Power Group
BGA _196	BGA _121	ELQ FP_1 00				
B12	D10	75	PD05	GPIO_D_05(ALT0) MCAN1_TXD(ALT7) ESC0_GPI_26(ALT10) ESC0_GPO_37(ALT11) PPI0_DQ_13(ALT13) PWM0_P_5(ALT16) TRGM_P_05(ALT17) QEIO_Z(ALT20) QEO0_Z(ALT21) ESC0_CTR_5(ALT27)	-	B00
C12	D11	74	PD06	GPIO_D_06(ALT0) I2C3_SDA(ALT4) ESC0_GPI_25(ALT10) ESC0_GPO_38(ALT11) PPI0_DQ_14(ALT13) PWM0_P_6(ALT16) TRGM_P_06(ALT17) QEIO_B(ALT20) QEO0_B(ALT21) ESC0_CTR_6(ALT27)	-	B00
B13	E9	73	PD07	GPIO_D_07(ALT0) I2C3_SCL(ALT4) ESC0_GPI_24(ALT10) ESC0_GPO_39(ALT11) PPI0_CTR_3(ALT13) PWM0_P_7(ALT16) TRGM_P_07(ALT17) QEIO_A(ALT20) QEO0_A(ALT21) ESC0_CTR_7(ALT27)	-	B00

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

19 PIN CONFIGURATION AND FUNCTION

PINMUX

Package			PIN NAME	Digital Function	Analog Function	Power Group
BGA _196	BGA _121	ELQ FP_1 00				
C13	-	-	PD08	GPIO_D_08(ALT0) I2C0_SCL(ALT4) MCAN2_TXD(ALT7) ESC0_GPI_23(ALT10) ESC0_GPO_40(ALT11) PPI0_CS_3(ALT13) PWM1_P_0(ALT16) TRGM_P_08(ALT17) ESC0_SCL(ALT18) ESC0_CTR_8(ALT27)	-	B00
D12	-	-	PD09	GPIO_D_09(ALT0) I2C0_SDA(ALT4) MCAN2_RXD(ALT7) ESC0_GPI_22(ALT10) ESC0_GPO_41(ALT11) PWM1_P_1(ALT16) TRGM_P_09(ALT17) ESC0_SDA(ALT18)	-	B00
E12	-	-	PD10	GPIO_D_10(ALT0) MCAN2_STBY(ALT7) ESC0_GPI_21(ALT10) ESC0_GPO_42(ALT11) PPI0_CTR_1(ALT13) PWM1_P_2(ALT16) TRGM_P_10(ALT17)	-	B00
E11	-	-	PD11	GPIO_D_11(ALT0) ESC0_GPI_20(ALT10) ESC0_GPO_43(ALT11) PPI0_CS_2(ALT13) PWM1_P_3(ALT16) TRGM_P_11(ALT17)	-	B00
F12	-	-	PD12	GPIO_D_12(ALT0) I2C1_SDA(ALT4) ESC0_GPI_19(ALT10) ESC0_GPO_44(ALT11) PPI0_DQ_15(ALT13) PWM1_P_4(ALT16) TRGM_P_12(ALT17)	-	B00

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

19 PIN CONFIGURATION AND FUNCTION

PINMUX

Package			PIN NAME	Digital Function	Analog Function	Power Group
BGA _196	BGA _121	ELQ FP_1 00				
F11	-	-	PD13	GPIO_D_13(ALT0) I2C1_SCL(ALT4) MCAN3_STBY(ALT7) ESC0_GPI_18(ALT10) ESC0_GPO_45(ALT11) PPI0_DM_1(ALT13) PWM1_P_5(ALT16) TRGM_P_13(ALT17)	-	B00
G14	-	-	PD14	GPIO_D_14(ALT0) MCAN3_RXD(ALT7) ESC0_GPI_17(ALT10) ESC0_GPO_46(ALT11) PPI0_DQ_08(ALT13) PWM1_P_6(ALT16) TRGM_P_14(ALT17) ESC0_REFCK(ALT18) SYSCTL_CLK_OBS_3(ALT 24)	-	B00
G13	-	-	PD15	GPIO_D_15(ALT0) MCAN3_TXD(ALT7) ESC0_GPI_16(ALT10) ESC0_GPO_47(ALT11) PPI0_DM_1(ALT13) PWM1_P_7(ALT16) TRGM_P_15(ALT17) ESC0_REFCK(ALT18) SYSCTL_CLK_OBS_2(ALT 24)	-	B00
G12	-	-	PD16	GPIO_D_16(ALT0) ESC0_GPI_15(ALT10) ESC0_GPO_48(ALT11) PPI0_DQ_10(ALT13) ESC0_GPO_00(ALT18) SYSCTL_CLK_OBS_0(ALT 24) ESC0_P2_RXDV(ALT27)	-	B00

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

19 PIN CONFIGURATION AND FUNCTION

PINMUX

Package			PIN NAME	Digital Function	Analog Function	Power Group
BGA _196	BGA _121	ELQ FP_1 00				
H14	-	-	PD17	GPIO_D_17(ALT0) ESC0_GPI_14(ALT10) ESC0_GPO_49(ALT11) PPI0_DQ_09(ALT13) ESC0_GPO_01(ALT18) SYSCTL_CLK_OBS_1(ALT 24) ESC0_P2_RXD_0(ALT27)	-	B00
H13	-	-	PD18	GPIO_D_18(ALT0) ESC0_GPI_13(ALT10) ESC0_GPO_50(ALT11) PPI0_DQ_12(ALT13) ESC0_GPO_02(ALT18) ESC0_P2_RXD_1(ALT27)	-	B00
H12	-	-	PD19	GPIO_D_19(ALT0) ESC0_GPI_12(ALT10) ESC0_GPO_51(ALT11) PPI0_DQ_11(ALT13) ESC0_GPO_03(ALT18) ESC0_P2_RXD_2(ALT27)	-	B00
J14	-	-	PD20	GPIO_D_20(ALT0) ESC0_GPI_11(ALT10) ESC0_GPO_52(ALT11) PPI0_DQ_14(ALT13) ESC0_GPO_04(ALT18) ESC0_P2_RXD_3(ALT27)	-	B00
J13	-	-	PD21	GPIO_D_21(ALT0) ESC0_GPI_10(ALT10) ESC0_GPO_53(ALT11) PPI0_DQ_13(ALT13) ESC0_GPO_05(ALT18) QEO1_Z(ALT21) ESC0_P2_RXCK(ALT27)	-	B00

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

19 PIN CONFIGURATION AND FUNCTION

PINMUX

Package			PIN NAME	Digital Function	Analog Function	Power Group
BGA _196	BGA _121	ELQ FP_1 00				
J12	-	-	PD22	GPIO_D_22(ALT0) ESC0_GPI_09(ALT10) ESC0_GPO_54(ALT11) PPI0_DQ_15(ALT13) ESC0_GPO_06(ALT18) QEO1_B(ALT21) ESC0_P2_RXER(ALT27)	-	B00
F10	-	-	PD23	GPIO_D_23(ALT0) ESC0_GPI_08(ALT10) ESC0_GPO_55(ALT11) PPI0_DM_0(ALT13) ESC0_GPO_07(ALT18) QEO1_A(ALT21) ESC0_P2_TXCK(ALT27)	-	B00
G11	-	-	PD24	GPIO_D_24(ALT0) ESC0_GPI_07(ALT10) ESC0_GPO_56(ALT11) PPI0_DQ_06(ALT13) ESC0_GPO_08(ALT18) SDM0_DAT_3(ALT23) ESC0_P2_TXD_0(ALT27)	-	B00
G10	-	-	PD25	GPIO_D_25(ALT0) ESC0_GPI_06(ALT10) ESC0_GPO_57(ALT11) PPI0_DQ_07(ALT13) ESC0_GPO_09(ALT18) SDM0_CLK_3(ALT23) OWR0_DAT(ALT26) ESC0_P2_TXD_1(ALT27)	-	B00
H11	-	-	PD26	GPIO_D_26(ALT0) ESC0_GPI_05(ALT10) ESC0_GPO_58(ALT11) PPI0_DQ_05(ALT13) ESC0_GPO_10(ALT18) QE11_H1(ALT20) SDM0_CLK_2(ALT23) OWR1_DAT(ALT26) ESC0_P2_TXD_2(ALT27)	-	B00

Package			PIN NAME	Digital Function	Analog Function	Power Group
BGA _196	BGA _121	ELQ FP_1 00				
H10	-	-	PD27	GPIO_D_27(ALT0) ESC0_GPI_04(ALT10) ESC0_GPO_59(ALT11) PPI0_DQ_04(ALT13) ESC0_GPO_11(ALT18) QE11_F(ALT20) SDM0_DAT_2(ALT23) ESC0_P2_TXD_3(ALT27)	-	B00
J11	-	-	PD28	GPIO_D_28(ALT0) ESC0_GPI_03(ALT10) ESC0_GPO_60(ALT11) PPI0_DQ_03(ALT13) ESC0_GPO_12(ALT18) QE11_H0(ALT20) SDM0_CLK_1(ALT23) ESC0_P2_TXEN(ALT27)	-	B00
J10	-	-	PD29	GPIO_D_29(ALT0) ESC0_GPI_02(ALT10) ESC0_GPO_61(ALT11) PPI0_DQ_02(ALT13) ESC0_GPO_13(ALT18) QE11_Z(ALT20) SDM0_DAT_1(ALT23) ESC0_REFCK(ALT27)	-	B00
K12	-	-	PD30	GPIO_D_30(ALT0) ESC0_GPI_01(ALT10) ESC0_GPO_62(ALT11) PPI0_DQ_01(ALT13) ESC0_GPO_14(ALT18) QE11_A(ALT20) SDM0_CLK_0(ALT23) ESC0_MDIO(ALT27)	-	B00

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

19 PIN CONFIGURATION AND FUNCTION

PINMUX

Package			PIN NAME	Digital Function	Analog Function	Power Group
BGA _196	BGA _121	ELQ FP_1 00				
K11	-	-	PD31	GPIO_D_31(ALT0) ESC0_GPI_00(ALT10) ESC0_GPO_63(ALT11) PPI0_DQ_00(ALT13) ESC0_GPO_15(ALT18) QE11_B(ALT20) SDM0_DAT_0(ALT23) ESC0_MDC(ALT27)	-	B00
M14	-	-	PE00	GPIO_E_00(ALT0) GPTMR1_COMP_0(ALT1) UART0_TXD(ALT2) MCAN0_TXD(ALT7) ESC0_CTR_4(ALT11) PPI0_CTR_4(ALT13) PWM0_P_0(ALT16) TRGM_P_00(ALT17) QEO1_A(ALT21) USB0_ID(ALT24) ETH0_EVTO_1(ALT25) ESC0_EVTO_1(ALT27)	-	B01
M13	-	-	PE01	GPIO_E_01(ALT0) GPTMR1_CAPT_0(ALT1) UART0_RXD(ALT2) MCAN0_RXD(ALT7) ESC0_CTR_5(ALT11) PPI0_CTR_5(ALT13) PWM0_P_1(ALT16) TRGM_P_01(ALT17) QEO1_B(ALT21) USB0_PWR(ALT24)	-	B01

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

19 PIN CONFIGURATION AND FUNCTION

PINMUX

Package			PIN NAME	Digital Function	Analog Function	Power Group
BGA _196	BGA _121	ELQ FP_1 00				
M12	-	-	PE02	GPIO_E_02(ALT0) GPTMR1_COMP_1(ALT1) UART0_DE(ALT2) UART0_RTS(ALT3) I2C2_SCL(ALT4) MCAN0_STBY(ALT7) ESC0_CTR_6(ALT11) PPI0_CTR_6(ALT13) PWM0_P_2(ALT16) TRGM_P_02(ALT17) QEO1_Z(ALT21)	-	B01
N14	-	-	PE03	GPIO_E_03(ALT0) GPTMR1_CAPT_1(ALT1) UART0_CTS(ALT3) I2C2_SDA(ALT4) MCAN1_STBY(ALT7) ESC0_CTR_1(ALT11) PPI0_CTR_7(ALT13) PWM0_P_3(ALT16) TRGM_P_03(ALT17) USB0_OC(ALT24)	-	B01
N13	-	-	PE04	GPIO_E_04(ALT0) UART1_CTS(ALT3) MCAN1_RXD(ALT7) ESC0_CTR_2(ALT11) PWM0_P_4(ALT16) TRGM_P_04(ALT17) ETH0_COL(ALT18) ETH0_EVTI_1(ALT25) ESC0_EVTI_1(ALT27)	-	B01

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

19 PIN CONFIGURATION AND FUNCTION

PINMUX

Package			PIN NAME	Digital Function	Analog Function	Power Group
BGA _196	BGA _121	ELQ FP_1 00				
P13	-	-	PE05	GPIO_E_05(ALT0) GPTMR1_COMP_2(ALT1) UART1_DE(ALT2) UART1_RTS(ALT3) MCAN1_TXD(ALT7) ESC0_CTR_3(ALT11) PWM0_P_5(ALT16) TRGM_P_05(ALT17) ETH0_CRS(ALT18)	-	B01
N10	-	-	PF00	GPIO_F_00(ALT0) MCAN0_TXD(ALT7) ESC0_MDC(ALT11) PWM0_P_0(ALT16) TRGM_P_00(ALT17) ETH0_MDC(ALT18) QEO1_A(ALT21)	-	B01
L10	-	-	PF01	GPIO_F_01(ALT0) MCAN0_RXD(ALT7) ESC0_MDIO(ALT11) PWM0_P_1(ALT16) TRGM_P_01(ALT17) ETH0_MDIO(ALT18) QEO1_B(ALT21)	-	B01
P11	L10	57	PF02	GPIO_F_02(ALT0) I2C2_SCL(ALT4) MCAN0_STBY(ALT7) ESC0_P1_TXCK(ALT11) PWM0_P_2(ALT16) TRGM_P_02(ALT17) ETH0_TXCK(ALT18) QEI0_H1(ALT20) QEO1_Z(ALT21) EUI0_CK(ALT26)	-	B01

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

19 PIN CONFIGURATION AND FUNCTION

PINMUX

Package			PIN NAME	Digital Function	Analog Function	Power Group
BGA _196	BGA _121	ELQ FP_1 00				
M9	J10	63	PF03	GPIO_F_03(ALT0) I2C2_SDA(ALT4) MCAN1_STBY(ALT7) ESC0_P1_TXD_0(ALT11) PWM0_P_3(ALT16) TRGM_P_03(ALT17) ETH0_TXD_0(ALT18) QEIO_F(ALT20) EUI0_SH(ALT26)	-	B01
M8	H9	51	PF04	GPIO_F_04(ALT0) MCAN1_RXD(ALT7) ESC0_P1_TXD_1(ALT11) PWM0_P_4(ALT16) TRGM_P_04(ALT17) ETH0_TXD_1(ALT18) QEIO_H0(ALT20) EUI0_DI(ALT26)	-	B01
L8	G9	56	PF05	GPIO_F_05(ALT0) MCAN1_TXD(ALT7) ESC0_P1_TXD_2(ALT11) PWM0_P_5(ALT16) TRGM_P_05(ALT17) ETH0_TXD_2(ALT18) QEIO_Z(ALT20) QEO0_Z(ALT21) EUI0_DO(ALT26)	-	B01
P12	K11	58	PF06	GPIO_F_06(ALT0) I2C3_SDA(ALT4) ESC0_P1_TXD_3(ALT11) PWM0_P_6(ALT16) TRGM_P_06(ALT17) ETH0_TXD_3(ALT18) QEIO_B(ALT20) QEO0_B(ALT21)	-	B01

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

19 PIN CONFIGURATION AND FUNCTION

PINMUX

Package			PIN NAME	Digital Function	Analog Function	Power Group
BGA _196	BGA _121	ELQ FP_1 00				
M11	H11	61	PF07	GPIO_F_07(ALT0) I2C3_SCL(ALT4) ESC0_P1_TXEN(ALT11) PWM0_P_7(ALT16) TRGM_P_07(ALT17) ETH0_TXEN(ALT18) QEIO_A(ALT20) QEO0_A(ALT21)	-	B01
L11	G11	64	PF08	GPIO_F_08(ALT0) I2C0_SCL(ALT4) MCAN2_TXD(ALT7) ESC0_P1_RXDV(ALT11) PWM1_P_0(ALT16) TRGM_P_08(ALT17) ETH0_RXDV(ALT18) EUI1_CK(ALT26)	-	B01
M10	H10	62	PF09	GPIO_F_09(ALT0) I2C0_SDA(ALT4) MCAN2_RXD(ALT7) ESC0_P1_RXD_0(ALT11) PWM1_P_1(ALT16) TRGM_P_09(ALT17) ETH0_RXD_0(ALT18) EUI1_SH(ALT26)	-	B01
P10	L9	53	PF10	GPIO_F_10(ALT0) MCAN2_STBY(ALT7) ESC0_P1_RXD_1(ALT11) PWM1_P_2(ALT16) TRGM_P_10(ALT17) ETH0_RXD_1(ALT18) EUI1_DI(ALT26)	-	B01
L9	G10	65	PF11	GPIO_F_11(ALT0) ESC0_P1_RXD_2(ALT11) PWM1_P_3(ALT16) TRGM_P_11(ALT17) ETH0_RXD_2(ALT18) EUI1_DO(ALT26)	-	B01

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

19 PIN CONFIGURATION AND FUNCTION

PINMUX

Package			PIN NAME	Digital Function	Analog Function	Power Group
BGA _196	BGA _121	ELQ FP_1 00				
N11	K10	60	PF12	GPIO_F_12(ALT0) I2C1_SDA(ALT4) ESC0_P1_RXD_3(ALT11) PWM1_P_4(ALT16) TRGM_P_12(ALT17) ETH0_RXD_3(ALT18) OWR0_DAT(ALT26)	-	B01
N12	J11	59	PF13	GPIO_F_13(ALT0) I2C1_SCL(ALT4) MCAN3_STBY(ALT7) ESC0_P1_RXCK(ALT11) PWM1_P_5(ALT16) TRGM_P_13(ALT17) ETH0_RXCK(ALT18) OWR1_DAT(ALT26)	-	B01
N9	K9	54	PF14	GPIO_F_14(ALT0) MCAN3_RXD(ALT7) ESC0_P1_RXER(ALT11) PWM1_P_6(ALT16) TRGM_P_14(ALT17) ETH0_RXER(ALT18)	-	B01
P9	-	-	PF15	GPIO_F_15(ALT0) MCAN3_TXD(ALT7) PWM1_P_7(ALT16) TRGM_P_15(ALT17) ETH0_TXER(ALT18)	-	B01
N7	K6	47	PF16	GPIO_F_16(ALT0) SDM0_DAT_0(ALT23)	ADC0_IN08 ADC1_IN08 ACMP0_CH0_I NN7	B02
P7	-	-	PF17	GPIO_F_17(ALT0) SDM0_CLK_0(ALT23)	ADC0_IN12 ADC1_IN12 ACMP0_CH1_I NN7	B02

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

19 PIN CONFIGURATION AND FUNCTION

PINMUX

Package			PIN NAME	Digital Function	Analog Function	Power Group
BGA _196	BGA _121	ELQ FP_1 00				
P8	L7	49	PF18	GPIO_F_18(ALT0) SDM0_DAT_1(ALT23) USB0_PWR(ALT24) ETH0_EVTO_0(ALT25) ESC0_EVTO_0(ALT27)	ADC0_IN11 ADC1_IN11 ACMP0_CH1_I NP7	B02
M5	K5	42	PF19	GPIO_F_19(ALT0) SDM0_CLK_1(ALT23) USB0_PWR(ALT24)	ADC0_IN13 ADC1_IN13 ACMP0_CH0_I NP7	B02
N6	L5	48	PF20	GPIO_F_20(ALT0) SDM0_DAT_3(ALT23) USB0_OC(ALT24) ETH0_EVTO_1(ALT25) ESC0_EVTO_1(ALT27)	ADC0_IN03 ADC1_IN03 ACMP0_CH0_I NP4 ACMP0_CH1_I NP4	B02
L6	J8	44	PF21	GPIO_F_21(ALT0) SDM0_DAT_2(ALT23) USB0_ID(ALT24) ETH0_EVTI_0(ALT25) ESC0_EVTI_0(ALT27)	ADC0_IN02 ADC1_IN02 ACMP0_CH0_I NN4 ACMP0_CH1_I NN4	B02
N8	L8	46	PF22	GPIO_F_22(ALT0) SDM0_CLK_2(ALT23) USB0_ID(ALT24) ETH0_EVTI_1(ALT25) ESC0_EVTI_1(ALT27)	ADC0_IN14 ADC1_IN14 ACMP0_CH0_I NN5 ACMP0_CH1_I NN5	B02
L5	J7	43	PF23	GPIO_F_23(ALT0) SDM0_CLK_3(ALT23) USB0_OC(ALT24)	ADC0_IN15 ADC1_IN15 ACMP0_CH0_I NN3 ACMP0_CH1_I NN3	B02

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

19 PIN CONFIGURATION AND FUNCTION

PINMUX

Package			PIN NAME	Digital Function	Analog Function	Power Group
BGA _196	BGA _121	ELQ FP_1 00				
M6	K7	41	PF24	GPIO_F_24(ALT0) SDM0_DAT_0(ALT23)	ADC0_IN10 ADC1_IN10 ACMP0_CH0_I NP5 ACMP0_CH1_I NP5	B02
L7	-	-	PF25	GPIO_F_25(ALT0) SDM0_CLK_0(ALT23)	ADC0_IN09 ADC1_IN09 ACMP0_CH0_I NP3 ACMP0_CH1_I NP3	B02
P6	L6	50	PF26	GPIO_F_26(ALT0) SDM0_DAT_1(ALT23)	ADC0_IN01 ADC1_IN01 ACMP0_CH0_I NN6 ACMP0_CH1_I NN6	B02
M7	K8	40	PF27	GPIO_F_27(ALT0) SDM0_CLK_1(ALT23)	ADC0_IN00 ADC1_IN00 ACMP0_CH0_I NP6 ACMP0_CH1_I NP6	B02
P5	-	-	PF28	GPIO_F_28(ALT0) SDM0_DAT_3(ALT23)	ADC0_IN05 ADC1_IN05 ACMP0_CH0_I NN2 ACMP0_CH1_I NN2	B02
M4	-	-	PF29	GPIO_F_29(ALT0) SDM0_DAT_2(ALT23)	ADC0_IN04 ADC1_IN04 ACMP0_CH0_I NP2 ACMP0_CH1_I NP2	B02

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

19 PIN CONFIGURATION AND FUNCTION

PINMUX

Package			PIN NAME	Digital Function	Analog Function	Power Group
BGA _196	BGA _121	ELQ FP_1 00				
L4	-	-	PF30	GPIO_F_30(ALT0) SDM0_CLK_2(ALT23)	ADC0_IN06 ADC1_IN06 ACMP0_CH0_I NN1 ACMP0_CH1_I NN1	B02
N5	-	-	PF31	GPIO_F_31(ALT0) SDM0_CLK_3(ALT23)	ADC0_IN07 ADC1_IN07 ACMP0_CH0_I NP1 ACMP0_CH1_I NP1	B02
-	-	-	PV00	GPIO_V_00(ALT0) GPTMR1_COMP_0(ALT1) UART0_TXD(ALT2) MCAN0_TXD(ALT7) ESC0_P0_RXDV(ALT11)	-	B00
-	-	-	PV01	GPIO_V_01(ALT0) GPTMR1_CAPT_0(ALT1) UART0_RXD(ALT2) MCAN0_RXD(ALT7) ESC0_P0_RXD_0(ALT11)	-	B00
-	-	-	PV02	GPIO_V_02(ALT0) GPTMR1_COMP_1(ALT1) UART0_DE(ALT2) UART0_RTS(ALT3) I2C2_SCL(ALT4) MCAN0_STBY(ALT7) ESC0_P0_RXD_1(ALT11)	-	B00
-	-	-	PV03	GPIO_V_03(ALT0) GPTMR1_CAPT_1(ALT1) UART0_CTS(ALT3) I2C2_SDA(ALT4) SPI1_CS_3(ALT5) MCAN1_STBY(ALT7) ESC0_P0_RXD_2(ALT11)	-	B00

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

19 PIN CONFIGURATION AND FUNCTION

PINMUX

Package			PIN NAME	Digital Function	Analog Function	Power Group
BGA _196	BGA _121	ELQ FP_1 00				
-	-	-	PV04	GPIO_V_04(ALT0) UART1_CTS(ALT3) SPI0_SCLK(ALT5) MCAN1_RXD(ALT7) ESC0_P0_RXD_3(ALT11)	-	B00
-	-	-	PV05	GPIO_V_05(ALT0) GPTMR1_COMP_2(ALT1) UART1_DE(ALT2) UART1_RTS(ALT3) SPI0_CS_0(ALT5) MCAN1_TXD(ALT7) ESC0_P0_RXCK(ALT11)	-	B00
-	-	-	PV06	GPIO_V_06(ALT0) GPTMR0_CAPT_0(ALT1) UART1_RXD(ALT2) I2C3_SDA(ALT4) SPI0_MISO(ALT5) ESC0_P0_TXCK(ALT11)	-	B00
-	-	-	PV07	GPIO_V_07(ALT0) GPTMR0_COMP_0(ALT1) UART1_TXD(ALT2) I2C3_SCL(ALT4) SPI0_MOSI(ALT5) ESC0_P0_TXD_0(ALT11)	-	B00
-	-	-	PV08	GPIO_V_08(ALT0) GPTMR0_COMP_1(ALT1) UART2_TXD(ALT2) I2C0_SCL(ALT4) SPI1_CS_2(ALT5) MCAN2_TXD(ALT7) ESC0_P0_TXD_1(ALT11)	-	B00
-	-	-	PV09	GPIO_V_09(ALT0) GPTMR0_CAPT_1(ALT1) UART2_RXD(ALT2) I2C0_SDA(ALT4) SPI1_CS_1(ALT5) MCAN2_RXD(ALT7) ESC0_P0_TXD_2(ALT11)	-	B00

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

19 PIN CONFIGURATION AND FUNCTION

PINMUX

Package			PIN NAME	Digital Function	Analog Function	Power Group
BGA _196	BGA _121	ELQ FP_1 00				
-	-	-	PV10	GPIO_V_10(ALT0) GPTMR0_COMP_2(ALT1) UART2_DE(ALT2) UART2_RTS(ALT3) SPI1_SCLK(ALT5) MCAN2_STBY(ALT7) ESC0_P0_TXD_3(ALT11)	-	B00
-	-	-	PV11	GPIO_V_11(ALT0) UART2_CTS(ALT3) SPI1_CS_0(ALT5) ESC0_P0_TXEN(ALT11)	-	B00
-	-	-	PV12	GPIO_V_12(ALT0) SPI1_MISO(ALT5) ESC0_CTR_0(ALT11)	-	B00
-	-	-	PV15	GPIO_V_15(ALT0) GPTMR0_COMP_3(ALT1) SPI1_DAT3(ALT5) ESC0_P0_RXER(ALT11)	-	B00
-	-	-	PW00	GPIO_W_00(ALT0) MCAN0_TXD(ALT7) ESC0_P1_RXDV(ALT11) ETH0_RXDV(ALT18)	-	B00
-	-	-	PW01	GPIO_W_01(ALT0) MCAN0_RXD(ALT7) ESC0_P1_RXD_0(ALT11) ETH0_RXD_0(ALT18)	-	B00
-	-	-	PW02	GPIO_W_02(ALT0) MCAN0_STBY(ALT7) ESC0_P1_RXD_1(ALT11) ETH0_RXD_1(ALT18)	-	B00
-	-	-	PW03	GPIO_W_03(ALT0) MCAN1_STBY(ALT7) ESC0_P1_RXD_2(ALT11) ETH0_RXD_2(ALT18)	-	B00
-	-	-	PW04	GPIO_W_04(ALT0) MCAN1_RXD(ALT7) ESC0_P1_RXD_3(ALT11) ETH0_RXD_3(ALT18)	-	B00

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

19 PIN CONFIGURATION AND FUNCTION

PINMUX

Package			PIN NAME	Digital Function	Analog Function	Power Group
BGA _196	BGA _121	ELQ FP_1 00				
-	-	-	PW05	GPIO_W_05(ALT0) MCAN1_TXD(ALT7) ESC0_P1_RXCK(ALT11) ETH0_RXCK(ALT18)	-	B00
-	-	-	PW06	GPIO_W_06(ALT0) ESC0_P1_TXCK(ALT11) ETH0_TXCK(ALT18)	-	B00
-	-	-	PW07	GPIO_W_07(ALT0) ESC0_P1_TXD_0(ALT11) ETH0_TXD_0(ALT18)	-	B00
-	-	-	PW08	GPIO_W_08(ALT0) MCAN2_TXD(ALT7) ESC0_P1_TXD_1(ALT11) ETH0_TXD_1(ALT18)	-	B00
-	-	-	PW09	GPIO_W_09(ALT0) MCAN2_RXD(ALT7) ESC0_P1_TXD_2(ALT11) ETH0_TXD_2(ALT18)	-	B00
-	-	-	PW10	GPIO_W_10(ALT0) MCAN2_STBY(ALT7) ESC0_P1_TXD_3(ALT11) ETH0_TXD_3(ALT18)	-	B00
-	-	-	PW11	GPIO_W_11(ALT0) ESC0_P1_TXEN(ALT11) ETH0_TXEN(ALT18)	-	B00
-	-	-	PW12	GPIO_W_12(ALT0) ESC0_CTR_1(ALT11) ETH0_RXER(ALT18)	-	B00
-	-	-	PW15	GPIO_W_15(ALT0) ESC0_P1_RXER(ALT11) ETH0_CRS(ALT18)	-	B00
-	-	-	PW16	GPIO_W_16(ALT0) ESC0_CTR_6(ALT11) ETH0_MDIO(ALT18)	-	B00
-	-	-	PW17	GPIO_W_17(ALT0) ESC0_CTR_7(ALT11) ETH0_MDC(ALT18)	-	B00

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

19 PIN CONFIGURATION AND FUNCTION

PINMUX

Package			PIN NAME	Digital Function	Analog Function	Power Group
BGA _196	BGA _121	ELQ FP_1 00				
-	-	-	PW20	GPIO_W_20(ALT0) ESC0_REFCK(ALT11) ETH0_TXER(ALT18) SOC_REF0(ALT24)	-	B00
-	-	-	PW21	GPIO_W_21(ALT0) ESC0_REFCK(ALT11) ETH0_COL(ALT18) SOC_REF1(ALT24)	-	B00
-	-	-	PX00	GPIO_X_00(ALT0) GPTMR3_COMP_0(ALT1) UART4_TXD(ALT2) XPIO_CB_DQS(ALT14)	-	B00
-	-	-	PX01	GPIO_X_01(ALT0) GPTMR3_CAPT_0(ALT1) UART4_RXD(ALT2) XPIO_CA_D_0(ALT14)	-	B00
-	-	-	PX02	GPIO_X_02(ALT0) GPTMR3_COMP_1(ALT1) UART4_DE(ALT2) UART4_RTS(ALT3) XPIO_CA_SCLK(ALT14)	-	B00
-	-	-	PX03	GPIO_X_03(ALT0) GPTMR3_CAPT_1(ALT1) UART4_CTS(ALT3) XPIO_CA_D_3(ALT14)	-	B00
-	-	-	PX04	GPIO_X_04(ALT0) UART5_CTS(ALT3) SPI3_SCLK(ALT5) XPIO_CA_DQS(ALT14)	-	B00
-	-	-	PX05	GPIO_X_05(ALT0) GPTMR3_COMP_2(ALT1) UART5_DE(ALT2) UART5_RTS(ALT3) SPI3_CS_0(ALT5) XPIO_CA_CS0(ALT14)	-	B00

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

19 PIN CONFIGURATION AND FUNCTION

PINMUX

Package			PIN NAME	Digital Function	Analog Function	Power Group
BGA _196	BGA _121	ELQ FP_1 00				
-	-	-	PX06	GPIO_X_06(ALT0) GPTMR2_CAPT_0(ALT1) UART5_RXD(ALT2) SPI3_MISO(ALT5) XPI0_CA_D_1(ALT14)	-	B00
-	-	-	PX07	GPIO_X_07(ALT0) GPTMR2_COMP_0(ALT1) UART5_TXD(ALT2) SPI3_MOSI(ALT5) XPI0_CA_D_2(ALT14)	-	B00
A1,P 1,N3, E5,K 5,F7, G7,H 7,J7, F8,G 8,E1 0,K1 0,A1 4,P1 4	A1,L 1,C3, F3,J3 ,C6,E 6,F6, G6,J 6,E7, F7,C 9,F9, J9,A1 1,L11	-	VSS	-	-	-
M1,M 2	K2,L 2	31	DCDC_IN	-	-	-
N1,P 2	K3,L 3	29	DCDC_GND	-	-	-
N2,P 3	K4,L 4	30	DCDC_LP	-	-	-
L3	J5	36	WKUP	-	-	-
M3	J4	35	RSTN	-	-	-
N4,P 4	H5	33	VPMC	-	-	-
E6,K 6,E7, E8,E 9	E4,D 7,D8, E8	19,55 ,66,7 2,82, 99	VIO_B01	-	-	-

Package			PIN NAME	Digital Function	Analog Function	Power Group
BGA _196	BGA _121	ELQ FP_1 00				
F6,G 6,F9, G9,H 9	D5,E 5,F5, G5,D 6	4,52, 71,77 ,98	VDD_SOC	-	-	-
H6	D4	-	VDD_PHY1CAP	-	-	-
J6	G4	-	VDD_PHY0CAP	-	-	-
K7	H4	34	VDD_PMCCAP	-	-	-
H8	G7	38	VSSA	-	-	-
J8	H8	38	VREFL	-	-	-
K8	F4	23	VDD_OTPCAP	-	-	-
J9	H6	37	VANA	-	-	-
K9	H7	39	VREFH	-	-	-
L12	F8	-	USB0_VBUS	-	-	-
K13	E11	70	XTALO	-	-	-
L13	F10	67	USB0_DN	-	-	-
K14	E10	69	XTALI	-	-	-
L14	F11	68	USB0_DP	-	-	-
-	G8	45	VIO_B02	-	-	-
-	-	32	DCDC_SNS	-	-	-

Table 91: SOC IOMUX

19.2 System Power Out-Field Pin Allocation

The pin allocations of different peripherals in the system power domain are summarized as follows:

ACMP0 signal name	Pin
CH0_INN1	PF30
CH0_INN2	PF28
CH0_INN3	PF23
CH0_INN4	PF21

ACMP0 signal name	Pin
CH0_INN5	PF22
CH0_INN6	PF26
CH0_INN7	PF16
CH0_INP1	PF31
CH0_INP2	PF29
CH0_INP3	PF25
CH0_INP4	PF20
CH0_INP5	PF24
CH0_INP6	PF27
CH0_INP7	PF19
CH1_INN1	PF30
CH1_INN2	PF28
CH1_INN3	PF23
CH1_INN4	PF21
CH1_INN5	PF22
CH1_INN6	PF26
CH1_INN7	PF17
CH1_INP1	PF31
CH1_INP2	PF29
CH1_INP3	PF25
CH1_INP4	PF20
CH1_INP5	PF24
CH1_INP6	PF27
CH1_INP7	PF18

Table 92: ACMP0 SIGNAL PIN

ADC0 signal name	Pin
IN00	PF27
IN01	PF26
IN02	PF21
IN03	PF20
IN04	PF29
IN05	PF28
IN06	PF30
IN07	PF31
IN08	PF16
IN09	PF25
IN10	PF24
IN11	PF18
IN12	PF17

ADC0 signal name	Pin
IN13	PF19
IN14	PF22
IN15	PF23

Table 93: ADC0 SIGNAL PIN

ADC1 signal name	Pin
IN00	PF27
IN01	PF26
IN02	PF21
IN03	PF20
IN04	PF29
IN05	PF28
IN06	PF30
IN07	PF31
IN08	PF16
IN09	PF25
IN10	PF24
IN11	PF18
IN12	PF17
IN13	PF19
IN14	PF22
IN15	PF23

Table 94: ADC1 SIGNAL PIN

ESC0 Port	High Speed Group	High speed signal name	Pin
	A	P0_RXCK	PA21
		P0_RXDV	PA16
		P0_RXD_0	PA17
		P0_RXD_1	PA18
		P0_RXD_2	PA19
		P0_RXD_3	PA20
		P0_RXER	PA23
		P0_TXCK	PA24
		P0_TXD_0	PA25
		P0_TXD_1	PA26
		P0_TXD_2	PA27
		P0_TXD_3	PA28
		P0_TXEN	PA29
		P0_RXCK	PB05

ESC0 Port	High Speed Group	High speed signal name	Pin
		P0_RXDV	PB00
		P0_RXD_0	PB01
		P0_RXD_1	PB02
		P0_RXD_2	PB03
		P0_RXD_3	PB04
		P0_RXER	PA23
		P0_TXCK	PB06
		P0_TXD_0	PB07
		P0_TXD_1	PB08
		P0_TXD_2	PB09
		P0_TXD_3	PB10
		P0_TXEN	PB11
	C	P0_RXCK	PV05
		P0_RXDV	PV00
		P0_RXD_0	PV01
		P0_RXD_1	PV02
		P0_RXD_2	PV03
		P0_RXD_3	PV04
		P0_RXER	PV15
		P0_TXCK	PV06
		P0_TXD_0	PV07
		P0_TXD_1	PV08
		P0_TXD_2	PV09
P1	A	P1_RXCK	PF13
		P1_RXDV	PF08
		P1_RXD_0	PF09
		P1_RXD_1	PF10
		P1_RXD_2	PF11
		P1_RXD_3	PF12
		P1_RXER	PF14
		P1_TXCK	PF02
		P1_TXD_0	PF03
		P1_TXD_1	PF04
		P1_TXD_2	PF05
		P1_TXD_3	PF06
		P1_TXEN	PF07
		P1_RXCK	PW05
		P1_RXDV	PW00
		P1_RXD_0	PW01

ESC0 Port	High Speed Group	High speed signal name	Pin
		P1_RXD_1	PW02
		P1_RXD_2	PW03
		P1_RXD_3	PW04
		P1_RXER	PW15
		P1_TXCK	PW06
		P1_TXD_0	PW07
		P1_TXD_1	PW08
		P1_TXD_2	PW09
		P1_TXD_3	PW10
		P1_TXEN	PW11
P2	A	P2_RXCK	PD21
		P2_RXDV	PD16
		P2_RXD_0	PD17
		P2_RXD_1	PD18
		P2_RXD_2	PD19
		P2_RXD_3	PD20
		P2_RXER	PD22
		P2_TXCK	PD23
		P2_TXD_0	PD24
		P2_TXD_1	PD25
		P2_TXD_2	PD26
		P2_TXD_3	PD27
		P2_TXEN	PD28
-	A	MDC	PA31
		MDIO	PA30
	B	MDC	PF00
		MDIO	PF01
	C	MDC	PD31
		MDIO	PD30

Note: Signals on the same port must use the same group of pins, and signals on different ports can use different groups of pins.

Table 95: ESC0 HIGH SPEED SIGNAL PIN

ESC0 Non-high speed signal name	Pin
CTR_0	PA25
	PC09
	PC18
	PD00
	PV12
	PA28

ESC0 Non-high speed signal name	Pin
	PC10 PC19 PD01 PE03 PW12
CTR_2	PC11 PC20 PD02 PE04
CTR_3	PC12 PC21 PD03 PE05
CTR_4	PB24 PC13 PC27 PD04 PE00
CTR_5	PB25 PC14 PC28 PD05 PE01
CTR_6	PB26 PC15 PC29 PD06 PE02 PW16
CTR_7	PB27 PB30 PC16 PC30 PD07 PW17
CTR_8	PB28 PB31 PC17 PC31 PD08

ESC0 Non-high speed signal name	Pin
EVTI_0	PB24 PF21
EVTI_1	PB26 PE04 PF22
EVTO_0	PB25 PF18
EVTO_1	PB29 PE00 PF20
GPI_00	PC15 PD31
GPI_01	PC14 PD30
GPI_02	PC13 PD29
GPI_03	PC12 PD28
GPI_04	PC11 PD27
GPI_05	PC10 PD26
GPI_06	PC09 PD25
GPI_07	PC08 PD24
GPI_08	PC07 PD23
GPI_09	PC06 PD22
GPI_10	PC05 PD21
GPI_11	PC04 PD20
GPI_12	PC03 PD19
GPI_13	PC02 PD18
GPI_14	PC01 PD17

ESC0 Non-high speed signal name	Pin
GPI_15	PC00 PD16
GPI_16	PD15
GPI_17	PD14
GPI_18	PD13
GPI_19	PD12
GPI_20	PD11
GPI_21	PD10
GPI_22	PD09
GPI_23	PD08
GPI_24	PD07
GPI_25	PD06
GPI_26	PD05
GPI_27	PD04
GPI_28	PD03
GPI_29	PD02
GPI_30	PD01
GPI_31	PD00
GPI_32	PC31
GPI_33	PC30
GPI_34	PC29
GPI_35	PC28
GPI_36	PC27
GPI_37	PC26
GPI_38	PC25
GPI_39	PC24
GPI_40	PC23
GPI_41	PC22
GPI_42	PC21
GPI_43	PC20
GPI_44	PC19
GPI_45	PC18
GPI_46	PC17
GPI_47	PC16
GPI_48	PC15
GPI_49	PC14
GPI_50	PC13
GPI_51	PC12
GPI_52	PC11

ESC0 Non-high speed signal name	Pin
GPI_53	PC10
GPI_54	PC09
GPI_55	PC08
GPI_56	PC07
GPI_57	PC06
GPI_58	PC05
GPI_59	PC04
GPI_60	PC03
GPI_61	PC02
GPI_62	PC01
GPI_63	PC00
GPO_00	PC00 PD16
GPO_01	PC01 PD17
GPO_02	PC02 PD18
GPO_03	PC03 PD19
GPO_04	PC04 PD20
GPO_05	PC05 PD21
GPO_06	PC06 PD22
GPO_07	PC07 PD23
GPO_08	PC08 PD24
GPO_09	PC09 PD25
GPO_10	PC10 PD26
GPO_11	PC11 PD27
GPO_12	PC12 PD28
GPO_13	PC13 PD29
GPO_14	PC14

ESC0 Non-high speed signal name	Pin
	PD30
GPO_15	PC15 PD31
GPO_16	PC16
GPO_17	PC17
GPO_18	PC18
GPO_19	PC19
GPO_20	PC20
GPO_21	PC21
GPO_22	PC22
GPO_23	PC23
GPO_24	PC24
GPO_25	PC25
GPO_26	PC26
GPO_27	PC27
GPO_28	PC28
GPO_29	PC29
GPO_30	PC30
GPO_31	PC31
GPO_32	PD00
GPO_33	PD01
GPO_34	PD02
GPO_35	PD03
GPO_36	PD04
GPO_37	PD05
GPO_38	PD06
GPO_39	PD07
GPO_40	PD08
GPO_41	PD09
GPO_42	PD10
GPO_43	PD11
GPO_44	PD12
GPO_45	PD13
GPO_46	PD14
GPO_47	PD15
GPO_48	PD16
GPO_49	PD17
GPO_50	PD18
GPO_51	PD19

ESC0 Non-high speed signal name	Pin
GPO_52	PD20
GPO_53	PD21
GPO_54	PD22
GPO_55	PD23
GPO_56	PD24
GPO_57	PD25
GPO_58	PD26
GPO_59	PD27
GPO_60	PD28
GPO_61	PD29
GPO_62	PD30
GPO_63	PD31
REFCK	PA09 PD00 PD01 PD14 PD15 PD29 PW20 PW21
SCL	PD02 PD08
SDA	PD03 PD09

Table 96: ESC0 NON-HIGH SPEED SIGNAL PIN

ETH0 High Speed Group	High speed signal name	Pin
A	MDC	PF00
	MDIO	PF01
	RXCK	PF13
	RXDV	PF08
	RXD_0	PF09
	RXD_1	PF10
	RXD_2	PF11
	RXD_3	PF12
	RXER	PF14
	TXCK	PF02
	TXD_0	PF03
	TXD_1	PF04

ETH0 High Speed Group	High speed signal name	Pin
	TXD_2	PF05
	TXD_3	PF06
	TXEN	PF07
	TXER	PF15
B	MDC	PA31
	MDIO	PA30
	RXCK	PB05
	RXDV	PB00
	RXD_0	PB01
	RXD_1	PB02
	RXD_2	PB03
	RXD_3	PB04
	RXER	PA23
	TXCK	PB06
	TXD_0	PB07
	TXD_1	PB08
	TXD_2	PB09
	TXD_3	PB10
	TXEN	PB11
	TXER	PA22
C	MDC	PW17
	MDIO	PW16
	RXCK	PW05
	RXDV	PW00
	RXD_0	PW01
	RXD_1	PW02
	RXD_2	PW03
	RXD_3	PW04
	RXER	PW12
	TXCK	PW06
	TXD_0	PW07
	TXD_1	PW08
	TXD_2	PW09
	TXD_3	PW10
	TXEN	PW11
	TXER	PW20

Table 97: ETH0 HIGH SPEED SIGNAL PIN

ETH0 Non-high speed signal name	Pin
COL	PE04 PW21
CRS	PE05 PW15
EVTI_0	PB24 PF21
EVTI_1	PB26 PE04 PF22
EVTO_0	PB25 PF18
EVTO_1	PB29 PE00 PF20

Table 98: ETH0 NON-HIGH SPEED SIGNAL PIN

EUI0 signal name	Pin
CK	PA16 PF02
DI	PA18 PF04
DO	PA19 PF05
SH	PA17 PF03

Table 99: EUI0 SIGNAL PIN

EUI1 signal name	Pin
CK	PB26 PF08
DI	PB28 PF10
DO	PB29 PF11
SH	PB27 PF09

Table 100: EUI1 SIGNAL PIN

EWDG0 signal name	Pin
RST	PA26

Table 101: EWDG0 SIGNAL PIN

EWDG1 signal name	Pin
RST	PD02

Table 102: EWDG1 SIGNAL PIN

GPIO signal name	Pin
A_00	PA00
A_01	PA01
A_02	PA02
A_03	PA03
A_04	PA04
A_05	PA05
A_06	PA06
A_07	PA07
A_08	PA08
A_09	PA09
A_16	PA16
A_17	PA17
A_18	PA18
A_19	PA19
A_20	PA20
A_21	PA21
A_22	PA22
A_23	PA23
A_24	PA24
A_25	PA25
A_26	PA26
A_27	PA27
A_28	PA28
A_29	PA29
A_30	PA30
A_31	PA31
B_00	PB00
B_01	PB01
B_02	PB02
B_03	PB03
B_04	PB04

GPIO signal name	Pin
B_05	PB05
B_06	PB06
B_07	PB07
B_08	PB08
B_09	PB09
B_10	PB10
B_11	PB11
B_24	PB24
B_25	PB25
B_26	PB26
B_27	PB27
B_28	PB28
B_29	PB29
B_30	PB30
B_31	PB31
C_00	PC00
C_01	PC01
C_02	PC02
C_03	PC03
C_04	PC04
C_05	PC05
C_06	PC06
C_07	PC07
C_08	PC08
C_09	PC09
C_10	PC10
C_11	PC11
C_12	PC12
C_13	PC13
C_14	PC14
C_15	PC15
C_16	PC16
C_17	PC17
C_18	PC18
C_19	PC19
C_20	PC20
C_21	PC21
C_22	PC22
C_23	PC23
C_24	PC24

GPIO signal name	Pin
C_25	PC25
C_26	PC26
C_27	PC27
C_28	PC28
C_29	PC29
C_30	PC30
C_31	PC31
D_00	PD00
D_01	PD01
D_02	PD02
D_03	PD03
D_04	PD04
D_05	PD05
D_06	PD06
D_07	PD07
D_08	PD08
D_09	PD09
D_10	PD10
D_11	PD11
D_12	PD12
D_13	PD13
D_14	PD14
D_15	PD15
D_16	PD16
D_17	PD17
D_18	PD18
D_19	PD19
D_20	PD20
D_21	PD21
D_22	PD22
D_23	PD23
D_24	PD24
D_25	PD25
D_26	PD26
D_27	PD27
D_28	PD28
D_29	PD29
D_30	PD30
D_31	PD31
E_00	PE00

GPIO signal name	Pin
E_01	PE01
E_02	PE02
E_03	PE03
E_04	PE04
E_05	PE05
F_00	PF00
F_01	PF01
F_02	PF02
F_03	PF03
F_04	PF04
F_05	PF05
F_06	PF06
F_07	PF07
F_08	PF08
F_09	PF09
F_10	PF10
F_11	PF11
F_12	PF12
F_13	PF13
F_14	PF14
F_15	PF15
F_16	PF16
F_17	PF17
F_18	PF18
F_19	PF19
F_20	PF20
F_21	PF21
F_22	PF22
F_23	PF23
F_24	PF24
F_25	PF25
F_26	PF26
F_27	PF27
F_28	PF28
F_29	PF29
F_30	PF30
F_31	PF31
V_00	PV00
V_01	PV01
V_02	PV02

GPIO signal name	Pin
V_03	PV03
V_04	PV04
V_05	PV05
V_06	PV06
V_07	PV07
V_08	PV08
V_09	PV09
V_10	PV10
V_11	PV11
V_12	PV12
V_15	PV15
W_00	PW00
W_01	PW01
W_02	PW02
W_03	PW03
W_04	PW04
W_05	PW05
W_06	PW06
W_07	PW07
W_08	PW08
W_09	PW09
W_10	PW10
W_11	PW11
W_12	PW12
W_15	PW15
W_16	PW16
W_17	PW17
W_20	PW20
W_21	PW21
X_00	PX00
X_01	PX01
X_02	PX02
X_03	PX03
X_04	PX04
X_05	PX05
X_06	PX06
X_07	PX07

Table 103: GPIO SIGNAL PIN

GPTMR0 signal name	Pin
CAPT_0	PA06
	PC06
	PV06
CAPT_1	PA09
	PC09
	PV09
COMP_0	PA07
	PC07
	PV07
COMP_1	PA08
	PC08
	PV08
COMP_2	PC10
	PV10
COMP_3	PC15
	PV15

Table 104: GPTMR0 SIGNAL PIN

GPTMR1 signal name	Pin
CAPT_0	PA01
	PC01
	PE01
	PV01
CAPT_1	PA03
	PC03
	PE03
	PV03
COMP_0	PA00
	PC00
	PE00
	PV00
COMP_1	PA02
	PC02
	PE02
	PV02
COMP_2	PA05
	PC05
	PE05
	PV05
COMP_3	PC13

GPTMR1 signal name	Pin

Table 105: GPTMR1 SIGNAL PIN

GPTMR2 signal name	Pin
CAPT_0	PA22
	PC22
	PX06
CAPT_1	PA25
	PC25
COMP_0	PA23
	PC23
	PX07
COMP_1	PA24
	PC24
COMP_2	PA26
	PC26
COMP_3	PA31
	PC31

Table 106: GPTMR2 SIGNAL PIN

GPTMR3 signal name	Pin
CAPT_0	PA17
	PC17
	PX01
CAPT_1	PA19
	PC19
	PX03
COMP_0	PA16
	PC16
	PX00
COMP_1	PA18
	PC18
	PX02
COMP_2	PA21
	PC21
	PX05
COMP_3	PA29
	PC29

Table 107: GPTMR3 SIGNAL PIN

I2C0 signal name	Pin
SCL	PA08
	PB08
	PC08
	PD08
	PF08
	PV08
SDA	PA09
	PB09
	PC09
	PD09
	PF09
	PV09

Table 108: I2C0 SIGNAL PIN

I2C1 signal name	Pin
SCL	PC13
	PD13
	PF13
SDA	PC12
	PD12
	PF12

Table 109: I2C1 SIGNAL PIN

I2C2 signal name	Pin
SCL	PA02
	PB02
	PC02
	PD02
	PE02
	PF02
	PV02
SDA	PA03
	PB03
	PC03
	PD03
	PE03
	PF03
	PV03

Table 110: I2C2 SIGNAL PIN

I2C3 signal name	Pin
SCL	PA07
	PB07
	PC07
	PD07
	PF07
	PV07
SDA	PA06
	PB06
	PC06
	PD06
	PF06
	PV06

Table 111: I2C3 SIGNAL PIN

JTAG signal name	Pin
TCK	PA06
TDI	PA05
TDO	PA04
TMS	PA07
TRST	PA08

Table 112: JTAG SIGNAL PIN

MCAN0 signal name	Pin
RXD	PA01
	PB01
	PC01
	PD01
	PE01
	PF01
	PV01
	PW01
STBY	PA02
	PB02
	PC02
	PD02
	PE02
	PF02
	PV02
	PW02
	PA00

MCAN0 signal name	Pin
	PB00
	PC00
	PD00
	PE00
	PF00
	PV00
	PW00

Table 113: MCAN0 SIGNAL PIN

MCAN1 signal name	Pin
RXD	PA04
	PB04
	PC04
	PD04
	PE04
	PF04
	PV04
STBY	PW04
	PA03
	PB03
	PC03
	PD03
	PE03
	PF03
TXD	PV03
	PW03
	PA05
	PB05
	PC05
	PD05
	PE05
	PF05
	PV05
	PW05

Table 114: MCAN1 SIGNAL PIN

MCAN2 signal name	Pin
RXD	PA09
	PB09
	PC09

MCAN2 signal name	Pin
	PD09 PF09 PV09 PW09
STBY	PB10 PC10 PD10 PF10 PV10 PW10
TXD	PA08 PB08 PC08 PD08 PF08 PV08 PW08

Table 115: MCAN2 SIGNAL PIN

MCAN3 signal name	Pin
RXD	PC14 PD14 PF14
STBY	PC13 PD13 PF13
TXD	PC15 PD15 PF15

Table 116: MCAN3 SIGNAL PIN

OWR0 signal name	Pin
DAT	PA21 PC00 PD25 PF12

Table 117: OWR0 SIGNAL PIN

OWR1 signal name	Pin
DAT	PA22
	PC01
	PD26
	PF13

Table 118: OWR1 SIGNAL PIN

PPI0 High Speed Group	High speed signal name	Pin
A	CLK	PC29
	CS_0	PC30
	CS_1	PC31
	CS_2	PD11
	CS_3	PD08
	CTR_0	PC27
	CTR_1	PD10
	CTR_2	PC22
	CTR_3	PD07
	CTR_4	PE00
	CTR_5	PE01
	CTR_6	PE02
	CTR_7	PE03
	DM_0	PD23
	DM_1	PD15
	DM_2	PC12
	DM_3	PC15
	DQ_00	PD31
	DQ_01	PD30
	DQ_02	PD29
	DQ_03	PD28
	DQ_04	PD27
	DQ_05	PD26
	DQ_06	PD24
	DQ_07	PD25
	DQ_08	PD14
	DQ_09	PD17
	DQ_10	PD16
	DQ_11	PD19
	DQ_12	PD18
	DQ_13	PD21
	DQ_14	PD20
	DQ_15	PD22

PPI0 High Speed Group	High speed signal name	Pin
	DQ_16	PC16
	DQ_17	PC17
	DQ_18	PC13
	DQ_19	PC14
	DQ_20	PC10
	DQ_21	PC11
	DQ_22	PC02
	DQ_23	PC09
	DQ_24	PC00
	DQ_25	PC01
	DQ_26	PC03
	DQ_27	PC04
	DQ_28	PC05
	DQ_29	PC06
	DQ_30	PC07
	DQ_31	PC08
B	CLK	PC29
	CS_0	PC30
	CS_1	PC31
	CS_2	PD11
	CS_3	PD08
	CTR_0	PC27
	CTR_1	PD10
	CTR_2	PC22
	CTR_3	PD07
	CTR_4	PE00
	CTR_5	PE01
	CTR_6	PE02
	CTR_7	PE03
	DM_0	PC28
	DM_1	PD13
	DQ_00	PD02
	DQ_01	PD03
	DQ_02	PD00
	DQ_03	PD01
	DQ_04	PC18
	DQ_05	PC19
	DQ_06	PC20
	DQ_07	PC21
	DQ_08	PC23
	DQ_09	PC24

PPI0 High Speed Group	High speed signal name	Pin
	DQ_10	PD04
	DQ_11	PC25
	DQ_12	PC26
	DQ_13	PD05
	DQ_14	PD06
	DQ_15	PD12

Table 119: PPI0 HIGH SPEED SIGNAL PIN

PWM0 High Speed Group	High speed signal name	Pin
A	P_0	PA00
	P_1	PA01
	P_2	PA02
	P_3	PA03
	P_4	PA04
	P_5	PA05
	P_6	PA06
	P_7	PA07
B	P_0	PB00
	P_1	PB01
	P_2	PB02
	P_3	PB03
	P_4	PB04
	P_5	PB05
	P_6	PB06
	P_7	PB07
C	P_0	PC00
	P_1	PC01
	P_2	PC02
	P_3	PC03
	P_4	PC04
	P_5	PC05
	P_6	PC06
	P_7	PC07
D	P_0	PD00
	P_1	PD01
	P_2	PD02
	P_3	PD03
	P_4	PD04
	P_5	PD05

PWM0 High Speed Group	High speed signal name	Pin
	P_6	PD06
	P_7	PD07
E	P_0	PE00
	P_1	PE01
	P_2	PE02
	P_3	PE03
	P_4	PE04
	P_5	PE05
F	P_0	PF00
	P_1	PF01
	P_2	PF02
	P_3	PF03
	P_4	PF04
	P_5	PF05
	P_6	PF06
	P_7	PF07

Table 120: PWM0 HIGH SPEED SIGNAL PIN

PWM1 High Speed Group	High speed signal name	Pin
A	P_0	PA08
	P_1	PA09
B	P_0	PB08
	P_1	PB09
	P_2	PB10
	P_3	PB11
C	P_0	PC08
	P_1	PC09
	P_2	PC10
	P_3	PC11
	P_4	PC12
	P_5	PC13
	P_6	PC14
	P_7	PC15
D	P_0	PD08
	P_1	PD09
	P_2	PD10
	P_3	PD11
	P_4	PD12
	P_5	PD13

PWM1 High Speed Group	High speed signal name	Pin
	P_6	PD14
	P_7	PD15
E	P_0	PF08
	P_1	PF09
	P_2	PF10
	P_3	PF11
	P_4	PF12
	P_5	PF13
	P_6	PF14
	P_7	PF15

Table 121: PWM1 HIGH SPEED SIGNAL PIN

QEIO signal name	Pin
A	PB07
	PC07
	PD07
	PF07
B	PB06
	PC06
	PD06
	PF06
F	PB03
	PC03
	PD03
	PF03
H0	PB04
	PC04
	PD04
	PF04
H1	PB02
	PC02
	PD02
	PF02
Z	PB05
	PC05
	PD05
	PF05

Table 122: QEIO SIGNAL PIN

QE11 signal name	Pin
A	PA30
	PB30
	PC13
	PC30
	PD30
B	PA31
	PB31
	PC14
	PC31
	PD31
F	PA27
	PB27
	PC10
	PC27
	PD27
H0	PA28
	PB28
	PC11
	PC28
	PD28
H1	PA26
	PB26
	PC09
	PC26
	PD26
Z	PA29
	PB29
	PC12
	PC29
	PD29

Table 123: QE11 SIGNAL PIN

QEO0 signal name	Pin
A	PA23
	PB07
	PC07
	PD07
	PF07
B	PA22
	PB06
	PC06

QEO0 signal name	Pin
Z	PD06
	PF06
	PA21
	PB05
	PC05
	PD05
	PF05

Table 124: QEO0 SIGNAL PIN

QEO1 signal name	Pin
A	PA16
	PB00
	PC13
	PC23
	PD23
	PE00
	PF00
B	PA17
	PB01
	PC14
	PC22
	PD22
	PE01
	PF01
Z	PA18
	PB02
	PC12
	PC21
	PD21
	PE02
	PF02

Table 125: QEO1 SIGNAL PIN

SDM0 High Speed Group	High speed signal name	Pin
A	CLK_0	PF17
	CLK_1	PF19
	CLK_2	PF22
	CLK_3	PF23
	DAT_0	PF16

SDM0 High Speed Group	High speed signal name	Pin
	DAT_1	PF18
	DAT_2	PF21
	DAT_3	PF20
B	CLK_0	PF25
	CLK_1	PF27
	CLK_2	PF30
	CLK_3	PF31
	DAT_0	PF24
	DAT_1	PF26
	DAT_2	PF29
C	DAT_3	PF28
	CLK_0	PA23
	CLK_1	PA21
	CLK_2	PA19
	CLK_3	PA17
	DAT_0	PA22
	DAT_1	PA20
	DAT_2	PA18
D	DAT_3	PA16
	CLK_0	PD30
	CLK_1	PD28
	CLK_2	PD26
	CLK_3	PD25
	DAT_0	PD31
	DAT_1	PD29
	DAT_2	PD27
E	DAT_3	PD24
	CLK_0	PB30
	CLK_1	PB28
	CLK_2	PB26
	CLK_3	PB24
	DAT_0	PB31
	DAT_1	PB29
	DAT_2	PB27
	DAT_3	PB25

Table 126: SDM0 HIGH SPEED SIGNAL PIN

SOC signal name	Pin
REF0	PC29
	PW20

SOC signal name	Pin
REF1	PC30 PW21

Table 127: SOC SIGNAL PIN

SPI0 High Speed Group	High speed signal name	Pin
A	CS_0	PC05
	MISO	PC06
	MOSI	PC07
	SCLK	PC04
B	CS_0	PV05
	MISO	PV06
	MOSI	PV07
	SCLK	PV04

Table 128: SPI0 HIGH SPEED SIGNAL PIN

SPI1 High Speed Group	High speed signal name	Pin
A	CS_0	PA05
	MISO	PA06
	MOSI	PA07
	SCLK	PA04
B	CS_0	PC11
	CS_1	PC09
	CS_2	PC08
	CS_3	PC03
	DAT2	PC14
	DAT3	PC15
	MISO	PC12
	MOSI	PC13
C	SCLK	PC10
	CS_0	PV11
	CS_1	PV09
	CS_2	PV08
	CS_3	PV03
	DAT3	PV15
	MISO	PV12
	SCLK	PV10

Table 129: SPI1 HIGH SPEED SIGNAL PIN

SPI2 High Speed Group	High speed signal name	Pin
A	CS_0	PA21
	MISO	PA22
	MOSI	PA23
	SCLK	PA20
B	CS_0	PC27
	CS_1	PC25
	CS_2	PC24
	CS_3	PC19
	DAT2	PC30
	DAT3	PC31
	MISO	PC28
	MOSI	PC29
	SCLK	PC26

Table 130: SPI2 HIGH SPEED SIGNAL PIN

SPI3 High Speed Group	High speed signal name	Pin
A	CS_0	PA27
	CS_1	PA25
	CS_2	PA24
	CS_3	PA19
	DAT2	PA30
	DAT3	PA31
	MISO	PA28
	MOSI	PA29
	SCLK	PA26
B	CS_0	PC21
	MISO	PC22
	MOSI	PC23
	SCLK	PC20
C	CS_0	PX05
	MISO	PX06
	MOSI	PX07
	SCLK	PX04

Table 131: SPI3 HIGH SPEED SIGNAL PIN

SYSCTL signal name	Pin
CLK_OBS_0	PA00
	PD16

SYSCTL signal name	Pin
CLK_OBS_1	PA02 PD17
CLK_OBS_2	PA01 PD15
CLK_OBS_3	PA03 PD14

Table 132: SYSCTL SIGNAL PIN

TRGM signal name	Pin
P_00	PA00 PB00 PC00 PD00 PE00 PF00
P_01	PA01 PB01 PC01 PD01 PE01 PF01
P_02	PA02 PB02 PC02 PD02 PE02 PF02
P_03	PA03 PB03 PC03 PD03 PE03 PF03
P_04	PA04 PB04 PC04 PD04 PE04 PF04
P_05	PA05 PB05

TRGM signal name	Pin
	PC05 PD05 PE05 PF05
P_06	PA06 PB06 PC06 PD06 PF06
P_07	PA07 PB07 PC07 PD07 PF07
P_08	PA08 PB08 PC08 PD08 PF08
P_09	PA09 PB09 PC09 PD09 PF09
P_10	PB10 PC10 PD10 PF10
P_11	PB11 PC11 PD11 PF11
P_12	PC12 PD12 PF12
P_13	PC13 PD13 PF13
P_14	PC14 PD14 PF14

TRGM signal name	Pin
P_15	PC15
	PD15
	PF15

Table 133: TRGM SIGNAL PIN

UART0 signal name	Pin
CTS	PA03
	PC03
	PE03
	PV03
DE	PA02
	PC02
	PE02
	PV02
RTS	PA02
	PC02
	PE02
	PV02
RXD	PA01
	PC01
	PE01
	PV01
TXD	PA00
	PC00
	PE00
	PV00

Table 134: UART0 SIGNAL PIN

UART1 signal name	Pin
CTS	PA04
	PC04
	PE04
	PV04
DE	PA05
	PC05
	PE05
	PV05
RTS	PA05
	PC05
	PE05

UART1 signal name	Pin
	PV05
RXD	PA06
	PC06
	PV06
TXD	PA07
	PC07
	PV07

Table 135: UART1 SIGNAL PIN

UART2 signal name	Pin
CTS	PC11
	PV11
DE	PC10
	PV10
RTS	PC10
	PV10
RXD	PA09
	PC09
	PV09
TXD	PA08
	PC08
	PV08

Table 136: UART2 SIGNAL PIN

UART3 signal name	Pin
CTS	PC12
DE	PC13
RTS	PC13
RXD	PC14
TXD	PC15

Table 137: UART3 SIGNAL PIN

UART4 signal name	Pin
CTS	PA19
	PC19
	PX03
DE	PA18
	PC18
	PX02

UART4 signal name	Pin
RTS	PA18
	PC18
	PX02
RXD	PA17
	PC17
	PX01
TXD	PA16
	PC16
	PX00

Table 138: UART4 SIGNAL PIN

UART5 signal name	Pin
CTS	PA20
	PC20
	PX04
DE	PA21
	PC21
	PX05
RTS	PA21
	PC21
	PX05
RXD	PA22
	PC22
	PX06
TXD	PA23
	PC23
	PX07

Table 139: UART5 SIGNAL PIN

UART6 signal name	Pin
CTS	PA27
	PC27
DE	PA26
	PC26
RTS	PA26
	PC26
RXD	PA25
	PC25
TXD	PA24
	PC24

UART6 signal name	Pin

Table 140: UART6 SIGNAL PIN

UART7 signal name	Pin
CTS	PA28
	PC28
DE	PA29
	PC29
RTS	PA29
	PC29
RXD	PA30
	PC30
TXD	PA31
	PC31

Table 141: UART7 SIGNAL PIN

USB0 signal name	Pin
ID	PE00
	PF21
	PF22
OC	PE03
	PF20
	PF23
PWR	PE01
	PF18
	PF19

Table 142: USB0 SIGNAL PIN

XPI0 High Speed Group	High speed signal name	Pin
A	CA_CS0	PB30
	CA_CS1	PB24
	CA_DQS	PB25
	CA_D_0	PB27
	CA_D_1	PB28
	CA_D_2	PB26
	CA_D_3	PB31
	CA_SCLK	PB29
	CB_CS0	PC00
	CB_CS1	PC09
	CB_DQS	PC11

XPI0 High Speed Group	High speed signal name	Pin
	CB_D_0	PC10
	CB_D_1	PC01
	CB_D_2	PC02
	CB_D_3	PC03
	CB_SCLK	PC04
B	CA_CS0	PB30
	CA_CS1	PB24
	CA_DQS	PX04
	CA_D_0	PB27
	CA_D_1	PB28
	CA_D_2	PB26
	CA_D_3	PB31
	CA_SCLK	PB29
	CB_CS0	PC00
	CB_CS1	PC09
	CB_DQS	PX00
	CB_D_0	PC10
	CB_D_1	PC01
	CB_D_2	PC02
	CB_D_3	PC03
	CB_SCLK	PC04
C	CA_CS0	PX05
	CA_DQS	PX04
	CA_D_0	PX01
	CA_D_1	PX06
	CA_D_2	PX07
	CA_D_3	PX03
	CA_SCLK	PX02
D	CA_CS0	PC05
	CA_CS1	PC13
	CA_DQS	PX04
	CA_D_0	PC15
	CA_D_1	PC06
	CA_D_2	PC12
	CA_D_3	PC07
	CA_SCLK	PC08
E	CA_CS0	PC05
	CA_CS1	PC13
	CA_DQS	PC16
	CA_D_0	PC15

XPI0 High Speed Group	High speed signal name	Pin
	CA_D_1	PC06
	CA_D_2	PC12
	CA_D_3	PC07
	CA_SCLK	PC08

Table 143: XPI0 HIGH SPEED SIGNAL PIN

19.3 Special Function Pins

By default, the chip selects three different boot modes through the BOOT_MODE[0:1]=[PA02:PA03] pins. The boot configuration is shown in [Table 144](#). Other special pin configurations are shown in [Table 145](#).

Boot Mode Selection Pins		Boot Mode	Description
BOOT_MODE1	BOOT_MODE0		
0	0	XPI NOR Boot	Boot from the serial NOR FLASH connected to XPI0/1
0	1	In-System Programming (ISP)/Serial Boot	Burn firmware or OTP from UART0/USB0, or boot from UART0/USB0
1	0	In-System Programming (ISP)/Serial Boot	Burn firmware or OTP from UART0/USB0, or boot from UART0/USB0
1	1	Reserved Mode	Reserved mode

Table 144: Boot Configuration Table

Pin Name	Description	Recommended Usage
XTAL_IN	24MHz clock input	Connect to 24MHz crystal or active clock
XTAL_OUT	24MHz clock output	Connect to 24MHz crystal or leave floating

Table 145: Special Function Pin Configuration

19.4 IO reset status

[Table 146](#) list the status of all IOs after reset:

Name	pin	Post-reset status
JTAG.TDO	PA04	high-impedance
JTAG.TDI	PA05	Input Internal pull-up
JTAG.TCK	PA06	Input Internal Pull-Down
JTAG.TMS	PA07	Input Internal Pull-Up
JTAG.TRST	PA08	Input Internal Pull-Up
GPIO	Other IO	Input High Impedance

Table 146: IO reset status table

20 Input and Output Module

This chapter introduces the input and output IO modules, including the general IO control module IOC, GPIO controller, fast GPIO controller and GPIO management.

20.1 IO Controller

IOC can control general-purpose IO (PA, PB, PC, PD, PE, PF, PX).

IOC supports any IO's configuration, such as open-drain control, internal pull-up and down control, Schmitt trigger, slew rate, drive strength, etc. You can also configure peripheral reuse function mapping, analog input and IO status monitoring functions. The peripheral reuse function of general IO is configured by IOC, which shown in [Figure 20](#).

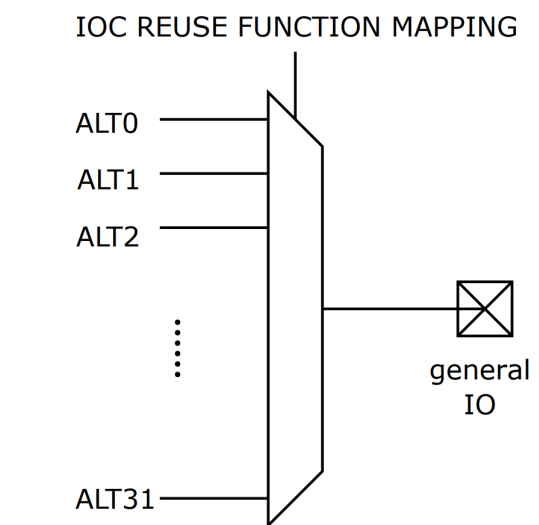


Figure 20: General IO peripheral multiplexing function selection

20.2 GPIO Controller

GPIO controllers include: GPIO controller (GPIO0), fast GPIO controller (FGPIO0).

GPIO0 and FGPIO0 can control general purpose IO (PA, PB, PC, PD, PE, PF, PX). For any IO, GPIOM determines which controller takes effect.

GPIO controller GPIO0 and fast GPIO controller FGPIO0 can control general-purpose IO (PA, PB, PC, PD, PE, PF, PX). The control of

general IO is like [Figure 21](#).

The Fast GPIO Controller (FGPIO0) is a private peripheral of CPU0 and can only be accessed by the RISC-V processor. The processor accesses its own fast GPIO controller with zero wait states.

The GPIO controller has the same basic functions as the fast GPIO controller. You can read inputs according to the IO port, configure the IO direction (input or output), set IO output, or configure one or more IO outputs to high, low or flip at the same time.

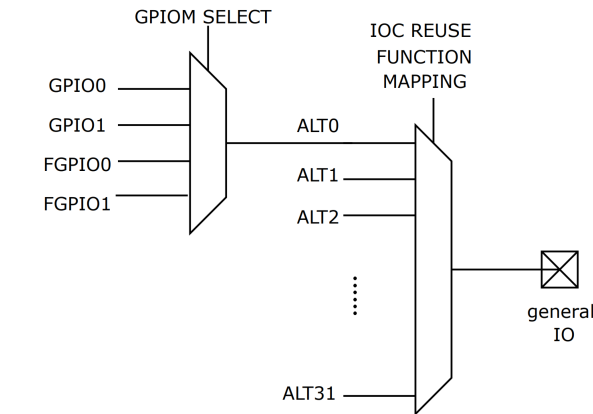


Figure 21: General IO GPIO control options

The GPIO controller supports configuring and generating GPIO interrupts. Fast GPIO controllers do not support interrupts.

20.3 GPIO Management GPIOM

The GPIO manager is used to manage all IO. Its main function is to configure the management authority for IOs. GPIOM selects either the fast GPIO controller or the GPIO controller for each IO to serve as the controller for that IO.

GPIOM supports configuring the input visibility of any IO, that is, whether the input registers of the fast GPIO controller and the GPIO controller can read the IO's input.

After reset, GPIOM assigns GPIO0 to control all general purpose IOs by default. Both GPIO controller and fast GPIO controller can read the IO input.

21 IO Controller IOC

This chapter describes the main features and functions of the IO controller IOC.

21.1 Feature Summary

IO controller IOC are as follows:

- Peripheral reuse function mapping
- Output loopback control
- Analog function configuration
- Slew rate configuration
- Open drain settings
- Schmitt Trigger configuration
- pull-up/pull-down configuration
- Drive strength configuration

21.2 Function Description

This section describes the functions of IOC.

21.2.1 IO basic configuration

IOC can be used to configure basic IO attributes, including IO open-drain selection, Schmitt trigger switches, slew rate, internal pull-up and pull-down resistors and drive strength. Users can configure the basic attributes of any IO through the IOC_X_PAD_CTL register.

All IOs only support 3.3V operating voltage.

IO open-drain selection means that the user can configure IO as open drain output. If IO is configured as an open-drain output, then when the output level is low, the output will be normal; when the output level is high, IO will output a high resistance other than high level, which requires users to add an external pull-up resistor.

The Schmitt trigger of IO means that the user can turn on the input Schmitt trigger of IO, that is, turn on the input hysteresis to increase the noise immunity.

IO can also turn on the internal pull-up resistor, which can be configured as internal pull-up or pull-down.

IO supports configuring slew rates.

IO supports configuration drive strength.

21.2.2 IO peripheral function configuration

Users can configure IO's peripheral functions through IOC's IOC_X_FUNC_CTL register, including output loopback function, analog function and peripheral function mapping.

After the user turns on the output loopback function of IO, the output signal can be read at the input terminal.

After turning on the analog function of IO, the IO can be used as pins for analog peripherals, such as ADC, ACMP, etc. It should be noted that when a pin is selected as an analog function through the ANALOG bit field configuration of the IOC_X_FUNC_CTL register, the digital input and output function of the pin will be automatically masked, including turning off the pin digital input and output enable (OE), pull-up and pull-down enable (PE),

and keeper enable (KE) to reduce the impact on the input and output of the analog module. Of course, when you need to use the digital output as the analog input on the same pin, (For example, when measuring the GPIO output voltage value of this pin through the ADC module), or when the analog output is used as the digital input (for example, the GPIO reads the approximate output of the DAC module, i.e., high/low levels), there is no need to enable the ANALOG bit field of the IOC_X_FUNC_CTL register, but the digital function of this pin needs to be configured in advance to avoid situations where the digital and analog modules generate outputs at the same time.

Users can select the peripheral functions mapped to this IO through the IO's peripheral function checker. For a mapping table of IO and peripheral functions, please refer to the relevant chapter of PINMUX.

21.3 IOC register

21.3.1 register description

IOC is as follows:

IOC base address: 0xF4040000

PIOC base address: 0xF4118000

Offset	Register Name	Description	Reset Value
0x0000	PAD[PA00][FUNC_CTL]	ALT SELECT	0x00000000
0x0004	PAD[PA00][PAD_CTL]	PAD SETTINGS	0x01010056
0x0008	PAD[PA01][FUNC_CTL]	ALT SELECT	0x00000000
0x000C	PAD[PA01][PAD_CTL]	PAD SETTINGS	0x01010056
0x0010	PAD[PA02][FUNC_CTL]	ALT SELECT	0x00000000
0x0014	PAD[PA02][PAD_CTL]	PAD SETTINGS	0x01010056
0x0018	PAD[PA03][FUNC_CTL]	ALT SELECT	0x00000000
0x001C	PAD[PA03][PAD_CTL]	PAD SETTINGS	0x01010056
0x0020	PAD[PA04][FUNC_CTL]	ALT SELECT	0x00000000
0x0024	PAD[PA04][PAD_CTL]	PAD SETTINGS	0x01010056
0x0028	PAD[PA05][FUNC_CTL]	ALT SELECT	0x00000000
0x002C	PAD[PA05][PAD_CTL]	PAD SETTINGS	0x01010056
0x0030	PAD[PA06][FUNC_CTL]	ALT SELECT	0x00000000
0x0034	PAD[PA06][PAD_CTL]	PAD SETTINGS	0x01010056
0x0038	PAD[PA07][FUNC_CTL]	ALT SELECT	0x00000000
0x003C	PAD[PA07][PAD_CTL]	PAD SETTINGS	0x01010056
0x0040	PAD[PA08][FUNC_CTL]	ALT SELECT	0x00000000
0x0044	PAD[PA08][PAD_CTL]	PAD SETTINGS	0x01010056
0x0048	PAD[PA09][FUNC_CTL]	ALT SELECT	0x00000000
0x004C	PAD[PA09][PAD_CTL]	PAD SETTINGS	0x01010056
0x0080	PAD[PA16][FUNC_CTL]	ALT SELECT	0x00000000
0x0084	PAD[PA16][PAD_CTL]	PAD SETTINGS	0x01010056
0x0088	PAD[PA17][FUNC_CTL]	ALT SELECT	0x00000000
0x008C	PAD[PA17][PAD_CTL]	PAD SETTINGS	0x01010056

Offset	Register Name	Description	Reset Value
0x0090	PAD[PA18][FUNC_CTL]	ALT SELECT	0x00000000
0x0094	PAD[PA18][PAD_CTL]	PAD SETTINGS	0x01010056
0x0098	PAD[PA19][FUNC_CTL]	ALT SELECT	0x00000000
0x009C	PAD[PA19][PAD_CTL]	PAD SETTINGS	0x01010056
0x00A0	PAD[PA20][FUNC_CTL]	ALT SELECT	0x00000000
0x00A4	PAD[PA20][PAD_CTL]	PAD SETTINGS	0x01010056
0x00A8	PAD[PA21][FUNC_CTL]	ALT SELECT	0x00000000
0x00AC	PAD[PA21][PAD_CTL]	PAD SETTINGS	0x01010056
0x00B0	PAD[PA22][FUNC_CTL]	ALT SELECT	0x00000000
0x00B4	PAD[PA22][PAD_CTL]	PAD SETTINGS	0x01010056
0x00B8	PAD[PA23][FUNC_CTL]	ALT SELECT	0x00000000
0x00BC	PAD[PA23][PAD_CTL]	PAD SETTINGS	0x01010056
0x00C0	PAD[PA24][FUNC_CTL]	ALT SELECT	0x00000000
0x00C4	PAD[PA24][PAD_CTL]	PAD SETTINGS	0x01010056
0x00C8	PAD[PA25][FUNC_CTL]	ALT SELECT	0x00000000
0x00CC	PAD[PA25][PAD_CTL]	PAD SETTINGS	0x01010056
0x00D0	PAD[PA26][FUNC_CTL]	ALT SELECT	0x00000000
0x00D4	PAD[PA26][PAD_CTL]	PAD SETTINGS	0x01010056
0x00D8	PAD[PA27][FUNC_CTL]	ALT SELECT	0x00000000
0x00DC	PAD[PA27][PAD_CTL]	PAD SETTINGS	0x01010056
0x00E0	PAD[PA28][FUNC_CTL]	ALT SELECT	0x00000000
0x00E4	PAD[PA28][PAD_CTL]	PAD SETTINGS	0x01010056
0x00E8	PAD[PA29][FUNC_CTL]	ALT SELECT	0x00000000
0x00EC	PAD[PA29][PAD_CTL]	PAD SETTINGS	0x01010056
0x00F0	PAD[PA30][FUNC_CTL]	ALT SELECT	0x00000000
0x00F4	PAD[PA30][PAD_CTL]	PAD SETTINGS	0x01010056
0x00F8	PAD[PA31][FUNC_CTL]	ALT SELECT	0x00000000
0x00FC	PAD[PA31][PAD_CTL]	PAD SETTINGS	0x01010056
0x0100	PAD[PB00][FUNC_CTL]	ALT SELECT	0x00000000
0x0104	PAD[PB00][PAD_CTL]	PAD SETTINGS	0x01010056
0x0108	PAD[PB01][FUNC_CTL]	ALT SELECT	0x00000000
0x010C	PAD[PB01][PAD_CTL]	PAD SETTINGS	0x01010056
0x0110	PAD[PB02][FUNC_CTL]	ALT SELECT	0x00000000
0x0114	PAD[PB02][PAD_CTL]	PAD SETTINGS	0x01010056
0x0118	PAD[PB03][FUNC_CTL]	ALT SELECT	0x00000000
0x011C	PAD[PB03][PAD_CTL]	PAD SETTINGS	0x01010056
0x0120	PAD[PB04][FUNC_CTL]	ALT SELECT	0x00000000
0x0124	PAD[PB04][PAD_CTL]	PAD SETTINGS	0x01010056
0x0128	PAD[PB05][FUNC_CTL]	ALT SELECT	0x00000000
0x012C	PAD[PB05][PAD_CTL]	PAD SETTINGS	0x01010056

Offset	Register Name	Description	Reset Value
0x0130	PAD[PB06][FUNC_CTL]	ALT SELECT	0x00000000
0x0134	PAD[PB06][PAD_CTL]	PAD SETTINGS	0x01010056
0x0138	PAD[PB07][FUNC_CTL]	ALT SELECT	0x00000000
0x013C	PAD[PB07][PAD_CTL]	PAD SETTINGS	0x01010056
0x0140	PAD[PB08][FUNC_CTL]	ALT SELECT	0x00000000
0x0144	PAD[PB08][PAD_CTL]	PAD SETTINGS	0x01010056
0x0148	PAD[PB09][FUNC_CTL]	ALT SELECT	0x00000000
0x014C	PAD[PB09][PAD_CTL]	PAD SETTINGS	0x01010056
0x0150	PAD[PB10][FUNC_CTL]	ALT SELECT	0x00000000
0x0154	PAD[PB10][PAD_CTL]	PAD SETTINGS	0x01010056
0x0158	PAD[PB11][FUNC_CTL]	ALT SELECT	0x00000000
0x015C	PAD[PB11][PAD_CTL]	PAD SETTINGS	0x01010056
0x01C0	PAD[PB24][FUNC_CTL]	ALT SELECT	0x00000000
0x01C4	PAD[PB24][PAD_CTL]	PAD SETTINGS	0x01010056
0x01C8	PAD[PB25][FUNC_CTL]	ALT SELECT	0x00000000
0x01CC	PAD[PB25][PAD_CTL]	PAD SETTINGS	0x01010056
0x01D0	PAD[PB26][FUNC_CTL]	ALT SELECT	0x00000000
0x01D4	PAD[PB26][PAD_CTL]	PAD SETTINGS	0x01010056
0x01D8	PAD[PB27][FUNC_CTL]	ALT SELECT	0x00000000
0x01DC	PAD[PB27][PAD_CTL]	PAD SETTINGS	0x01010056
0x01E0	PAD[PB28][FUNC_CTL]	ALT SELECT	0x00000000
0x01E4	PAD[PB28][PAD_CTL]	PAD SETTINGS	0x01010056
0x01E8	PAD[PB29][FUNC_CTL]	ALT SELECT	0x00000000
0x01EC	PAD[PB29][PAD_CTL]	PAD SETTINGS	0x01010056
0x01F0	PAD[PB30][FUNC_CTL]	ALT SELECT	0x00000000
0x01F4	PAD[PB30][PAD_CTL]	PAD SETTINGS	0x01010056
0x01F8	PAD[PB31][FUNC_CTL]	ALT SELECT	0x00000000
0x01FC	PAD[PB31][PAD_CTL]	PAD SETTINGS	0x01010056
0x0200	PAD[PC00][FUNC_CTL]	ALT SELECT	0x00000000
0x0204	PAD[PC00][PAD_CTL]	PAD SETTINGS	0x01010056
0x0208	PAD[PC01][FUNC_CTL]	ALT SELECT	0x00000000
0x020C	PAD[PC01][PAD_CTL]	PAD SETTINGS	0x01010056
0x0210	PAD[PC02][FUNC_CTL]	ALT SELECT	0x00000000
0x0214	PAD[PC02][PAD_CTL]	PAD SETTINGS	0x01010056
0x0218	PAD[PC03][FUNC_CTL]	ALT SELECT	0x00000000
0x021C	PAD[PC03][PAD_CTL]	PAD SETTINGS	0x01010056
0x0220	PAD[PC04][FUNC_CTL]	ALT SELECT	0x00000000
0x0224	PAD[PC04][PAD_CTL]	PAD SETTINGS	0x01010056
0x0228	PAD[PC05][FUNC_CTL]	ALT SELECT	0x00000000
0x022C	PAD[PC05][PAD_CTL]	PAD SETTINGS	0x01010056

Offset	Register Name	Description	Reset Value
0x0230	PAD[PC06][FUNC_CTL]	ALT SELECT	0x00000000
0x0234	PAD[PC06][PAD_CTL]	PAD SETTINGS	0x01010056
0x0238	PAD[PC07][FUNC_CTL]	ALT SELECT	0x00000000
0x023C	PAD[PC07][PAD_CTL]	PAD SETTINGS	0x01010056
0x0240	PAD[PC08][FUNC_CTL]	ALT SELECT	0x00000000
0x0244	PAD[PC08][PAD_CTL]	PAD SETTINGS	0x01010056
0x0248	PAD[PC09][FUNC_CTL]	ALT SELECT	0x00000000
0x024C	PAD[PC09][PAD_CTL]	PAD SETTINGS	0x01010056
0x0250	PAD[PC10][FUNC_CTL]	ALT SELECT	0x00000000
0x0254	PAD[PC10][PAD_CTL]	PAD SETTINGS	0x01010056
0x0258	PAD[PC11][FUNC_CTL]	ALT SELECT	0x00000000
0x025C	PAD[PC11][PAD_CTL]	PAD SETTINGS	0x01010056
0x0260	PAD[PC12][FUNC_CTL]	ALT SELECT	0x00000000
0x0264	PAD[PC12][PAD_CTL]	PAD SETTINGS	0x01010056
0x0268	PAD[PC13][FUNC_CTL]	ALT SELECT	0x00000000
0x026C	PAD[PC13][PAD_CTL]	PAD SETTINGS	0x01010056
0x0270	PAD[PC14][FUNC_CTL]	ALT SELECT	0x00000000
0x0274	PAD[PC14][PAD_CTL]	PAD SETTINGS	0x01010056
0x0278	PAD[PC15][FUNC_CTL]	ALT SELECT	0x00000000
0x027C	PAD[PC15][PAD_CTL]	PAD SETTINGS	0x01010056
0x0280	PAD[PC16][FUNC_CTL]	ALT SELECT	0x00000000
0x0284	PAD[PC16][PAD_CTL]	PAD SETTINGS	0x01010056
0x0288	PAD[PC17][FUNC_CTL]	ALT SELECT	0x00000000
0x028C	PAD[PC17][PAD_CTL]	PAD SETTINGS	0x01010056
0x0290	PAD[PC18][FUNC_CTL]	ALT SELECT	0x00000000
0x0294	PAD[PC18][PAD_CTL]	PAD SETTINGS	0x01010056
0x0298	PAD[PC19][FUNC_CTL]	ALT SELECT	0x00000000
0x029C	PAD[PC19][PAD_CTL]	PAD SETTINGS	0x01010056
0x02A0	PAD[PC20][FUNC_CTL]	ALT SELECT	0x00000000
0x02A4	PAD[PC20][PAD_CTL]	PAD SETTINGS	0x01010056
0x02A8	PAD[PC21][FUNC_CTL]	ALT SELECT	0x00000000
0x02AC	PAD[PC21][PAD_CTL]	PAD SETTINGS	0x01010056
0x02B0	PAD[PC22][FUNC_CTL]	ALT SELECT	0x00000000
0x02B4	PAD[PC22][PAD_CTL]	PAD SETTINGS	0x01010056
0x02B8	PAD[PC23][FUNC_CTL]	ALT SELECT	0x00000000
0x02BC	PAD[PC23][PAD_CTL]	PAD SETTINGS	0x01010056
0x02C0	PAD[PC24][FUNC_CTL]	ALT SELECT	0x00000000
0x02C4	PAD[PC24][PAD_CTL]	PAD SETTINGS	0x01010056
0x02C8	PAD[PC25][FUNC_CTL]	ALT SELECT	0x00000000
0x02CC	PAD[PC25][PAD_CTL]	PAD SETTINGS	0x01010056

Offset	Register Name	Description	Reset Value
0x02D0	PAD[PC26][FUNC_CTL]	ALT SELECT	0x00000000
0x02D4	PAD[PC26][PAD_CTL]	PAD SETTINGS	0x01010056
0x02D8	PAD[PC27][FUNC_CTL]	ALT SELECT	0x00000000
0x02DC	PAD[PC27][PAD_CTL]	PAD SETTINGS	0x01010056
0x02E0	PAD[PC28][FUNC_CTL]	ALT SELECT	0x00000000
0x02E4	PAD[PC28][PAD_CTL]	PAD SETTINGS	0x01010056
0x02E8	PAD[PC29][FUNC_CTL]	ALT SELECT	0x00000000
0x02EC	PAD[PC29][PAD_CTL]	PAD SETTINGS	0x01010056
0x02F0	PAD[PC30][FUNC_CTL]	ALT SELECT	0x00000000
0x02F4	PAD[PC30][PAD_CTL]	PAD SETTINGS	0x01010056
0x02F8	PAD[PC31][FUNC_CTL]	ALT SELECT	0x00000000
0x02FC	PAD[PC31][PAD_CTL]	PAD SETTINGS	0x01010056
0x0300	PAD[PD00][FUNC_CTL]	ALT SELECT	0x00000000
0x0304	PAD[PD00][PAD_CTL]	PAD SETTINGS	0x01010056
0x0308	PAD[PD01][FUNC_CTL]	ALT SELECT	0x00000000
0x030C	PAD[PD01][PAD_CTL]	PAD SETTINGS	0x01010056
0x0310	PAD[PD02][FUNC_CTL]	ALT SELECT	0x00000000
0x0314	PAD[PD02][PAD_CTL]	PAD SETTINGS	0x01010056
0x0318	PAD[PD03][FUNC_CTL]	ALT SELECT	0x00000000
0x031C	PAD[PD03][PAD_CTL]	PAD SETTINGS	0x01010056
0x0320	PAD[PD04][FUNC_CTL]	ALT SELECT	0x00000000
0x0324	PAD[PD04][PAD_CTL]	PAD SETTINGS	0x01010056
0x0328	PAD[PD05][FUNC_CTL]	ALT SELECT	0x00000000
0x032C	PAD[PD05][PAD_CTL]	PAD SETTINGS	0x01010056
0x0330	PAD[PD06][FUNC_CTL]	ALT SELECT	0x00000000
0x0334	PAD[PD06][PAD_CTL]	PAD SETTINGS	0x01010056
0x0338	PAD[PD07][FUNC_CTL]	ALT SELECT	0x00000000
0x033C	PAD[PD07][PAD_CTL]	PAD SETTINGS	0x01010056
0x0340	PAD[PD08][FUNC_CTL]	ALT SELECT	0x00000000
0x0344	PAD[PD08][PAD_CTL]	PAD SETTINGS	0x01010056
0x0348	PAD[PD09][FUNC_CTL]	ALT SELECT	0x00000000
0x034C	PAD[PD09][PAD_CTL]	PAD SETTINGS	0x01010056
0x0350	PAD[PD10][FUNC_CTL]	ALT SELECT	0x00000000
0x0354	PAD[PD10][PAD_CTL]	PAD SETTINGS	0x01010056
0x0358	PAD[PD11][FUNC_CTL]	ALT SELECT	0x00000000
0x035C	PAD[PD11][PAD_CTL]	PAD SETTINGS	0x01010056
0x0360	PAD[PD12][FUNC_CTL]	ALT SELECT	0x00000000
0x0364	PAD[PD12][PAD_CTL]	PAD SETTINGS	0x01010056
0x0368	PAD[PD13][FUNC_CTL]	ALT SELECT	0x00000000
0x036C	PAD[PD13][PAD_CTL]	PAD SETTINGS	0x01010056

Offset	Register Name	Description	Reset Value
0x0370	PAD[PD14][FUNC_CTL]	ALT SELECT	0x00000000
0x0374	PAD[PD14][PAD_CTL]	PAD SETTINGS	0x01010056
0x0378	PAD[PD15][FUNC_CTL]	ALT SELECT	0x00000000
0x037C	PAD[PD15][PAD_CTL]	PAD SETTINGS	0x01010056
0x0380	PAD[PD16][FUNC_CTL]	ALT SELECT	0x00000000
0x0384	PAD[PD16][PAD_CTL]	PAD SETTINGS	0x01010056
0x0388	PAD[PD17][FUNC_CTL]	ALT SELECT	0x00000000
0x038C	PAD[PD17][PAD_CTL]	PAD SETTINGS	0x01010056
0x0390	PAD[PD18][FUNC_CTL]	ALT SELECT	0x00000000
0x0394	PAD[PD18][PAD_CTL]	PAD SETTINGS	0x01010056
0x0398	PAD[PD19][FUNC_CTL]	ALT SELECT	0x00000000
0x039C	PAD[PD19][PAD_CTL]	PAD SETTINGS	0x01010056
0x03A0	PAD[PD20][FUNC_CTL]	ALT SELECT	0x00000000
0x03A4	PAD[PD20][PAD_CTL]	PAD SETTINGS	0x01010056
0x03A8	PAD[PD21][FUNC_CTL]	ALT SELECT	0x00000000
0x03AC	PAD[PD21][PAD_CTL]	PAD SETTINGS	0x01010056
0x03B0	PAD[PD22][FUNC_CTL]	ALT SELECT	0x00000000
0x03B4	PAD[PD22][PAD_CTL]	PAD SETTINGS	0x01010056
0x03B8	PAD[PD23][FUNC_CTL]	ALT SELECT	0x00000000
0x03BC	PAD[PD23][PAD_CTL]	PAD SETTINGS	0x01010056
0x03C0	PAD[PD24][FUNC_CTL]	ALT SELECT	0x00000000
0x03C4	PAD[PD24][PAD_CTL]	PAD SETTINGS	0x01010056
0x03C8	PAD[PD25][FUNC_CTL]	ALT SELECT	0x00000000
0x03CC	PAD[PD25][PAD_CTL]	PAD SETTINGS	0x01010056
0x03D0	PAD[PD26][FUNC_CTL]	ALT SELECT	0x00000000
0x03D4	PAD[PD26][PAD_CTL]	PAD SETTINGS	0x01010056
0x03D8	PAD[PD27][FUNC_CTL]	ALT SELECT	0x00000000
0x03DC	PAD[PD27][PAD_CTL]	PAD SETTINGS	0x01010056
0x03E0	PAD[PD28][FUNC_CTL]	ALT SELECT	0x00000000
0x03E4	PAD[PD28][PAD_CTL]	PAD SETTINGS	0x01010056
0x03E8	PAD[PD29][FUNC_CTL]	ALT SELECT	0x00000000
0x03EC	PAD[PD29][PAD_CTL]	PAD SETTINGS	0x01010056
0x03F0	PAD[PD30][FUNC_CTL]	ALT SELECT	0x00000000
0x03F4	PAD[PD30][PAD_CTL]	PAD SETTINGS	0x01010056
0x03F8	PAD[PD31][FUNC_CTL]	ALT SELECT	0x00000000
0x03FC	PAD[PD31][PAD_CTL]	PAD SETTINGS	0x01010056
0x0400	PAD[PE00][FUNC_CTL]	ALT SELECT	0x00000000
0x0404	PAD[PE00][PAD_CTL]	PAD SETTINGS	0x01010056
0x0408	PAD[PE01][FUNC_CTL]	ALT SELECT	0x00000000
0x040C	PAD[PE01][PAD_CTL]	PAD SETTINGS	0x01010056

Offset	Register Name	Description	Reset Value
0x0410	PAD[PE02][FUNC_CTL]	ALT SELECT	0x00000000
0x0414	PAD[PE02][PAD_CTL]	PAD SETTINGS	0x01010056
0x0418	PAD[PE03][FUNC_CTL]	ALT SELECT	0x00000000
0x041C	PAD[PE03][PAD_CTL]	PAD SETTINGS	0x01010056
0x0420	PAD[PE04][FUNC_CTL]	ALT SELECT	0x00000000
0x0424	PAD[PE04][PAD_CTL]	PAD SETTINGS	0x01010056
0x0428	PAD[PE05][FUNC_CTL]	ALT SELECT	0x00000000
0x042C	PAD[PE05][PAD_CTL]	PAD SETTINGS	0x01010056
0x0500	PAD[PF00][FUNC_CTL]	ALT SELECT	0x00000000
0x0504	PAD[PF00][PAD_CTL]	PAD SETTINGS	0x01010056
0x0508	PAD[PF01][FUNC_CTL]	ALT SELECT	0x00000000
0x050C	PAD[PF01][PAD_CTL]	PAD SETTINGS	0x01010056
0x0510	PAD[PF02][FUNC_CTL]	ALT SELECT	0x00000000
0x0514	PAD[PF02][PAD_CTL]	PAD SETTINGS	0x01010056
0x0518	PAD[PF03][FUNC_CTL]	ALT SELECT	0x00000000
0x051C	PAD[PF03][PAD_CTL]	PAD SETTINGS	0x01010056
0x0520	PAD[PF04][FUNC_CTL]	ALT SELECT	0x00000000
0x0524	PAD[PF04][PAD_CTL]	PAD SETTINGS	0x01010056
0x0528	PAD[PF05][FUNC_CTL]	ALT SELECT	0x00000000
0x052C	PAD[PF05][PAD_CTL]	PAD SETTINGS	0x01010056
0x0530	PAD[PF06][FUNC_CTL]	ALT SELECT	0x00000000
0x0534	PAD[PF06][PAD_CTL]	PAD SETTINGS	0x01010056
0x0538	PAD[PF07][FUNC_CTL]	ALT SELECT	0x00000000
0x053C	PAD[PF07][PAD_CTL]	PAD SETTINGS	0x01010056
0x0540	PAD[PF08][FUNC_CTL]	ALT SELECT	0x00000000
0x0544	PAD[PF08][PAD_CTL]	PAD SETTINGS	0x01010056
0x0548	PAD[PF09][FUNC_CTL]	ALT SELECT	0x00000000
0x054C	PAD[PF09][PAD_CTL]	PAD SETTINGS	0x01010056
0x0550	PAD[PF10][FUNC_CTL]	ALT SELECT	0x00000000
0x0554	PAD[PF10][PAD_CTL]	PAD SETTINGS	0x01010056
0x0558	PAD[PF11][FUNC_CTL]	ALT SELECT	0x00000000
0x055C	PAD[PF11][PAD_CTL]	PAD SETTINGS	0x01010056
0x0560	PAD[PF12][FUNC_CTL]	ALT SELECT	0x00000000
0x0564	PAD[PF12][PAD_CTL]	PAD SETTINGS	0x01010056
0x0568	PAD[PF13][FUNC_CTL]	ALT SELECT	0x00000000
0x056C	PAD[PF13][PAD_CTL]	PAD SETTINGS	0x01010056
0x0570	PAD[PF14][FUNC_CTL]	ALT SELECT	0x00000000
0x0574	PAD[PF14][PAD_CTL]	PAD SETTINGS	0x01010056
0x0578	PAD[PF15][FUNC_CTL]	ALT SELECT	0x00000000
0x057C	PAD[PF15][PAD_CTL]	PAD SETTINGS	0x01010056

Offset	Register Name	Description	Reset Value
0x0580	PAD[PF16][FUNC_CTL]	ALT SELECT	0x00000000
0x0584	PAD[PF16][PAD_CTL]	PAD SETTINGS	0x01010056
0x0588	PAD[PF17][FUNC_CTL]	ALT SELECT	0x00000000
0x058C	PAD[PF17][PAD_CTL]	PAD SETTINGS	0x01010056
0x0590	PAD[PF18][FUNC_CTL]	ALT SELECT	0x00000000
0x0594	PAD[PF18][PAD_CTL]	PAD SETTINGS	0x01010056
0x0598	PAD[PF19][FUNC_CTL]	ALT SELECT	0x00000000
0x059C	PAD[PF19][PAD_CTL]	PAD SETTINGS	0x01010056
0x05A0	PAD[PF20][FUNC_CTL]	ALT SELECT	0x00000000
0x05A4	PAD[PF20][PAD_CTL]	PAD SETTINGS	0x01010056
0x05A8	PAD[PF21][FUNC_CTL]	ALT SELECT	0x00000000
0x05AC	PAD[PF21][PAD_CTL]	PAD SETTINGS	0x01010056
0x05B0	PAD[PF22][FUNC_CTL]	ALT SELECT	0x00000000
0x05B4	PAD[PF22][PAD_CTL]	PAD SETTINGS	0x01010056
0x05B8	PAD[PF23][FUNC_CTL]	ALT SELECT	0x00000000
0x05BC	PAD[PF23][PAD_CTL]	PAD SETTINGS	0x01010056
0x05C0	PAD[PF24][FUNC_CTL]	ALT SELECT	0x00000000
0x05C4	PAD[PF24][PAD_CTL]	PAD SETTINGS	0x01010056
0x05C8	PAD[PF25][FUNC_CTL]	ALT SELECT	0x00000000
0x05CC	PAD[PF25][PAD_CTL]	PAD SETTINGS	0x01010056
0x05D0	PAD[PF26][FUNC_CTL]	ALT SELECT	0x00000000
0x05D4	PAD[PF26][PAD_CTL]	PAD SETTINGS	0x01010056
0x05D8	PAD[PF27][FUNC_CTL]	ALT SELECT	0x00000000
0x05DC	PAD[PF27][PAD_CTL]	PAD SETTINGS	0x01010056
0x05E0	PAD[PF28][FUNC_CTL]	ALT SELECT	0x00000000
0x05E4	PAD[PF28][PAD_CTL]	PAD SETTINGS	0x01010056
0x05E8	PAD[PF29][FUNC_CTL]	ALT SELECT	0x00000000
0x05EC	PAD[PF29][PAD_CTL]	PAD SETTINGS	0x01010056
0x05F0	PAD[PF30][FUNC_CTL]	ALT SELECT	0x00000000
0x05F4	PAD[PF30][PAD_CTL]	PAD SETTINGS	0x01010056
0x05F8	PAD[PF31][FUNC_CTL]	ALT SELECT	0x00000000
0x05FC	PAD[PF31][PAD_CTL]	PAD SETTINGS	0x01010056
0x0B00	PAD[PV00][FUNC_CTL]	ALT SELECT	0x00000000
0x0B04	PAD[PV00][PAD_CTL]	PAD SETTINGS	0x01010056
0x0B08	PAD[PV01][FUNC_CTL]	ALT SELECT	0x00000000
0x0B0C	PAD[PV01][PAD_CTL]	PAD SETTINGS	0x01010056
0x0B10	PAD[PV02][FUNC_CTL]	ALT SELECT	0x00000000
0x0B14	PAD[PV02][PAD_CTL]	PAD SETTINGS	0x01010056
0x0B18	PAD[PV03][FUNC_CTL]	ALT SELECT	0x00000000
0x0B1C	PAD[PV03][PAD_CTL]	PAD SETTINGS	0x01010056

Offset	Register Name	Description	Reset Value
0x0B20	PAD[PV04][FUNC_CTL]	ALT SELECT	0x00000000
0x0B24	PAD[PV04][PAD_CTL]	PAD SETTINGS	0x01010056
0x0B28	PAD[PV05][FUNC_CTL]	ALT SELECT	0x00000000
0x0B2C	PAD[PV05][PAD_CTL]	PAD SETTINGS	0x01010056
0x0B30	PAD[PV06][FUNC_CTL]	ALT SELECT	0x00000000
0x0B34	PAD[PV06][PAD_CTL]	PAD SETTINGS	0x01010056
0x0B38	PAD[PV07][FUNC_CTL]	ALT SELECT	0x00000000
0x0B3C	PAD[PV07][PAD_CTL]	PAD SETTINGS	0x01010056
0x0B40	PAD[PV08][FUNC_CTL]	ALT SELECT	0x00000000
0x0B44	PAD[PV08][PAD_CTL]	PAD SETTINGS	0x01010056
0x0B48	PAD[PV09][FUNC_CTL]	ALT SELECT	0x00000000
0x0B4C	PAD[PV09][PAD_CTL]	PAD SETTINGS	0x01010056
0x0B50	PAD[PV10][FUNC_CTL]	ALT SELECT	0x00000000
0x0B54	PAD[PV10][PAD_CTL]	PAD SETTINGS	0x01010056
0x0B58	PAD[PV11][FUNC_CTL]	ALT SELECT	0x00000000
0x0B5C	PAD[PV11][PAD_CTL]	PAD SETTINGS	0x01010056
0x0B60	PAD[PV12][FUNC_CTL]	ALT SELECT	0x00000000
0x0B64	PAD[PV12][PAD_CTL]	PAD SETTINGS	0x01010056
0x0B78	PAD[PV15][FUNC_CTL]	ALT SELECT	0x00000000
0x0B7C	PAD[PV15][PAD_CTL]	PAD SETTINGS	0x01010056
0x0C00	PAD[PW00][FUNC_CTL]	ALT SELECT	0x00000000
0x0C04	PAD[PW00][PAD_CTL]	PAD SETTINGS	0x01010056
0x0C08	PAD[PW01][FUNC_CTL]	ALT SELECT	0x00000000
0x0C0C	PAD[PW01][PAD_CTL]	PAD SETTINGS	0x01010056
0x0C10	PAD[PW02][FUNC_CTL]	ALT SELECT	0x00000000
0x0C14	PAD[PW02][PAD_CTL]	PAD SETTINGS	0x01010056
0x0C18	PAD[PW03][FUNC_CTL]	ALT SELECT	0x00000000
0x0C1C	PAD[PW03][PAD_CTL]	PAD SETTINGS	0x01010056
0x0C20	PAD[PW04][FUNC_CTL]	ALT SELECT	0x00000000
0x0C24	PAD[PW04][PAD_CTL]	PAD SETTINGS	0x01010056
0x0C28	PAD[PW05][FUNC_CTL]	ALT SELECT	0x00000000
0x0C2C	PAD[PW05][PAD_CTL]	PAD SETTINGS	0x01010056
0x0C30	PAD[PW06][FUNC_CTL]	ALT SELECT	0x00000000
0x0C34	PAD[PW06][PAD_CTL]	PAD SETTINGS	0x01010056
0x0C38	PAD[PW07][FUNC_CTL]	ALT SELECT	0x00000000
0x0C3C	PAD[PW07][PAD_CTL]	PAD SETTINGS	0x01010056
0x0C40	PAD[PW08][FUNC_CTL]	ALT SELECT	0x00000000
0x0C44	PAD[PW08][PAD_CTL]	PAD SETTINGS	0x01010056
0x0C48	PAD[PW09][FUNC_CTL]	ALT SELECT	0x00000000
0x0C4C	PAD[PW09][PAD_CTL]	PAD SETTINGS	0x01010056

Offset	Register Name	Description	Reset Value
0x0C50	PAD[PW10][FUNC_CTL]	ALT SELECT	0x00000000
0x0C54	PAD[PW10][PAD_CTL]	PAD SETTINGS	0x01010056
0x0C58	PAD[PW11][FUNC_CTL]	ALT SELECT	0x00000000
0x0C5C	PAD[PW11][PAD_CTL]	PAD SETTINGS	0x01010056
0x0C60	PAD[PW12][FUNC_CTL]	ALT SELECT	0x00000000
0x0C64	PAD[PW12][PAD_CTL]	PAD SETTINGS	0x01010056
0x0C78	PAD[PW15][FUNC_CTL]	ALT SELECT	0x00000000
0x0C7C	PAD[PW15][PAD_CTL]	PAD SETTINGS	0x01010056
0x0C80	PAD[PW16][FUNC_CTL]	ALT SELECT	0x00000000
0x0C84	PAD[PW16][PAD_CTL]	PAD SETTINGS	0x01010056
0x0C88	PAD[PW17][FUNC_CTL]	ALT SELECT	0x00000000
0x0C8C	PAD[PW17][PAD_CTL]	PAD SETTINGS	0x01010056
0x0CA0	PAD[PW20][FUNC_CTL]	ALT SELECT	0x00000000
0x0CA4	PAD[PW20][PAD_CTL]	PAD SETTINGS	0x01010056
0x0CA8	PAD[PW21][FUNC_CTL]	ALT SELECT	0x00000000
0x0CAC	PAD[PW21][PAD_CTL]	PAD SETTINGS	0x01010056
0x0D00	PAD[PX00][FUNC_CTL]	ALT SELECT	0x00000000
0x0D04	PAD[PX00][PAD_CTL]	PAD SETTINGS	0x01010056
0x0D08	PAD[PX01][FUNC_CTL]	ALT SELECT	0x00000000
0x0D0C	PAD[PX01][PAD_CTL]	PAD SETTINGS	0x01010056
0x0D10	PAD[PX02][FUNC_CTL]	ALT SELECT	0x00000000
0x0D14	PAD[PX02][PAD_CTL]	PAD SETTINGS	0x01010056
0x0D18	PAD[PX03][FUNC_CTL]	ALT SELECT	0x00000000
0x0D1C	PAD[PX03][PAD_CTL]	PAD SETTINGS	0x01010056
0x0D20	PAD[PX04][FUNC_CTL]	ALT SELECT	0x00000000
0x0D24	PAD[PX04][PAD_CTL]	PAD SETTINGS	0x01010056
0x0D28	PAD[PX05][FUNC_CTL]	ALT SELECT	0x00000000
0x0D2C	PAD[PX05][PAD_CTL]	PAD SETTINGS	0x01010056
0x0D30	PAD[PX06][FUNC_CTL]	ALT SELECT	0x00000000
0x0D34	PAD[PX06][PAD_CTL]	PAD SETTINGS	0x01010056
0x0D38	PAD[PX07][FUNC_CTL]	ALT SELECT	0x00000000
0x0D3C	PAD[PX07][PAD_CTL]	PAD SETTINGS	0x01010056

Table 147: IOC Register List

IOC are as follows:

21.3.2 PAD[FUNC_CTL] (0x0 + 0x8 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD															LOOP_BACK	RSVD						ANALOG	RSVD			ALT_SELECT					
N/A															RW	N/A						RW	N/A			RW					
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	x	x	x	x	x	x	x	0	x	x	x	0	0	0	0	0

PAD[FUNC_CTL] [31:0]

Bits	Name	Description
16	LOOP_BACK	force input on 0: disable 1: enable
8	ANALOG	select analog pin in pad 0: disable 1: enable
4-0	ALT_SELECT	alt select 0: ALT0 1: ALT1 ... 31:ALT31

PAD[FUNC_CTL] Bits

21.3.3 PAD[PAD_CTL] (0x4 + 0x8 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								HYS	RSVD		PRS	RSVD		PS	PE	KE	RSVD						OD	RSVD		SR	SPD		RSVD	DS	
N/A								RW	N/A		RW	N/A		RW	RW	RW	N/A						RW	N/A		RW	RW		N/A	RW	
x	x	x	x	x	x	x	x	1	x	x	0	0	x	0	0	1	x	x	x	x	x	x	0	x	1	0	1	x	1	1	0

PAD[PAD_CTL] [31:0]

Bits	Name	Description
24	HYS	schmitt trigger enable 0: disable 1: enable

Bits	Name	Description
21-20	PRS	select pull up/down internal resistance strength: For pull down, only have 100 Kohm resistance For pull up: 00: 100 KOhm 01: 47 KOhm 10: 22 KOhm 11: 22 KOhm
18	PS	pull select 0: pull down 1: pull up
17	PE	pull enable 0: pull disable 1: pull enable
16	KE	keeper capability enable 0: keeper disable 1: keeper enable
8	OD	open drain 0: open drain disable 1: open drain enable
6	SR	slew rate 0: Slow slew rate 1: Fast slew rate
5-4	SPD	additional 2-bit slew rate to select IO cell operation frequency range with reduced switching noise 00: Slow frequency slew rate(50Mhz) 01: Medium frequency slew rate(100 Mhz) 10: Fast frequency slew rate(150 Mhz) 11: Max frequency slew rate(200Mhz)

Bits	Name	Description
2-0	DS	drive strength 1.8V Mode: 000: 260 Ohm 001: 260 Ohm 010: 130 Ohm 011: 88 Ohm 100: 65 Ohm 101: 52 Ohm 110: 43 Ohm 111: 37 Ohm 3.3V Mode: 000: 157 Ohm 001: 157 Ohm 010: 78 Ohm 011: 53 Ohm 100: 39 Ohm 101: 32 Ohm 110: 26 Ohm 111: 23 Ohm

PAD[PAD_CTL] Bits

22 GPIO Controller

This section describes the main features and functions of the GPIO controller. GPIO controller types include:

- GPIO controller: GPIO0
- fast GPIO controller: FGPI00 (CPU0 fast access)
- Power management domain GPIO controller: PGPI0

22.1 Feature Summary

This chapter describes the main features of the GPIO controller:

- Configure IO as input or output
- Read the status of IO input
- Set the output of IO
- Atomization operation sets IO output high, output low, flip

22.2 Function Description

22.2.1 GPIO Control

The GPIO controller supports input and output direction control, and each IO has a corresponding Direction control bit in the OE register. The user can set this bit to 1 to configure the corresponding IO as a GPIO output; clearing 0 to configure the IO as a GPIO input. When

is used as the output mode, the user can configure the output state of the GPIO through the DO register of the GPIO controller.

GPIO output supports atomization operation register:

- outputs the high register SET. Setting the corresponding bit in this register to 1 will set the corresponding IO output high; setting it to 0 will keep the IO output unchanged.
- outputs the low register CLEAR. Setting the corresponding bit in this register to 1 will set the corresponding IO output low; setting it to 0 will keep the IO output unchanged.
- flips the register TOGGLE. Setting the corresponding bit in this register to 1 will flip the corresponding IO output; setting it to 0 will keep the IO output unchanged.

Users can read the level status of IO through the DI register of the GPIO controller.

Note that the DI register can implement IO snooping. That is, no matter whether the GPIO is configured as an input or an output, or whether the corresponding IO controller IOC maps the IO function to a GPIO (at this time, the output loopback function, i.e. loopback, needs to be turned on through the IOC), the user can always read the IO status from the DI.

22.2.2 GPIO interrupt

GPIO controllers, the following controllers support interrupt generation: GPIO0

_SUPPORT_GPIO1=1 、 GPIO1_SUPPORT_PGPI0=1 、 PGPI0_SUPPORT_BGPI0=1 、 BGPI0

Among the GPIO controllers, the following controllers do not support generation of interrupts: FGPI00

_SUPPORT_FGPI01=1 、 FGPI01

Users can specify the interrupt type through the TP register of the GPIO controller. The corresponding bit 1

means that the interrupt is triggered by an edge, and the corresponding bit is cleared to 0 means that the interrupt is triggered by a level.

users can specify the polarity of the interrupt through the PL register of the GPIO controller. The corresponding bit 1 means that the interrupt is triggered by a falling edge or a low level, and the corresponding bit is cleared to 0 means that the interrupt is triggered by a rising edge or a high level.

GPIO supports the generation of asynchronous interrupts. Users can select synchronous or asynchronous interrupts through the AS register. Asynchronous interrupts allow interrupts to be generated when the system clock is abnormal or when there is no clock.

Through the PD register, you can select whether to enable dual edge-triggered interrupts. The corresponding bit 1 means that rising edge and falling edge interrupts are enabled. The corresponding bit is cleared to 0 means that only single edge-triggered interrupts are allowed. The specific rising edge or falling edge depends on the PL register. Configuration determines.

_SUPPORT_BGPI0=1 It should be noted that BGPI0 does not support double edge-triggered interrupts.

interrupt option is configured, the user can turn on the GPIO interrupt through the IE register of the GPIO controller, and the corresponding bit 1 in the IE register can enable the interrupt corresponding to the IO. When an interrupt event is triggered, the user can check the status of the interrupt through the IF register of the GPIO controller. The corresponding flag position is 1, which means that there is an interrupt to be processed for the corresponding IO. Write 1 to the flag bit to clear this flag bit.

22.3 GPIO register list

GPIO is as follows:

FGPI0 base address: 0x00300000

GPIO0 base address: 0xF00D0000

PGPI0 base address: 0xF411C000

Offset	Register Name	Description	Reset Value
0x0000	DI[GPIOA][VALUE]	GPIO input value	0x00000000
0x0010	DI[GPIOB][VALUE]	GPIO input value	0x00000000
0x0020	DI[GPIOC][VALUE]	GPIO input value	0x00000000
0x0030	DI[GIPIOD][VALUE]	GPIO input value	0x00000000
0x0040	DI[GPIOE][VALUE]	GPIO input value	0x00000000
0x0050	DI[GPIOF][VALUE]	GPIO input value	0x00000000
0x00B0	DI[GPIOV][VALUE]	GPIO input value	0x00000000
0x00C0	DI[GPIOW][VALUE]	GPIO input value	0x00000000
0x00D0	DI[GPIOX][VALUE]	GPIO input value	0x00000000
0x00E0	DI[GPIOY][VALUE]	GPIO input value	0x00000000
0x0100	DO[GPIOA][VALUE]	GPIO output value	0x00000000
0x0104	DO[GPIOA][SET]	GPIO output set	0x00000000
0x0108	DO[GPIOA][CLEAR]	GPIO output clear	0x00000000

Offset	Register Name	Description	Reset Value
0x010C	DO[GPIOA][TOGGLE]	GPIO output toggle	0x00000000
0x0110	DO[GPIOB][VALUE]	GPIO output value	0x00000000
0x0114	DO[GPIOB][SET]	GPIO output set	0x00000000
0x0118	DO[GPIOB][CLEAR]	GPIO output clear	0x00000000
0x011C	DO[GPIOB][TOGGLE]	GPIO output toggle	0x00000000
0x0120	DO[GPIOC][VALUE]	GPIO output value	0x00000000
0x0124	DO[GPIOC][SET]	GPIO output set	0x00000000
0x0128	DO[GPIOC][CLEAR]	GPIO output clear	0x00000000
0x012C	DO[GPIOC][TOGGLE]	GPIO output toggle	0x00000000
0x0130	DO[GPIOD][VALUE]	GPIO output value	0x00000000
0x0134	DO[GPIOD][SET]	GPIO output set	0x00000000
0x0138	DO[GPIOD][CLEAR]	GPIO output clear	0x00000000
0x013C	DO[GPIOD][TOGGLE]	GPIO output toggle	0x00000000
0x0140	DO[GPIOE][VALUE]	GPIO output value	0x00000000
0x0144	DO[GPIOE][SET]	GPIO output set	0x00000000
0x0148	DO[GPIOE][CLEAR]	GPIO output clear	0x00000000
0x014C	DO[GPIOE][TOGGLE]	GPIO output toggle	0x00000000
0x0150	DO[GPIOF][VALUE]	GPIO output value	0x00000000
0x0154	DO[GPIOF][SET]	GPIO output set	0x00000000
0x0158	DO[GPIOF][CLEAR]	GPIO output clear	0x00000000
0x015C	DO[GPIOF][TOGGLE]	GPIO output toggle	0x00000000
0x01B0	DO[GPIOV][VALUE]	GPIO output value	0x00000000
0x01B4	DO[GPIOV][SET]	GPIO output set	0x00000000
0x01B8	DO[GPIOV][CLEAR]	GPIO output clear	0x00000000
0x01BC	DO[GPIOV][TOGGLE]	GPIO output toggle	0x00000000
0x01C0	DO[GPIOW][VALUE]	GPIO output value	0x00000000
0x01C4	DO[GPIOW][SET]	GPIO output set	0x00000000
0x01C8	DO[GPIOW][CLEAR]	GPIO output clear	0x00000000
0x01CC	DO[GPIOW][TOGGLE]	GPIO output toggle	0x00000000
0x01D0	DO[GPIOX][VALUE]	GPIO output value	0x00000000
0x01D4	DO[GPIOX][SET]	GPIO output set	0x00000000
0x01D8	DO[GPIOX][CLEAR]	GPIO output clear	0x00000000
0x01DC	DO[GPIOX][TOGGLE]	GPIO output toggle	0x00000000
0x01E0	DO[GPIOY][VALUE]	GPIO output value	0x00000000
0x01E4	DO[GPIOY][SET]	GPIO output set	0x00000000
0x01E8	DO[GPIOY][CLEAR]	GPIO output clear	0x00000000
0x01EC	DO[GPIOY][TOGGLE]	GPIO output toggle	0x00000000
0x0200	OE[GPIOA][VALUE]	GPIO direction value	0x00000000
0x0204	OE[GPIOA][SET]	GPIO direction set	0x00000000
0x0208	OE[GPIOA][CLEAR]	GPIO direction clear	0x00000000

Offset	Register Name	Description	Reset Value
0x020C	OE[GPIOA][TOGGLE]	GPIO direction toggle	0x00000000
0x0210	OE[GPIOB][VALUE]	GPIO direction value	0x00000000
0x0214	OE[GPIOB][SET]	GPIO direction set	0x00000000
0x0218	OE[GPIOB][CLEAR]	GPIO direction clear	0x00000000
0x021C	OE[GPIOB][TOGGLE]	GPIO direction toggle	0x00000000
0x0220	OE[GPIOC][VALUE]	GPIO direction value	0x00000000
0x0224	OE[GPIOC][SET]	GPIO direction set	0x00000000
0x0228	OE[GPIOC][CLEAR]	GPIO direction clear	0x00000000
0x022C	OE[GPIOC][TOGGLE]	GPIO direction toggle	0x00000000
0x0230	OE[GPIOD][VALUE]	GPIO direction value	0x00000000
0x0234	OE[GPIOD][SET]	GPIO direction set	0x00000000
0x0238	OE[GPIOD][CLEAR]	GPIO direction clear	0x00000000
0x023C	OE[GPIOD][TOGGLE]	GPIO direction toggle	0x00000000
0x0240	OE[GPIOE][VALUE]	GPIO direction value	0x00000000
0x0244	OE[GPIOE][SET]	GPIO direction set	0x00000000
0x0248	OE[GPIOE][CLEAR]	GPIO direction clear	0x00000000
0x024C	OE[GPIOE][TOGGLE]	GPIO direction toggle	0x00000000
0x0250	OE[GPIOF][VALUE]	GPIO direction value	0x00000000
0x0254	OE[GPIOF][SET]	GPIO direction set	0x00000000
0x0258	OE[GPIOF][CLEAR]	GPIO direction clear	0x00000000
0x025C	OE[GPIOF][TOGGLE]	GPIO direction toggle	0x00000000
0x02B0	OE[GPIOV][VALUE]	GPIO direction value	0x00000000
0x02B4	OE[GPIOV][SET]	GPIO direction set	0x00000000
0x02B8	OE[GPIOV][CLEAR]	GPIO direction clear	0x00000000
0x02BC	OE[GPIOV][TOGGLE]	GPIO direction toggle	0x00000000
0x02C0	OE[GPIOW][VALUE]	GPIO direction value	0x00000000
0x02C4	OE[GPIOW][SET]	GPIO direction set	0x00000000
0x02C8	OE[GPIOW][CLEAR]	GPIO direction clear	0x00000000
0x02CC	OE[GPIOW][TOGGLE]	GPIO direction toggle	0x00000000
0x02D0	OE[GPIOX][VALUE]	GPIO direction value	0x00000000
0x02D4	OE[GPIOX][SET]	GPIO direction set	0x00000000
0x02D8	OE[GPIOX][CLEAR]	GPIO direction clear	0x00000000
0x02DC	OE[GPIOX][TOGGLE]	GPIO direction toggle	0x00000000
0x02E0	OE[GPIOY][VALUE]	GPIO direction value	0x00000000
0x02E4	OE[GPIOY][SET]	GPIO direction set	0x00000000
0x02E8	OE[GPIOY][CLEAR]	GPIO direction clear	0x00000000
0x02EC	OE[GPIOY][TOGGLE]	GPIO direction toggle	0x00000000
0x0300	IF[GPIOA][VALUE]	GPIO interrupt flag value	0x00000000
0x0310	IF[GPIOB][VALUE]	GPIO interrupt flag value	0x00000000
0x0320	IF[GPIOC][VALUE]	GPIO interrupt flag value	0x00000000

Offset	Register Name	Description	Reset Value
0x0330	IF[GPIOD][VALUE]	GPIO interrupt flag value	0x00000000
0x0340	IF[GPIOE][VALUE]	GPIO interrupt flag value	0x00000000
0x0350	IF[GPIOF][VALUE]	GPIO interrupt flag value	0x00000000
0x03B0	IF[GPIOV][VALUE]	GPIO interrupt flag value	0x00000000
0x03C0	IF[GPIOW][VALUE]	GPIO interrupt flag value	0x00000000
0x03D0	IF[GPIOX][VALUE]	GPIO interrupt flag value	0x00000000
0x03E0	IF[GPIOY][VALUE]	GPIO interrupt flag value	0x00000000
0x0400	IE[GPIOA][VALUE]	GPIO interrupt enable value	0x00000000
0x0404	IE[GPIOA][SET]	GPIO interrupt enable set	0x00000000
0x0408	IE[GPIOA][CLEAR]	GPIO interrupt enable clear	0x00000000
0x040C	IE[GPIOA][TOGGLE]	GPIO interrupt enable toggle	0x00000000
0x0410	IE[GPIOB][VALUE]	GPIO interrupt enable value	0x00000000
0x0414	IE[GPIOB][SET]	GPIO interrupt enable set	0x00000000
0x0418	IE[GPIOB][CLEAR]	GPIO interrupt enable clear	0x00000000
0x041C	IE[GPIOB][TOGGLE]	GPIO interrupt enable toggle	0x00000000
0x0420	IE[GPIOC][VALUE]	GPIO interrupt enable value	0x00000000
0x0424	IE[GPIOC][SET]	GPIO interrupt enable set	0x00000000
0x0428	IE[GPIOC][CLEAR]	GPIO interrupt enable clear	0x00000000
0x042C	IE[GPIOC][TOGGLE]	GPIO interrupt enable toggle	0x00000000
0x0430	IE[GPIOD][VALUE]	GPIO interrupt enable value	0x00000000
0x0434	IE[GPIOD][SET]	GPIO interrupt enable set	0x00000000
0x0438	IE[GPIOD][CLEAR]	GPIO interrupt enable clear	0x00000000
0x043C	IE[GPIOD][TOGGLE]	GPIO interrupt enable toggle	0x00000000
0x0440	IE[GPIOE][VALUE]	GPIO interrupt enable value	0x00000000
0x0444	IE[GPIOE][SET]	GPIO interrupt enable set	0x00000000
0x0448	IE[GPIOE][CLEAR]	GPIO interrupt enable clear	0x00000000
0x044C	IE[GPIOE][TOGGLE]	GPIO interrupt enable toggle	0x00000000
0x0450	IE[GPIOF][VALUE]	GPIO interrupt enable value	0x00000000
0x0454	IE[GPIOF][SET]	GPIO interrupt enable set	0x00000000
0x0458	IE[GPIOF][CLEAR]	GPIO interrupt enable clear	0x00000000
0x045C	IE[GPIOF][TOGGLE]	GPIO interrupt enable toggle	0x00000000
0x04B0	IE[GPIOV][VALUE]	GPIO interrupt enable value	0x00000000
0x04B4	IE[GPIOV][SET]	GPIO interrupt enable set	0x00000000
0x04B8	IE[GPIOV][CLEAR]	GPIO interrupt enable clear	0x00000000
0x04BC	IE[GPIOV][TOGGLE]	GPIO interrupt enable toggle	0x00000000
0x04C0	IE[GPIOW][VALUE]	GPIO interrupt enable value	0x00000000
0x04C4	IE[GPIOW][SET]	GPIO interrupt enable set	0x00000000
0x04C8	IE[GPIOW][CLEAR]	GPIO interrupt enable clear	0x00000000
0x04CC	IE[GPIOW][TOGGLE]	GPIO interrupt enable toggle	0x00000000
0x04D0	IE[GPIOX][VALUE]	GPIO interrupt enable value	0x00000000

Offset	Register Name	Description	Reset Value
0x04D4	IE[GPIOX][SET]	GPIO interrupt enable set	0x00000000
0x04D8	IE[GPIOX][CLEAR]	GPIO interrupt enable clear	0x00000000
0x04DC	IE[GPIOX][TOGGLE]	GPIO interrupt enable toggle	0x00000000
0x04E0	IE[GPIOY][VALUE]	GPIO interrupt enable value	0x00000000
0x04E4	IE[GPIOY][SET]	GPIO interrupt enable set	0x00000000
0x04E8	IE[GPIOY][CLEAR]	GPIO interrupt enable clear	0x00000000
0x04EC	IE[GPIOY][TOGGLE]	GPIO interrupt enable toggle	0x00000000
0x0500	PL[GPIOA][VALUE]	GPIO interrupt polarity value	0x00000000
0x0504	PL[GPIOA][SET]	GPIO interrupt polarity set	0x00000000
0x0508	PL[GPIOA][CLEAR]	GPIO interrupt polarity clear	0x00000000
0x050C	PL[GPIOA][TOGGLE]	GPIO interrupt polarity toggle	0x00000000
0x0510	PL[GPIOB][VALUE]	GPIO interrupt polarity value	0x00000000
0x0514	PL[GPIOB][SET]	GPIO interrupt polarity set	0x00000000
0x0518	PL[GPIOB][CLEAR]	GPIO interrupt polarity clear	0x00000000
0x051C	PL[GPIOB][TOGGLE]	GPIO interrupt polarity toggle	0x00000000
0x0520	PL[GPIOC][VALUE]	GPIO interrupt polarity value	0x00000000
0x0524	PL[GPIOC][SET]	GPIO interrupt polarity set	0x00000000
0x0528	PL[GPIOC][CLEAR]	GPIO interrupt polarity clear	0x00000000
0x052C	PL[GPIOC][TOGGLE]	GPIO interrupt polarity toggle	0x00000000
0x0530	PL[GPIOD][VALUE]	GPIO interrupt polarity value	0x00000000
0x0534	PL[GPIOD][SET]	GPIO interrupt polarity set	0x00000000
0x0538	PL[GPIOD][CLEAR]	GPIO interrupt polarity clear	0x00000000
0x053C	PL[GPIOD][TOGGLE]	GPIO interrupt polarity toggle	0x00000000
0x0540	PL[GPIOE][VALUE]	GPIO interrupt polarity value	0x00000000
0x0544	PL[GPIOE][SET]	GPIO interrupt polarity set	0x00000000
0x0548	PL[GPIOE][CLEAR]	GPIO interrupt polarity clear	0x00000000
0x054C	PL[GPIOE][TOGGLE]	GPIO interrupt polarity toggle	0x00000000
0x0550	PL[GPIOF][VALUE]	GPIO interrupt polarity value	0x00000000
0x0554	PL[GPIOF][SET]	GPIO interrupt polarity set	0x00000000
0x0558	PL[GPIOF][CLEAR]	GPIO interrupt polarity clear	0x00000000
0x055C	PL[GPIOF][TOGGLE]	GPIO interrupt polarity toggle	0x00000000
0x05B0	PL[GPIOV][VALUE]	GPIO interrupt polarity value	0x00000000
0x05B4	PL[GPIOV][SET]	GPIO interrupt polarity set	0x00000000
0x05B8	PL[GPIOV][CLEAR]	GPIO interrupt polarity clear	0x00000000
0x05BC	PL[GPIOV][TOGGLE]	GPIO interrupt polarity toggle	0x00000000
0x05C0	PL[GPIOW][VALUE]	GPIO interrupt polarity value	0x00000000
0x05C4	PL[GPIOW][SET]	GPIO interrupt polarity set	0x00000000
0x05C8	PL[GPIOW][CLEAR]	GPIO interrupt polarity clear	0x00000000
0x05CC	PL[GPIOW][TOGGLE]	GPIO interrupt polarity toggle	0x00000000
0x05D0	PL[GPIOX][VALUE]	GPIO interrupt polarity value	0x00000000

Offset	Register Name	Description	Reset Value
0x05D4	PL[GPIOX][SET]	GPIO interrupt polarity set	0x00000000
0x05D8	PL[GPIOX][CLEAR]	GPIO interrupt polarity clear	0x00000000
0x05DC	PL[GPIOX][TOGGLE]	GPIO interrupt polarity toggle	0x00000000
0x05E0	PL[GPIOW][VALUE]	GPIO interrupt polarity value	0x00000000
0x05E4	PL[GPIOW][SET]	GPIO interrupt polarity set	0x00000000
0x05E8	PL[GPIOW][CLEAR]	GPIO interrupt polarity clear	0x00000000
0x05EC	PL[GPIOW][TOGGLE]	GPIO interrupt polarity toggle	0x00000000
0x0600	TP[GPIOA][VALUE]	GPIO interrupt type value	0x00000000
0x0604	TP[GPIOA][SET]	GPIO interrupt type set	0x00000000
0x0608	TP[GPIOA][CLEAR]	GPIO interrupt type clear	0x00000000
0x060C	TP[GPIOA][TOGGLE]	GPIO interrupt type toggle	0x00000000
0x0610	TP[GPIOB][VALUE]	GPIO interrupt type value	0x00000000
0x0614	TP[GPIOB][SET]	GPIO interrupt type set	0x00000000
0x0618	TP[GPIOB][CLEAR]	GPIO interrupt type clear	0x00000000
0x061C	TP[GPIOB][TOGGLE]	GPIO interrupt type toggle	0x00000000
0x0620	TP[GPIOC][VALUE]	GPIO interrupt type value	0x00000000
0x0624	TP[GPIOC][SET]	GPIO interrupt type set	0x00000000
0x0628	TP[GPIOC][CLEAR]	GPIO interrupt type clear	0x00000000
0x062C	TP[GPIOC][TOGGLE]	GPIO interrupt type toggle	0x00000000
0x0630	TP[GPIOD][VALUE]	GPIO interrupt type value	0x00000000
0x0634	TP[GPIOD][SET]	GPIO interrupt type set	0x00000000
0x0638	TP[GPIOD][CLEAR]	GPIO interrupt type clear	0x00000000
0x063C	TP[GPIOD][TOGGLE]	GPIO interrupt type toggle	0x00000000
0x0640	TP[GPIOE][VALUE]	GPIO interrupt type value	0x00000000
0x0644	TP[GPIOE][SET]	GPIO interrupt type set	0x00000000
0x0648	TP[GPIOE][CLEAR]	GPIO interrupt type clear	0x00000000
0x064C	TP[GPIOE][TOGGLE]	GPIO interrupt type toggle	0x00000000
0x0650	TP[GPIOF][VALUE]	GPIO interrupt type value	0x00000000
0x0654	TP[GPIOF][SET]	GPIO interrupt type set	0x00000000
0x0658	TP[GPIOF][CLEAR]	GPIO interrupt type clear	0x00000000
0x065C	TP[GPIOF][TOGGLE]	GPIO interrupt type toggle	0x00000000
0x06B0	TP[GPIOV][VALUE]	GPIO interrupt type value	0x00000000
0x06B4	TP[GPIOV][SET]	GPIO interrupt type set	0x00000000
0x06B8	TP[GPIOV][CLEAR]	GPIO interrupt type clear	0x00000000
0x06BC	TP[GPIOV][TOGGLE]	GPIO interrupt type toggle	0x00000000
0x06C0	TP[GPIOW][VALUE]	GPIO interrupt type value	0x00000000
0x06C4	TP[GPIOW][SET]	GPIO interrupt type set	0x00000000
0x06C8	TP[GPIOW][CLEAR]	GPIO interrupt type clear	0x00000000
0x06CC	TP[GPIOW][TOGGLE]	GPIO interrupt type toggle	0x00000000
0x06D0	TP[GPIOX][VALUE]	GPIO interrupt type value	0x00000000

Offset	Register Name	Description	Reset Value
0x06D4	TP[GPIOX][SET]	GPIO interrupt type set	0x00000000
0x06D8	TP[GPIOX][CLEAR]	GPIO interrupt type clear	0x00000000
0x06DC	TP[GPIOX][TOGGLE]	GPIO interrupt type toggle	0x00000000
0x06E0	TP[GPIOY][VALUE]	GPIO interrupt type value	0x00000000
0x06E4	TP[GPIOY][SET]	GPIO interrupt type set	0x00000000
0x06E8	TP[GPIOY][CLEAR]	GPIO interrupt type clear	0x00000000
0x06EC	TP[GPIOY][TOGGLE]	GPIO interrupt type toggle	0x00000000
0x0700	AS[GPIOA][VALUE]	GPIO interrupt asynchronous value	0x00000000
0x0704	AS[GPIOA][SET]	GPIO interrupt asynchronous set	0x00000000
0x0708	AS[GPIOA][CLEAR]	GPIO interrupt asynchronous clear	0x00000000
0x070C	AS[GPIOA][TOGGLE]	GPIO interrupt asynchronous toggle	0x00000000
0x0710	AS[GPIOB][VALUE]	GPIO interrupt asynchronous value	0x00000000
0x0714	AS[GPIOB][SET]	GPIO interrupt asynchronous set	0x00000000
0x0718	AS[GPIOB][CLEAR]	GPIO interrupt asynchronous clear	0x00000000
0x071C	AS[GPIOB][TOGGLE]	GPIO interrupt asynchronous toggle	0x00000000
0x0720	AS[GPIOC][VALUE]	GPIO interrupt asynchronous value	0x00000000
0x0724	AS[GPIOC][SET]	GPIO interrupt asynchronous set	0x00000000
0x0728	AS[GPIOC][CLEAR]	GPIO interrupt asynchronous clear	0x00000000
0x072C	AS[GPIOC][TOGGLE]	GPIO interrupt asynchronous toggle	0x00000000
0x0730	AS[GPIOD][VALUE]	GPIO interrupt asynchronous value	0x00000000
0x0734	AS[GPIOD][SET]	GPIO interrupt asynchronous set	0x00000000
0x0738	AS[GPIOD][CLEAR]	GPIO interrupt asynchronous clear	0x00000000
0x073C	AS[GPIOD][TOGGLE]	GPIO interrupt asynchronous toggle	0x00000000
0x0740	AS[GPIOE][VALUE]	GPIO interrupt asynchronous value	0x00000000

Offset	Register Name	Description	Reset Value
0x0744	AS[GPIOE][SET]	GPIO interrupt asynchronous set	0x00000000
0x0748	AS[GPIOE][CLEAR]	GPIO interrupt asynchronous clear	0x00000000
0x074C	AS[GPIOE][TOGGLE]	GPIO interrupt asynchronous toggle	0x00000000
0x0750	AS[GPIOF][VALUE]	GPIO interrupt asynchronous value	0x00000000
0x0754	AS[GPIOF][SET]	GPIO interrupt asynchronous set	0x00000000
0x0758	AS[GPIOF][CLEAR]	GPIO interrupt asynchronous clear	0x00000000
0x075C	AS[GPIOF][TOGGLE]	GPIO interrupt asynchronous toggle	0x00000000
0x07B0	AS[GPIOV][VALUE]	GPIO interrupt asynchronous value	0x00000000
0x07B4	AS[GPIOV][SET]	GPIO interrupt asynchronous set	0x00000000
0x07B8	AS[GPIOV][CLEAR]	GPIO interrupt asynchronous clear	0x00000000
0x07BC	AS[GPIOV][TOGGLE]	GPIO interrupt asynchronous toggle	0x00000000
0x07C0	AS[GPIOW][VALUE]	GPIO interrupt asynchronous value	0x00000000
0x07C4	AS[GPIOW][SET]	GPIO interrupt asynchronous set	0x00000000
0x07C8	AS[GPIOW][CLEAR]	GPIO interrupt asynchronous clear	0x00000000
0x07CC	AS[GPIOW][TOGGLE]	GPIO interrupt asynchronous toggle	0x00000000
0x07D0	AS[GPIOX][VALUE]	GPIO interrupt asynchronous value	0x00000000
0x07D4	AS[GPIOX][SET]	GPIO interrupt asynchronous set	0x00000000
0x07D8	AS[GPIOX][CLEAR]	GPIO interrupt asynchronous clear	0x00000000
0x07DC	AS[GPIOX][TOGGLE]	GPIO interrupt asynchronous toggle	0x00000000
0x07E0	AS[GPIOY][VALUE]	GPIO interrupt asynchronous value	0x00000000

Offset	Register Name	Description	Reset Value
0x07E4	AS[GPIOW][SET]	GPIO interrupt asynchronous set	0x00000000
0x07E8	AS[GPIOW][CLEAR]	GPIO interrupt asynchronous clear	0x00000000
0x07EC	AS[GPIOW][TOGGLE]	GPIO interrupt asynchronous toggle	0x00000000
0x0800	PD[GPIOA][VALUE]	GPIO dual edge interrupt enable value	0x00000000
0x0804	PD[GPIOA][SET]	GPIO dual edge interrupt enable set	0x00000000
0x0808	PD[GPIOA][CLEAR]	GPIO dual edge interrupt enable clear	0x00000000
0x080C	PD[GPIOA][TOGGLE]	GPIO dual edge interrupt enable toggle	0x00000000
0x0810	PD[GPIOB][VALUE]	GPIO dual edge interrupt enable value	0x00000000
0x0814	PD[GPIOB][SET]	GPIO dual edge interrupt enable set	0x00000000
0x0818	PD[GPIOB][CLEAR]	GPIO dual edge interrupt enable clear	0x00000000
0x081C	PD[GPIOB][TOGGLE]	GPIO dual edge interrupt enable toggle	0x00000000
0x0820	PD[GPIOC][VALUE]	GPIO dual edge interrupt enable value	0x00000000
0x0824	PD[GPIOC][SET]	GPIO dual edge interrupt enable set	0x00000000
0x0828	PD[GPIOC][CLEAR]	GPIO dual edge interrupt enable clear	0x00000000
0x082C	PD[GPIOC][TOGGLE]	GPIO dual edge interrupt enable toggle	0x00000000
0x0830	PD[GPIOD][VALUE]	GPIO dual edge interrupt enable value	0x00000000
0x0834	PD[GPIOD][SET]	GPIO dual edge interrupt enable set	0x00000000
0x0838	PD[GPIOD][CLEAR]	GPIO dual edge interrupt enable clear	0x00000000
0x083C	PD[GPIOD][TOGGLE]	GPIO dual edge interrupt enable toggle	0x00000000
0x0840	PD[GPIOE][VALUE]	GPIO dual edge interrupt enable value	0x00000000

Offset	Register Name	Description	Reset Value
0x0844	PD[GPIOE][SET]	GPIO dual edge interrupt enable set	0x00000000
0x0848	PD[GPIOE][CLEAR]	GPIO dual edge interrupt enable clear	0x00000000
0x084C	PD[GPIOE][TOGGLE]	GPIO dual edge interrupt enable toggle	0x00000000
0x0850	PD[GPIOF][VALUE]	GPIO dual edge interrupt enable value	0x00000000
0x0854	PD[GPIOF][SET]	GPIO dual edge interrupt enable set	0x00000000
0x0858	PD[GPIOF][CLEAR]	GPIO dual edge interrupt enable clear	0x00000000
0x085C	PD[GPIOF][TOGGLE]	GPIO dual edge interrupt enable toggle	0x00000000
0x08B0	PD[GPIOV][VALUE]	GPIO dual edge interrupt enable value	0x00000000
0x08B4	PD[GPIOV][SET]	GPIO dual edge interrupt enable set	0x00000000
0x08B8	PD[GPIOV][CLEAR]	GPIO dual edge interrupt enable clear	0x00000000
0x08BC	PD[GPIOV][TOGGLE]	GPIO dual edge interrupt enable toggle	0x00000000
0x08C0	PD[GPIOW][VALUE]	GPIO dual edge interrupt enable value	0x00000000
0x08C4	PD[GPIOW][SET]	GPIO dual edge interrupt enable set	0x00000000
0x08C8	PD[GPIOW][CLEAR]	GPIO dual edge interrupt enable clear	0x00000000
0x08CC	PD[GPIOW][TOGGLE]	GPIO dual edge interrupt enable toggle	0x00000000
0x08D0	PD[GPIOX][VALUE]	GPIO dual edge interrupt enable value	0x00000000
0x08D4	PD[GPIOX][SET]	GPIO dual edge interrupt enable set	0x00000000
0x08D8	PD[GPIOX][CLEAR]	GPIO dual edge interrupt enable clear	0x00000000
0x08DC	PD[GPIOX][TOGGLE]	GPIO dual edge interrupt enable toggle	0x00000000
0x08E0	PD[GPIOY][VALUE]	GPIO dual edge interrupt enable value	0x00000000

Offset	Register Name	Description	Reset Value
0x08E4	PD[GPIOY][SET]	GPIO dual edge interrupt enable set	0x00000000
0x08E8	PD[GPIOY][CLEAR]	GPIO dual edge interrupt enable clear	0x00000000
0x08EC	PD[GPIOY][TOGGLE]	GPIO dual edge interrupt enable toggle	0x00000000

Table 148: GPIO Register List

22.4 GPIO register description

GPIO is as follows:

22.4.1 DI[VALUE] (0x0 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INPUT																															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DI[VALUE] [31:0]

Bits	Name	Description
31-0	INPUT	GPIO input bus value, each bit represents a bus bit 0: low level presents on chip pin 1: high level presents on chip pin

DI[VALUE] Bits

22.4.2 DO[VALUE] (0x100 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OUTPUT																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DO[VALUE] [31:0]

Bits	Name	Description
31-0	OUTPUT	GPIO output register value, each bit represents a bus bit 0: chip pin output low level when direction is output 1: chip pin output high level when direction is output

DO[VALUE] Bits

22.4.3 DO[SET] (0x104 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																OUTPUT															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DO[SET] [31:0]

Bits	Name	Description
31-0	OUTPUT	GPIO output register value, each bit represents a bus bit 0: chip pin output low level when direction is output 1: chip pin output high level when direction is output

DO[SET] Bits

22.4.4 DO[CLEAR] (0x108 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																OUTPUT															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DO[CLEAR] [31:0]

Bits	Name	Description
31-0	OUTPUT	GPIO output register value, each bit represents a bus bit 0: chip pin output low level when direction is output 1: chip pin output high level when direction is output

DO[CLEAR] Bits

22.4.5 DO[TOGGLE] (0x10C + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OUTPUT																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DO[TOGGLE] [31:0]

Bits	Name	Description
31-0	OUTPUT	GPIO output register value, each bit represents a bus bit 0: chip pin output low level when direction is output 1: chip pin output high level when direction is output

DO[TOGGLE] Bits

22.4.6 OE[VALUE] (0x200 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIRECTION																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OE[VALUE] [31:0]

Bits	Name	Description
31-0	DIRECTION	GPIO direction, each bit represents a bus bit 0: input 1: output

OE[VALUE] Bits

22.4.7 OE[SET] (0x204 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIRECTION																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OE[SET] [31:0]

Bits	Name	Description
31-0	DIRECTION	GPIO direction, each bit represents a bus bit 0: input 1: output

OE[SET] Bits

22.4.8 OE[CLEAR] (0x208 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIRECTION																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OE[CLEAR] [31:0]

Bits	Name	Description
31-0	DIRECTION	GPIO direction, each bit represents a bus bit 0: input 1: output

OE[CLEAR] Bits

22.4.9 OE[TOGGLE] (0x20C + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIRECTION																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OE[TOGGLE] [31:0]

Bits	Name	Description
31-0	DIRECTION	GPIO direction, each bit represents a bus bit 0: input 1: output

OE[TOGGLE] Bits

22.4.10 IF[VALUE] (0x300 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRQ_FLAG																W1C															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IF[VALUE] [31:0]

Bits	Name	Description
31-0	IRQ_FLAG	GPIO interrupt flag, write 1 to clear this flag 0: no irq 1: irq pending

IF[VALUE] Bits

22.4.11 IE[VALUE] (0x400 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRQ_EN																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IE[VALUE] [31:0]

Bits	Name	Description
31-0	IRQ_EN	GPIO interrupt enable, each bit represents a bus bit 0: irq is disabled 1: irq is enable

IE[VALUE] Bits

22.4.12 IE[SET] (0x404 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRQ_EN																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IE[SET] [31:0]

Bits	Name	Description
31-0	IRQ_EN	GPIO interrupt enable, each bit represents a bus bit 0: irq is disabled 1: irq is enable

IE[SET] Bits

22.4.13 IE[CLEAR] (0x408 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																IRQ_EN															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IE[CLEAR] [31:0]

Bits	Name	Description
31-0	IRQ_EN	GPIO interrupt enable, each bit represents a bus bit 0: irq is disabled 1: irq is enable

IE[CLEAR] Bits

22.4.14 IE[TOGGLE] (0x40C + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																	
																IRQ_EN																																																
																																RW																																
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																																	

IE[TOGGLE] [31:0]

Bits	Name	Description
31-0	IRQ_EN	GPIO interrupt enable, each bit represents a bus bit 0: irq is disabled 1: irq is enable

IE[TOGGLE] Bits

22.4.15 PL[VALUE] (0x500 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRQ_POL																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PL[VALUE] [31:0]

Bits	Name	Description
31-0	IRQ_POL	GPIO interrupt polarity, each bit represents a bus bit 0: irq is high level or rising edge 1: irq is low level or falling edge

PL[VALUE] Bits

22.4.16 PL[SET] (0x504 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRQ_POL																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PL[SET] [31:0]

Bits	Name	Description
31-0	IRQ_POL	GPIO interrupt polarity, each bit represents a bus bit 0: irq is high level or rising edge 1: irq is low level or falling edge

PL[SET] Bits

22.4.17 PL[CLEAR] (0x508 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRQ_POL																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PL[CLEAR] [31:0]

Bits	Name	Description
31-0	IRQ_POL	GPIO interrupt polarity, each bit represents a bus bit 0: irq is high level or rising edge 1: irq is low level or falling edge

PL[CLEAR] Bits

22.4.18 PL[TOGGLE] (0x50C + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRQ_POL																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PL[TOGGLE] [31:0]

Bits	Name	Description
31-0	IRQ_POL	GPIO interrupt polarity, each bit represents a bus bit 0: irq is high level or rising edge 1: irq is low level or falling edge

PL[TOGGLE] Bits

22.4.19 TP[VALUE] (0x600 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRQ_TYPE																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TP[VALUE] [31:0]

Bits	Name	Description
31-0	IRQ_TYPE	GPIO interrupt type, each bit represents a bus bit 0: irq is triggered by level 1: irq is triggered by edge

TP[VALUE] Bits

22.4.20 TP[SET] (0x604 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRQ_TYPE																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TP[SET] [31:0]

Bits	Name	Description
31-0	IRQ_TYPE	GPIO interrupt type, each bit represents a bus bit 0: irq is triggered by level 1: irq is triggered by edge

TP[SET] Bits

22.4.21 TP[CLEAR] (0x608 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRQ_TYPE																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TP[CLEAR] [31:0]

Bits	Name	Description
31-0	IRQ_TYPE	GPIO interrupt type, each bit represents a bus bit 0: irq is triggered by level 1: irq is triggered by edge

TP[CLEAR] Bits

22.4.22 TP[TOGGLE] (0x60C + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRQ_TYPE																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TP[TOGGLE] [31:0]

Bits	Name	Description
31-0	IRQ_TYPE	GPIO interrupt type, each bit represents a bus bit 0: irq is triggered by level 1: irq is triggered by edge

TP[TOGGLE] Bits

22.4.23 AS[VALUE] (0x700 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																IRQ_ASYNC															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

AS[VALUE] [31:0]

Bits	Name	Description
31-0	IRQ_ASYNC	GPIO interrupt asynchronous, each bit represents a bus bit 0: irq is triggered base on system clock 1: irq is triggered combinational Note: combinational interrupt is sensitive to environment noise

AS[VALUE] Bits

22.4.24 AS[SET] (0x704 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																IRQ_ASYNC																
RW																																
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

AS[SET] [31:0]

Bits	Name	Description
31-0	IRQ_ASYNC	GPIO interrupt asynchronous, each bit represents a bus bit 0: irq is triggered base on system clock 1: irq is triggered combinational Note: combinational interrupt is sensitive to environment noise

AS[SET] Bits

22.4.25 AS[CLEAR] (0x708 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRQ_ASYNC																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

AS[CLEAR] [31:0]

Bits	Name	Description
31-0	IRQ_ASYNC	GPIO interrupt asynchronous, each bit represents a bus bit 0: irq is triggered base on system clock 1: irq is triggered combinational Note: combinational interrupt is sensitive to environment noise

AS[CLEAR] Bits

22.4.26 AS[TOGGLE] (0x70C + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRQ_ASYNC																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

AS[TOGGLE] [31:0]

Bits	Name	Description
31-0	IRQ_ASYNC	GPIO interrupt asynchronous, each bit represents a bus bit 0: irq is triggered base on system clock 1: irq is triggered combinational Note: combinational interrupt is sensitive to environment noise

AS[TOGGLE] Bits

22.4.27 PD[VALUE] (0x800 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																IRQ_DUAL															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
N/A																															RW
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0

PD[VALUE] [31:0]

Bits	Name	Description
0	IRQ_DUAL	GPIO dual edge interrupt enable 0: single edge interrupt 1: dual edge interrupt enable

PD[VALUE] Bits

22.4.28 PD[SET] (0x804 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																															IRQ_DUAL
N/A																															RW
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0

PD[SET] [31:0]

Bits	Name	Description
0	IRQ_DUAL	GPIO dual edge interrupt enable set 0: keep original edge interrupt type 1: dual edge interrupt enable

PD[SET] Bits

22.4.29 PD[CLEAR] (0x808 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																															IRQ_DUAL
N/A																															RW
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0

PD[CLEAR] [31:0]

Bits	Name	Description
0	IRQ_DUAL	GPIO dual edge interrupt enable clear 0: keep original edge interrupt type 1: single edge interrupt enable

PD[CLEAR] Bits

22.4.30 PD[TOGGLE] (0x80C + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																IRQ_DUAL															
N/A																RW															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0

PD[TOGGLE] [31:0]

Bits	Name	Description
0	IRQ_DUAL	GPIO dual edge interrupt enable toggle 0: keep original edge interrupt type 1: change original edge interrupt type to another one.

PD[TOGGLE] Bits

23 GPIO Manager GPIOM

This chapter describes the main features and functions of the GPIO manager GPIOM. The main function of the GPIO manager is to specify the module for which the GPIO configuration is effective for any IO, and each IO can be individually designated as its controller from the GPIO controller (GPIO0) and the fast GPIO controller (FGPIO0).

23.1 Feature Summary

This chapter introduces the main features of the GPIO manager GPIOM:

- Allocate IO to the specified GPIO controller
- Configure whether IO inputs are visible to specific GPIO controllers
- GPIOM Register Access Control
- GPIOM register lock

23.2 Function Description

This chapter describes the functions of the GPIO manager GPIOM.

23.2.1 GPIO allocation

This product supports multiple GPIO controllers. Its purpose is to facilitate multiple cores and achieve GPIO resource isolation between multiple tasks. The main function of the GPIO manager is to manage all IOs. Each IO can be individually designated to be controlled by a certain GPIO controller. Other GPIO controllers cannot control GPIOs.

Users can manage the y pin of Px through the GPIOxASSIGNy register. The SELECT bit field of this register can select which GPIO controller control this IO receives:

- 0'b0, GPIO0
- 0'b1, FGPIO0

The HIDE bit field of the GPIOxASSIGNy register consists of 2 HIDE bits. Each bit is 1, which means that the input of this IO is not visible in the corresponding GPIO controller DI register, that is, the GPIO controller cannot read the input of this IO.

- HIDE[0], set to 1 means that IO is not visible to GPIO0
- HIDE[1], set to 1 means that IO is not visible to FGPIO0

23.2.2 Access Control

The LOCK bit of the GPIOxASSIGNy register can be used to lock the corresponding register for this IO. Once set to 1, the configuration of this register cannot be changed until the next reset.

The NON_SEC bit in the GPIOxASSIGNy register can control access rights according to the system security status. Once set to 1, the register can only be accessed in a safe state. Please visit the relevant chapter on System Security for detailed information on the security status of the system.

23.3 GPIOM register list

GPIOM is as follows:

GPIOM base address: 0xF00D8000

Offset	Register Name	Description	Reset Value
0x0000	ASSIGN[GPIOA][PIN][PIN00]	GPIO mananger	0x00000000
0x0004	ASSIGN[GPIOA][PIN][PIN01]	GPIO mananger	0x00000000
0x0008	ASSIGN[GPIOA][PIN][PIN02]	GPIO mananger	0x00000000
0x000C	ASSIGN[GPIOA][PIN][PIN03]	GPIO mananger	0x00000000
0x0010	ASSIGN[GPIOA][PIN][PIN04]	GPIO mananger	0x00000000
0x0014	ASSIGN[GPIOA][PIN][PIN05]	GPIO mananger	0x00000000
0x0018	ASSIGN[GPIOA][PIN][PIN06]	GPIO mananger	0x00000000
0x001C	ASSIGN[GPIOA][PIN][PIN07]	GPIO mananger	0x00000000
0x0020	ASSIGN[GPIOA][PIN][PIN08]	GPIO mananger	0x00000000
0x0024	ASSIGN[GPIOA][PIN][PIN09]	GPIO mananger	0x00000000
0x0028	ASSIGN[GPIOA][PIN][PIN10]	GPIO mananger	0x00000000
0x002C	ASSIGN[GPIOA][PIN][PIN11]	GPIO mananger	0x00000000
0x0030	ASSIGN[GPIOA][PIN][PIN12]	GPIO mananger	0x00000000
0x0034	ASSIGN[GPIOA][PIN][PIN13]	GPIO mananger	0x00000000
0x0038	ASSIGN[GPIOA][PIN][PIN14]	GPIO mananger	0x00000000
0x003C	ASSIGN[GPIOA][PIN][PIN15]	GPIO mananger	0x00000000
0x0040	ASSIGN[GPIOA][PIN][PIN16]	GPIO mananger	0x00000000
0x0044	ASSIGN[GPIOA][PIN][PIN17]	GPIO mananger	0x00000000
0x0048	ASSIGN[GPIOA][PIN][PIN18]	GPIO mananger	0x00000000
0x004C	ASSIGN[GPIOA][PIN][PIN19]	GPIO mananger	0x00000000
0x0050	ASSIGN[GPIOA][PIN][PIN20]	GPIO mananger	0x00000000
0x0054	ASSIGN[GPIOA][PIN][PIN21]	GPIO mananger	0x00000000
0x0058	ASSIGN[GPIOA][PIN][PIN22]	GPIO mananger	0x00000000
0x005C	ASSIGN[GPIOA][PIN][PIN23]	GPIO mananger	0x00000000
0x0060	ASSIGN[GPIOA][PIN][PIN24]	GPIO mananger	0x00000000
0x0064	ASSIGN[GPIOA][PIN][PIN25]	GPIO mananger	0x00000000
0x0068	ASSIGN[GPIOA][PIN][PIN26]	GPIO mananger	0x00000000
0x006C	ASSIGN[GPIOA][PIN][PIN27]	GPIO mananger	0x00000000
0x0070	ASSIGN[GPIOA][PIN][PIN28]	GPIO mananger	0x00000000
0x0074	ASSIGN[GPIOA][PIN][PIN29]	GPIO mananger	0x00000000
0x0078	ASSIGN[GPIOA][PIN][PIN30]	GPIO mananger	0x00000000
0x007C	ASSIGN[GPIOA][PIN][PIN31]	GPIO mananger	0x00000000
0x0080	ASSIGN[GPIOB][PIN][PIN00]	GPIO mananger	0x00000000
0x0084	ASSIGN[GPIOB][PIN][PIN01]	GPIO mananger	0x00000000
0x0088	ASSIGN[GPIOB][PIN][PIN02]	GPIO mananger	0x00000000
0x008C	ASSIGN[GPIOB][PIN][PIN03]	GPIO mananger	0x00000000

Offset	Register Name	Description	Reset Value
0x0090	ASSIGN[GPIOB][PIN][PIN04]	GPIO mananger	0x00000000
0x0094	ASSIGN[GPIOB][PIN][PIN05]	GPIO mananger	0x00000000
0x0098	ASSIGN[GPIOB][PIN][PIN06]	GPIO mananger	0x00000000
0x009C	ASSIGN[GPIOB][PIN][PIN07]	GPIO mananger	0x00000000
0x00A0	ASSIGN[GPIOB][PIN][PIN08]	GPIO mananger	0x00000000
0x00A4	ASSIGN[GPIOB][PIN][PIN09]	GPIO mananger	0x00000000
0x00A8	ASSIGN[GPIOB][PIN][PIN10]	GPIO mananger	0x00000000
0x00AC	ASSIGN[GPIOB][PIN][PIN11]	GPIO mananger	0x00000000
0x00B0	ASSIGN[GPIOB][PIN][PIN12]	GPIO mananger	0x00000000
0x00B4	ASSIGN[GPIOB][PIN][PIN13]	GPIO mananger	0x00000000
0x00B8	ASSIGN[GPIOB][PIN][PIN14]	GPIO mananger	0x00000000
0x00BC	ASSIGN[GPIOB][PIN][PIN15]	GPIO mananger	0x00000000
0x00C0	ASSIGN[GPIOB][PIN][PIN16]	GPIO mananger	0x00000000
0x00C4	ASSIGN[GPIOB][PIN][PIN17]	GPIO mananger	0x00000000
0x00C8	ASSIGN[GPIOB][PIN][PIN18]	GPIO mananger	0x00000000
0x00CC	ASSIGN[GPIOB][PIN][PIN19]	GPIO mananger	0x00000000
0x00D0	ASSIGN[GPIOB][PIN][PIN20]	GPIO mananger	0x00000000
0x00D4	ASSIGN[GPIOB][PIN][PIN21]	GPIO mananger	0x00000000
0x00D8	ASSIGN[GPIOB][PIN][PIN22]	GPIO mananger	0x00000000
0x00DC	ASSIGN[GPIOB][PIN][PIN23]	GPIO mananger	0x00000000
0x00E0	ASSIGN[GPIOB][PIN][PIN24]	GPIO mananger	0x00000000
0x00E4	ASSIGN[GPIOB][PIN][PIN25]	GPIO mananger	0x00000000
0x00E8	ASSIGN[GPIOB][PIN][PIN26]	GPIO mananger	0x00000000
0x00EC	ASSIGN[GPIOB][PIN][PIN27]	GPIO mananger	0x00000000
0x00F0	ASSIGN[GPIOB][PIN][PIN28]	GPIO mananger	0x00000000
0x00F4	ASSIGN[GPIOB][PIN][PIN29]	GPIO mananger	0x00000000
0x00F8	ASSIGN[GPIOB][PIN][PIN30]	GPIO mananger	0x00000000
0x00FC	ASSIGN[GPIOB][PIN][PIN31]	GPIO mananger	0x00000000
0x0100	ASSIGN[GPIOC][PIN][PIN00]	GPIO mananger	0x00000000
0x0104	ASSIGN[GPIOC][PIN][PIN01]	GPIO mananger	0x00000000
0x0108	ASSIGN[GPIOC][PIN][PIN02]	GPIO mananger	0x00000000
0x010C	ASSIGN[GPIOC][PIN][PIN03]	GPIO mananger	0x00000000
0x0110	ASSIGN[GPIOC][PIN][PIN04]	GPIO mananger	0x00000000
0x0114	ASSIGN[GPIOC][PIN][PIN05]	GPIO mananger	0x00000000
0x0118	ASSIGN[GPIOC][PIN][PIN06]	GPIO mananger	0x00000000
0x011C	ASSIGN[GPIOC][PIN][PIN07]	GPIO mananger	0x00000000
0x0120	ASSIGN[GPIOC][PIN][PIN08]	GPIO mananger	0x00000000
0x0124	ASSIGN[GPIOC][PIN][PIN09]	GPIO mananger	0x00000000
0x0128	ASSIGN[GPIOC][PIN][PIN10]	GPIO mananger	0x00000000
0x012C	ASSIGN[GPIOC][PIN][PIN11]	GPIO mananger	0x00000000

Offset	Register Name	Description	Reset Value
0x0130	ASSIGN[GPIOC][PIN][PIN12]	GPIO mananger	0x00000000
0x0134	ASSIGN[GPIOC][PIN][PIN13]	GPIO mananger	0x00000000
0x0138	ASSIGN[GPIOC][PIN][PIN14]	GPIO mananger	0x00000000
0x013C	ASSIGN[GPIOC][PIN][PIN15]	GPIO mananger	0x00000000
0x0140	ASSIGN[GPIOC][PIN][PIN16]	GPIO mananger	0x00000000
0x0144	ASSIGN[GPIOC][PIN][PIN17]	GPIO mananger	0x00000000
0x0148	ASSIGN[GPIOC][PIN][PIN18]	GPIO mananger	0x00000000
0x014C	ASSIGN[GPIOC][PIN][PIN19]	GPIO mananger	0x00000000
0x0150	ASSIGN[GPIOC][PIN][PIN20]	GPIO mananger	0x00000000
0x0154	ASSIGN[GPIOC][PIN][PIN21]	GPIO mananger	0x00000000
0x0158	ASSIGN[GPIOC][PIN][PIN22]	GPIO mananger	0x00000000
0x015C	ASSIGN[GPIOC][PIN][PIN23]	GPIO mananger	0x00000000
0x0160	ASSIGN[GPIOC][PIN][PIN24]	GPIO mananger	0x00000000
0x0164	ASSIGN[GPIOC][PIN][PIN25]	GPIO mananger	0x00000000
0x0168	ASSIGN[GPIOC][PIN][PIN26]	GPIO mananger	0x00000000
0x016C	ASSIGN[GPIOC][PIN][PIN27]	GPIO mananger	0x00000000
0x0170	ASSIGN[GPIOC][PIN][PIN28]	GPIO mananger	0x00000000
0x0174	ASSIGN[GPIOC][PIN][PIN29]	GPIO mananger	0x00000000
0x0178	ASSIGN[GPIOC][PIN][PIN30]	GPIO mananger	0x00000000
0x017C	ASSIGN[GPIOC][PIN][PIN31]	GPIO mananger	0x00000000
0x0180	ASSIGN[GPIOD][PIN][PIN00]	GPIO mananger	0x00000000
0x0184	ASSIGN[GPIOD][PIN][PIN01]	GPIO mananger	0x00000000
0x0188	ASSIGN[GPIOD][PIN][PIN02]	GPIO mananger	0x00000000
0x018C	ASSIGN[GPIOD][PIN][PIN03]	GPIO mananger	0x00000000
0x0190	ASSIGN[GPIOD][PIN][PIN04]	GPIO mananger	0x00000000
0x0194	ASSIGN[GPIOD][PIN][PIN05]	GPIO mananger	0x00000000
0x0198	ASSIGN[GPIOD][PIN][PIN06]	GPIO mananger	0x00000000
0x019C	ASSIGN[GPIOD][PIN][PIN07]	GPIO mananger	0x00000000
0x01A0	ASSIGN[GPIOD][PIN][PIN08]	GPIO mananger	0x00000000
0x01A4	ASSIGN[GPIOD][PIN][PIN09]	GPIO mananger	0x00000000
0x01A8	ASSIGN[GPIOD][PIN][PIN10]	GPIO mananger	0x00000000
0x01AC	ASSIGN[GPIOD][PIN][PIN11]	GPIO mananger	0x00000000
0x01B0	ASSIGN[GPIOD][PIN][PIN12]	GPIO mananger	0x00000000
0x01B4	ASSIGN[GPIOD][PIN][PIN13]	GPIO mananger	0x00000000
0x01B8	ASSIGN[GPIOD][PIN][PIN14]	GPIO mananger	0x00000000
0x01BC	ASSIGN[GPIOD][PIN][PIN15]	GPIO mananger	0x00000000
0x01C0	ASSIGN[GPIOD][PIN][PIN16]	GPIO mananger	0x00000000
0x01C4	ASSIGN[GPIOD][PIN][PIN17]	GPIO mananger	0x00000000
0x01C8	ASSIGN[GPIOD][PIN][PIN18]	GPIO mananger	0x00000000
0x01CC	ASSIGN[GPIOD][PIN][PIN19]	GPIO mananger	0x00000000

Offset	Register Name	Description	Reset Value
0x01D0	ASSIGN[GPIOD][PIN][PIN20]	GPIO mananger	0x00000000
0x01D4	ASSIGN[GPIOD][PIN][PIN21]	GPIO mananger	0x00000000
0x01D8	ASSIGN[GPIOD][PIN][PIN22]	GPIO mananger	0x00000000
0x01DC	ASSIGN[GPIOD][PIN][PIN23]	GPIO mananger	0x00000000
0x01E0	ASSIGN[GPIOD][PIN][PIN24]	GPIO mananger	0x00000000
0x01E4	ASSIGN[GPIOD][PIN][PIN25]	GPIO mananger	0x00000000
0x01E8	ASSIGN[GPIOD][PIN][PIN26]	GPIO mananger	0x00000000
0x01EC	ASSIGN[GPIOD][PIN][PIN27]	GPIO mananger	0x00000000
0x01F0	ASSIGN[GPIOD][PIN][PIN28]	GPIO mananger	0x00000000
0x01F4	ASSIGN[GPIOD][PIN][PIN29]	GPIO mananger	0x00000000
0x01F8	ASSIGN[GPIOD][PIN][PIN30]	GPIO mananger	0x00000000
0x01FC	ASSIGN[GPIOD][PIN][PIN31]	GPIO mananger	0x00000000
0x0200	ASSIGN[GPIOE][PIN][PIN00]	GPIO mananger	0x00000000
0x0204	ASSIGN[GPIOE][PIN][PIN01]	GPIO mananger	0x00000000
0x0208	ASSIGN[GPIOE][PIN][PIN02]	GPIO mananger	0x00000000
0x020C	ASSIGN[GPIOE][PIN][PIN03]	GPIO mananger	0x00000000
0x0210	ASSIGN[GPIOE][PIN][PIN04]	GPIO mananger	0x00000000
0x0214	ASSIGN[GPIOE][PIN][PIN05]	GPIO mananger	0x00000000
0x0218	ASSIGN[GPIOE][PIN][PIN06]	GPIO mananger	0x00000000
0x021C	ASSIGN[GPIOE][PIN][PIN07]	GPIO mananger	0x00000000
0x0220	ASSIGN[GPIOE][PIN][PIN08]	GPIO mananger	0x00000000
0x0224	ASSIGN[GPIOE][PIN][PIN09]	GPIO mananger	0x00000000
0x0228	ASSIGN[GPIOE][PIN][PIN10]	GPIO mananger	0x00000000
0x022C	ASSIGN[GPIOE][PIN][PIN11]	GPIO mananger	0x00000000
0x0230	ASSIGN[GPIOE][PIN][PIN12]	GPIO mananger	0x00000000
0x0234	ASSIGN[GPIOE][PIN][PIN13]	GPIO mananger	0x00000000
0x0238	ASSIGN[GPIOE][PIN][PIN14]	GPIO mananger	0x00000000
0x023C	ASSIGN[GPIOE][PIN][PIN15]	GPIO mananger	0x00000000
0x0240	ASSIGN[GPIOE][PIN][PIN16]	GPIO mananger	0x00000000
0x0244	ASSIGN[GPIOE][PIN][PIN17]	GPIO mananger	0x00000000
0x0248	ASSIGN[GPIOE][PIN][PIN18]	GPIO mananger	0x00000000
0x024C	ASSIGN[GPIOE][PIN][PIN19]	GPIO mananger	0x00000000
0x0250	ASSIGN[GPIOE][PIN][PIN20]	GPIO mananger	0x00000000
0x0254	ASSIGN[GPIOE][PIN][PIN21]	GPIO mananger	0x00000000
0x0258	ASSIGN[GPIOE][PIN][PIN22]	GPIO mananger	0x00000000
0x025C	ASSIGN[GPIOE][PIN][PIN23]	GPIO mananger	0x00000000
0x0260	ASSIGN[GPIOE][PIN][PIN24]	GPIO mananger	0x00000000
0x0264	ASSIGN[GPIOE][PIN][PIN25]	GPIO mananger	0x00000000
0x0268	ASSIGN[GPIOE][PIN][PIN26]	GPIO mananger	0x00000000
0x026C	ASSIGN[GPIOE][PIN][PIN27]	GPIO mananger	0x00000000

Offset	Register Name	Description	Reset Value
0x0270	ASSIGN[GPIOE][PIN][PIN28]	GPIO mananger	0x00000000
0x0274	ASSIGN[GPIOE][PIN][PIN29]	GPIO mananger	0x00000000
0x0278	ASSIGN[GPIOE][PIN][PIN30]	GPIO mananger	0x00000000
0x027C	ASSIGN[GPIOE][PIN][PIN31]	GPIO mananger	0x00000000
0x0280	ASSIGN[GPIOF][PIN][PIN00]	GPIO mananger	0x00000000
0x0284	ASSIGN[GPIOF][PIN][PIN01]	GPIO mananger	0x00000000
0x0288	ASSIGN[GPIOF][PIN][PIN02]	GPIO mananger	0x00000000
0x028C	ASSIGN[GPIOF][PIN][PIN03]	GPIO mananger	0x00000000
0x0290	ASSIGN[GPIOF][PIN][PIN04]	GPIO mananger	0x00000000
0x0294	ASSIGN[GPIOF][PIN][PIN05]	GPIO mananger	0x00000000
0x0298	ASSIGN[GPIOF][PIN][PIN06]	GPIO mananger	0x00000000
0x029C	ASSIGN[GPIOF][PIN][PIN07]	GPIO mananger	0x00000000
0x02A0	ASSIGN[GPIOF][PIN][PIN08]	GPIO mananger	0x00000000
0x02A4	ASSIGN[GPIOF][PIN][PIN09]	GPIO mananger	0x00000000
0x02A8	ASSIGN[GPIOF][PIN][PIN10]	GPIO mananger	0x00000000
0x02AC	ASSIGN[GPIOF][PIN][PIN11]	GPIO mananger	0x00000000
0x02B0	ASSIGN[GPIOF][PIN][PIN12]	GPIO mananger	0x00000000
0x02B4	ASSIGN[GPIOF][PIN][PIN13]	GPIO mananger	0x00000000
0x02B8	ASSIGN[GPIOF][PIN][PIN14]	GPIO mananger	0x00000000
0x02BC	ASSIGN[GPIOF][PIN][PIN15]	GPIO mananger	0x00000000
0x02C0	ASSIGN[GPIOF][PIN][PIN16]	GPIO mananger	0x00000000
0x02C4	ASSIGN[GPIOF][PIN][PIN17]	GPIO mananger	0x00000000
0x02C8	ASSIGN[GPIOF][PIN][PIN18]	GPIO mananger	0x00000000
0x02CC	ASSIGN[GPIOF][PIN][PIN19]	GPIO mananger	0x00000000
0x02D0	ASSIGN[GPIOF][PIN][PIN20]	GPIO mananger	0x00000000
0x02D4	ASSIGN[GPIOF][PIN][PIN21]	GPIO mananger	0x00000000
0x02D8	ASSIGN[GPIOF][PIN][PIN22]	GPIO mananger	0x00000000
0x02DC	ASSIGN[GPIOF][PIN][PIN23]	GPIO mananger	0x00000000
0x02E0	ASSIGN[GPIOF][PIN][PIN24]	GPIO mananger	0x00000000
0x02E4	ASSIGN[GPIOF][PIN][PIN25]	GPIO mananger	0x00000000
0x02E8	ASSIGN[GPIOF][PIN][PIN26]	GPIO mananger	0x00000000
0x02EC	ASSIGN[GPIOF][PIN][PIN27]	GPIO mananger	0x00000000
0x02F0	ASSIGN[GPIOF][PIN][PIN28]	GPIO mananger	0x00000000
0x02F4	ASSIGN[GPIOF][PIN][PIN29]	GPIO mananger	0x00000000
0x02F8	ASSIGN[GPIOF][PIN][PIN30]	GPIO mananger	0x00000000
0x02FC	ASSIGN[GPIOF][PIN][PIN31]	GPIO mananger	0x00000000
0x0580	ASSIGN[GPIOV][PIN][PIN00]	GPIO mananger	0x00000000
0x0584	ASSIGN[GPIOV][PIN][PIN01]	GPIO mananger	0x00000000
0x0588	ASSIGN[GPIOV][PIN][PIN02]	GPIO mananger	0x00000000
0x058C	ASSIGN[GPIOV][PIN][PIN03]	GPIO mananger	0x00000000

Offset	Register Name	Description	Reset Value
0x0590	ASSIGN[GPIOV][PIN][PIN04]	GPIO mananger	0x00000000
0x0594	ASSIGN[GPIOV][PIN][PIN05]	GPIO mananger	0x00000000
0x0598	ASSIGN[GPIOV][PIN][PIN06]	GPIO mananger	0x00000000
0x059C	ASSIGN[GPIOV][PIN][PIN07]	GPIO mananger	0x00000000
0x05A0	ASSIGN[GPIOV][PIN][PIN08]	GPIO mananger	0x00000000
0x05A4	ASSIGN[GPIOV][PIN][PIN09]	GPIO mananger	0x00000000
0x05A8	ASSIGN[GPIOV][PIN][PIN10]	GPIO mananger	0x00000000
0x05AC	ASSIGN[GPIOV][PIN][PIN11]	GPIO mananger	0x00000000
0x05B0	ASSIGN[GPIOV][PIN][PIN12]	GPIO mananger	0x00000000
0x05B4	ASSIGN[GPIOV][PIN][PIN13]	GPIO mananger	0x00000000
0x05B8	ASSIGN[GPIOV][PIN][PIN14]	GPIO mananger	0x00000000
0x05BC	ASSIGN[GPIOV][PIN][PIN15]	GPIO mananger	0x00000000
0x05C0	ASSIGN[GPIOV][PIN][PIN16]	GPIO mananger	0x00000000
0x05C4	ASSIGN[GPIOV][PIN][PIN17]	GPIO mananger	0x00000000
0x05C8	ASSIGN[GPIOV][PIN][PIN18]	GPIO mananger	0x00000000
0x05CC	ASSIGN[GPIOV][PIN][PIN19]	GPIO mananger	0x00000000
0x05D0	ASSIGN[GPIOV][PIN][PIN20]	GPIO mananger	0x00000000
0x05D4	ASSIGN[GPIOV][PIN][PIN21]	GPIO mananger	0x00000000
0x05D8	ASSIGN[GPIOV][PIN][PIN22]	GPIO mananger	0x00000000
0x05DC	ASSIGN[GPIOV][PIN][PIN23]	GPIO mananger	0x00000000
0x05E0	ASSIGN[GPIOV][PIN][PIN24]	GPIO mananger	0x00000000
0x05E4	ASSIGN[GPIOV][PIN][PIN25]	GPIO mananger	0x00000000
0x05E8	ASSIGN[GPIOV][PIN][PIN26]	GPIO mananger	0x00000000
0x05EC	ASSIGN[GPIOV][PIN][PIN27]	GPIO mananger	0x00000000
0x05F0	ASSIGN[GPIOV][PIN][PIN28]	GPIO mananger	0x00000000
0x05F4	ASSIGN[GPIOV][PIN][PIN29]	GPIO mananger	0x00000000
0x05F8	ASSIGN[GPIOV][PIN][PIN30]	GPIO mananger	0x00000000
0x05FC	ASSIGN[GPIOV][PIN][PIN31]	GPIO mananger	0x00000000
0x0600	ASSIGN[GPIOW][PIN][PIN00]	GPIO mananger	0x00000000
0x0604	ASSIGN[GPIOW][PIN][PIN01]	GPIO mananger	0x00000000
0x0608	ASSIGN[GPIOW][PIN][PIN02]	GPIO mananger	0x00000000
0x060C	ASSIGN[GPIOW][PIN][PIN03]	GPIO mananger	0x00000000
0x0610	ASSIGN[GPIOW][PIN][PIN04]	GPIO mananger	0x00000000
0x0614	ASSIGN[GPIOW][PIN][PIN05]	GPIO mananger	0x00000000
0x0618	ASSIGN[GPIOW][PIN][PIN06]	GPIO mananger	0x00000000
0x061C	ASSIGN[GPIOW][PIN][PIN07]	GPIO mananger	0x00000000
0x0620	ASSIGN[GPIOW][PIN][PIN08]	GPIO mananger	0x00000000
0x0624	ASSIGN[GPIOW][PIN][PIN09]	GPIO mananger	0x00000000
0x0628	ASSIGN[GPIOW][PIN][PIN10]	GPIO mananger	0x00000000
0x062C	ASSIGN[GPIOW][PIN][PIN11]	GPIO mananger	0x00000000

Offset	Register Name	Description	Reset Value
0x0630	ASSIGN[GPIOW][PIN][PIN12]	GPIO mananger	0x00000000
0x0634	ASSIGN[GPIOW][PIN][PIN13]	GPIO mananger	0x00000000
0x0638	ASSIGN[GPIOW][PIN][PIN14]	GPIO mananger	0x00000000
0x063C	ASSIGN[GPIOW][PIN][PIN15]	GPIO mananger	0x00000000
0x0640	ASSIGN[GPIOW][PIN][PIN16]	GPIO mananger	0x00000000
0x0644	ASSIGN[GPIOW][PIN][PIN17]	GPIO mananger	0x00000000
0x0648	ASSIGN[GPIOW][PIN][PIN18]	GPIO mananger	0x00000000
0x064C	ASSIGN[GPIOW][PIN][PIN19]	GPIO mananger	0x00000000
0x0650	ASSIGN[GPIOW][PIN][PIN20]	GPIO mananger	0x00000000
0x0654	ASSIGN[GPIOW][PIN][PIN21]	GPIO mananger	0x00000000
0x0658	ASSIGN[GPIOW][PIN][PIN22]	GPIO mananger	0x00000000
0x065C	ASSIGN[GPIOW][PIN][PIN23]	GPIO mananger	0x00000000
0x0660	ASSIGN[GPIOW][PIN][PIN24]	GPIO mananger	0x00000000
0x0664	ASSIGN[GPIOW][PIN][PIN25]	GPIO mananger	0x00000000
0x0668	ASSIGN[GPIOW][PIN][PIN26]	GPIO mananger	0x00000000
0x066C	ASSIGN[GPIOW][PIN][PIN27]	GPIO mananger	0x00000000
0x0670	ASSIGN[GPIOW][PIN][PIN28]	GPIO mananger	0x00000000
0x0674	ASSIGN[GPIOW][PIN][PIN29]	GPIO mananger	0x00000000
0x0678	ASSIGN[GPIOW][PIN][PIN30]	GPIO mananger	0x00000000
0x067C	ASSIGN[GPIOW][PIN][PIN31]	GPIO mananger	0x00000000
0x0680	ASSIGN[GPIOX][PIN][PIN00]	GPIO mananger	0x00000000
0x0684	ASSIGN[GPIOX][PIN][PIN01]	GPIO mananger	0x00000000
0x0688	ASSIGN[GPIOX][PIN][PIN02]	GPIO mananger	0x00000000
0x068C	ASSIGN[GPIOX][PIN][PIN03]	GPIO mananger	0x00000000
0x0690	ASSIGN[GPIOX][PIN][PIN04]	GPIO mananger	0x00000000
0x0694	ASSIGN[GPIOX][PIN][PIN05]	GPIO mananger	0x00000000
0x0698	ASSIGN[GPIOX][PIN][PIN06]	GPIO mananger	0x00000000
0x069C	ASSIGN[GPIOX][PIN][PIN07]	GPIO mananger	0x00000000
0x06A0	ASSIGN[GPIOX][PIN][PIN08]	GPIO mananger	0x00000000
0x06A4	ASSIGN[GPIOX][PIN][PIN09]	GPIO mananger	0x00000000
0x06A8	ASSIGN[GPIOX][PIN][PIN10]	GPIO mananger	0x00000000
0x06AC	ASSIGN[GPIOX][PIN][PIN11]	GPIO mananger	0x00000000
0x06B0	ASSIGN[GPIOX][PIN][PIN12]	GPIO mananger	0x00000000
0x06B4	ASSIGN[GPIOX][PIN][PIN13]	GPIO mananger	0x00000000
0x06B8	ASSIGN[GPIOX][PIN][PIN14]	GPIO mananger	0x00000000
0x06BC	ASSIGN[GPIOX][PIN][PIN15]	GPIO mananger	0x00000000
0x06C0	ASSIGN[GPIOX][PIN][PIN16]	GPIO mananger	0x00000000
0x06C4	ASSIGN[GPIOX][PIN][PIN17]	GPIO mananger	0x00000000
0x06C8	ASSIGN[GPIOX][PIN][PIN18]	GPIO mananger	0x00000000
0x06CC	ASSIGN[GPIOX][PIN][PIN19]	GPIO mananger	0x00000000

Offset	Register Name	Description	Reset Value
0x06D0	ASSIGN[GPIOX][PIN][PIN20]	GPIO mananger	0x00000000
0x06D4	ASSIGN[GPIOX][PIN][PIN21]	GPIO mananger	0x00000000
0x06D8	ASSIGN[GPIOX][PIN][PIN22]	GPIO mananger	0x00000000
0x06DC	ASSIGN[GPIOX][PIN][PIN23]	GPIO mananger	0x00000000
0x06E0	ASSIGN[GPIOX][PIN][PIN24]	GPIO mananger	0x00000000
0x06E4	ASSIGN[GPIOX][PIN][PIN25]	GPIO mananger	0x00000000
0x06E8	ASSIGN[GPIOX][PIN][PIN26]	GPIO mananger	0x00000000
0x06EC	ASSIGN[GPIOX][PIN][PIN27]	GPIO mananger	0x00000000
0x06F0	ASSIGN[GPIOX][PIN][PIN28]	GPIO mananger	0x00000000
0x06F4	ASSIGN[GPIOX][PIN][PIN29]	GPIO mananger	0x00000000
0x06F8	ASSIGN[GPIOX][PIN][PIN30]	GPIO mananger	0x00000000
0x06FC	ASSIGN[GPIOX][PIN][PIN31]	GPIO mananger	0x00000000
0x0700	ASSIGN[GPIOY][PIN][PIN00]	GPIO mananger	0x00000000
0x0704	ASSIGN[GPIOY][PIN][PIN01]	GPIO mananger	0x00000000
0x0708	ASSIGN[GPIOY][PIN][PIN02]	GPIO mananger	0x00000000
0x070C	ASSIGN[GPIOY][PIN][PIN03]	GPIO mananger	0x00000000
0x0710	ASSIGN[GPIOY][PIN][PIN04]	GPIO mananger	0x00000000
0x0714	ASSIGN[GPIOY][PIN][PIN05]	GPIO mananger	0x00000000
0x0718	ASSIGN[GPIOY][PIN][PIN06]	GPIO mananger	0x00000000
0x071C	ASSIGN[GPIOY][PIN][PIN07]	GPIO mananger	0x00000000
0x0720	ASSIGN[GPIOY][PIN][PIN08]	GPIO mananger	0x00000000
0x0724	ASSIGN[GPIOY][PIN][PIN09]	GPIO mananger	0x00000000
0x0728	ASSIGN[GPIOY][PIN][PIN10]	GPIO mananger	0x00000000
0x072C	ASSIGN[GPIOY][PIN][PIN11]	GPIO mananger	0x00000000
0x0730	ASSIGN[GPIOY][PIN][PIN12]	GPIO mananger	0x00000000
0x0734	ASSIGN[GPIOY][PIN][PIN13]	GPIO mananger	0x00000000
0x0738	ASSIGN[GPIOY][PIN][PIN14]	GPIO mananger	0x00000000
0x073C	ASSIGN[GPIOY][PIN][PIN15]	GPIO mananger	0x00000000
0x0740	ASSIGN[GPIOY][PIN][PIN16]	GPIO mananger	0x00000000
0x0744	ASSIGN[GPIOY][PIN][PIN17]	GPIO mananger	0x00000000
0x0748	ASSIGN[GPIOY][PIN][PIN18]	GPIO mananger	0x00000000
0x074C	ASSIGN[GPIOY][PIN][PIN19]	GPIO mananger	0x00000000
0x0750	ASSIGN[GPIOY][PIN][PIN20]	GPIO mananger	0x00000000
0x0754	ASSIGN[GPIOY][PIN][PIN21]	GPIO mananger	0x00000000
0x0758	ASSIGN[GPIOY][PIN][PIN22]	GPIO mananger	0x00000000
0x075C	ASSIGN[GPIOY][PIN][PIN23]	GPIO mananger	0x00000000
0x0760	ASSIGN[GPIOY][PIN][PIN24]	GPIO mananger	0x00000000
0x0764	ASSIGN[GPIOY][PIN][PIN25]	GPIO mananger	0x00000000
0x0768	ASSIGN[GPIOY][PIN][PIN26]	GPIO mananger	0x00000000
0x076C	ASSIGN[GPIOY][PIN][PIN27]	GPIO mananger	0x00000000

Offset	Register Name	Description	Reset Value
0x0770	ASSIGN[GPIOW][PIN][PIN28]	GPIO manager	0x00000000
0x0774	ASSIGN[GPIOW][PIN][PIN29]	GPIO manager	0x00000000
0x0778	ASSIGN[GPIOW][PIN][PIN30]	GPIO manager	0x00000000
0x077C	ASSIGN[GPIOW][PIN][PIN31]	GPIO manager	0x00000000

Table 149: GPIOM Register List

23.4 GPIOM register description

The register details of GPIOM are as follows:

23.4.1 ASSIGN[PIN] (0x0 + 0x80 * n + 0x4 * m)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LOCK	RSVD										HIDE										RSVD										SELECT
RW	N/A										RW										N/A										RW
0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	x	x	x	x	x	x	0	0

ASSIGN[PIN] [31:0]

Bits	Name	Description
31	LOCK	lock fields in this register, lock can only be cleared by soc reset 0: fields can be changed 1: fields locked to current value, not changeable
11-8	HIDE	pin value visibility to gpios, bit0: 1, invisible to soc gpio0; 0: visible to soc gpio0 bit2: 1, invisible to cpu0 fast gpio; 0: visible to cpu0 fast gpio bit3: 1, invisible to cpu1 fast gpio; 0: visible to cpu1 fast gpio
1-0	SELECT	select which gpio controls chip pin, 0: soc gpio0; 2: cpu0 fastgpio 3: cpu1 fast gpio

ASSIGN[PIN] Bits

24 EUI

This chapter introduces the functionality and features of the EUI.

24.1 Feature Summary

This section introduces the main features of the EUI:

- Supports 2 to 8 column select signals.
- Supports up to 128 independent keys, up to 8 seven-segment digital tubes or 4 sixteen-segment digital tubes.
- Supports key-on state detection interrupt.
- Supports configurable key press debounce length.
- Supports configurable timing parameters (output clock frequency, data refresh frequency, loop frequency, etc.).
- Supports 8 different sets of LED display content (64 bits per set), which can be displayed in pairs sequentially, or cycled between 4 pairs.
- Supports input and output signal inversion.

24.2 EUI Block Diagram

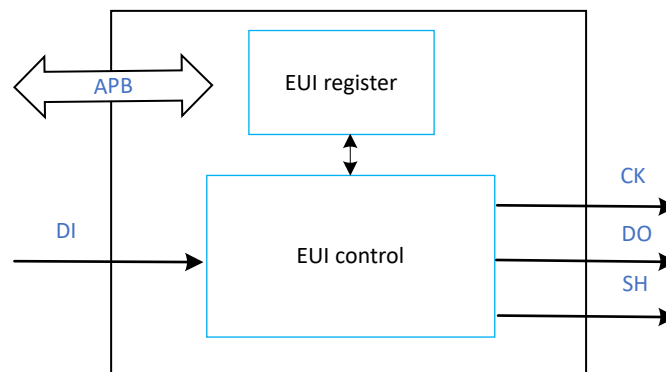


Figure 22: EUI Block Diagram

24.3 Functional Description

This section describes the functions of the EUI module. The EUI module can send serial data outward to drive external digital tubes (such as 74-series 595 chips) for data display. Additionally, the EUI module can also receive serial data input from external shift registers (such as 74-series 165 chips).

The connection diagram for the EUI module driving two cascaded shift registers (for a 7-segment digital tube) is as follows. Here, set the display mode of the EUI module to 8 * 8 (configured in the [DISP_MODE] field of the CTRLC register). The EUI module sends data outward to the external cascaded digital tube shift registers. After receiving the serial data from the EUI module, the external shift register converts it to parallel data to drive the digital tube for display.

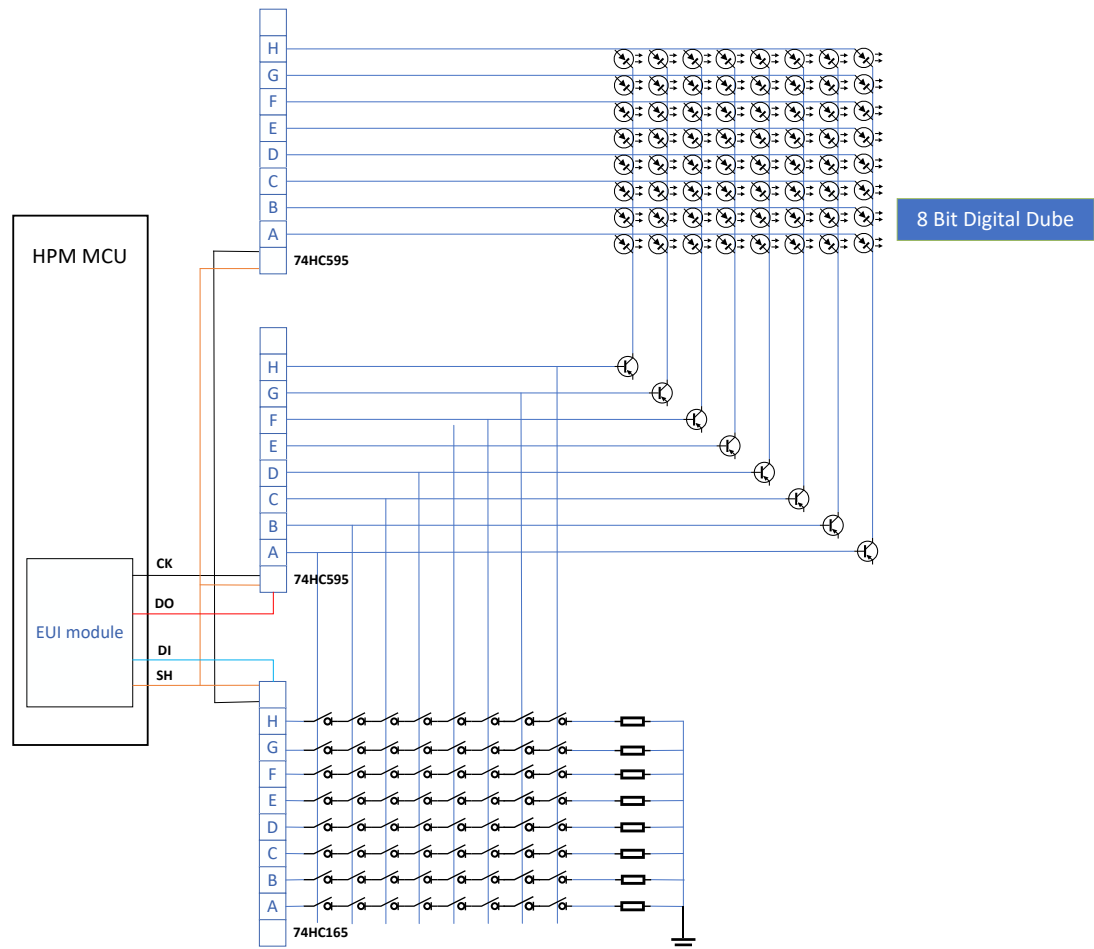


Figure 23: Schematic Diagram of EUI Driving 7-Segment Digital Tubes and Receiving from Shift Register

Additionally, the EUI module reads data from external shift registers. As shown in the figure below, keyboard keys can control changes in the shift register data. When an external keyboard key press causes data change in an external shift register (e.g., a 74-series 165 chip), the EUI module can read its data via the serial port, thereby obtaining the changed value to determine which key was triggered.

If the digital tube type is different, such as a 16-segment tube, the EUI display mode setting can be adjusted in the [DISP_MODE] field of the CTRLC register (set its mode to $16 * 4$). The connection diagram is shown below. The EUI module sends data outward to the external cascaded digital tube shift registers. After receiving the serial data from the EUI module, the external shift register converts it to parallel data to drive the digital tube for display.

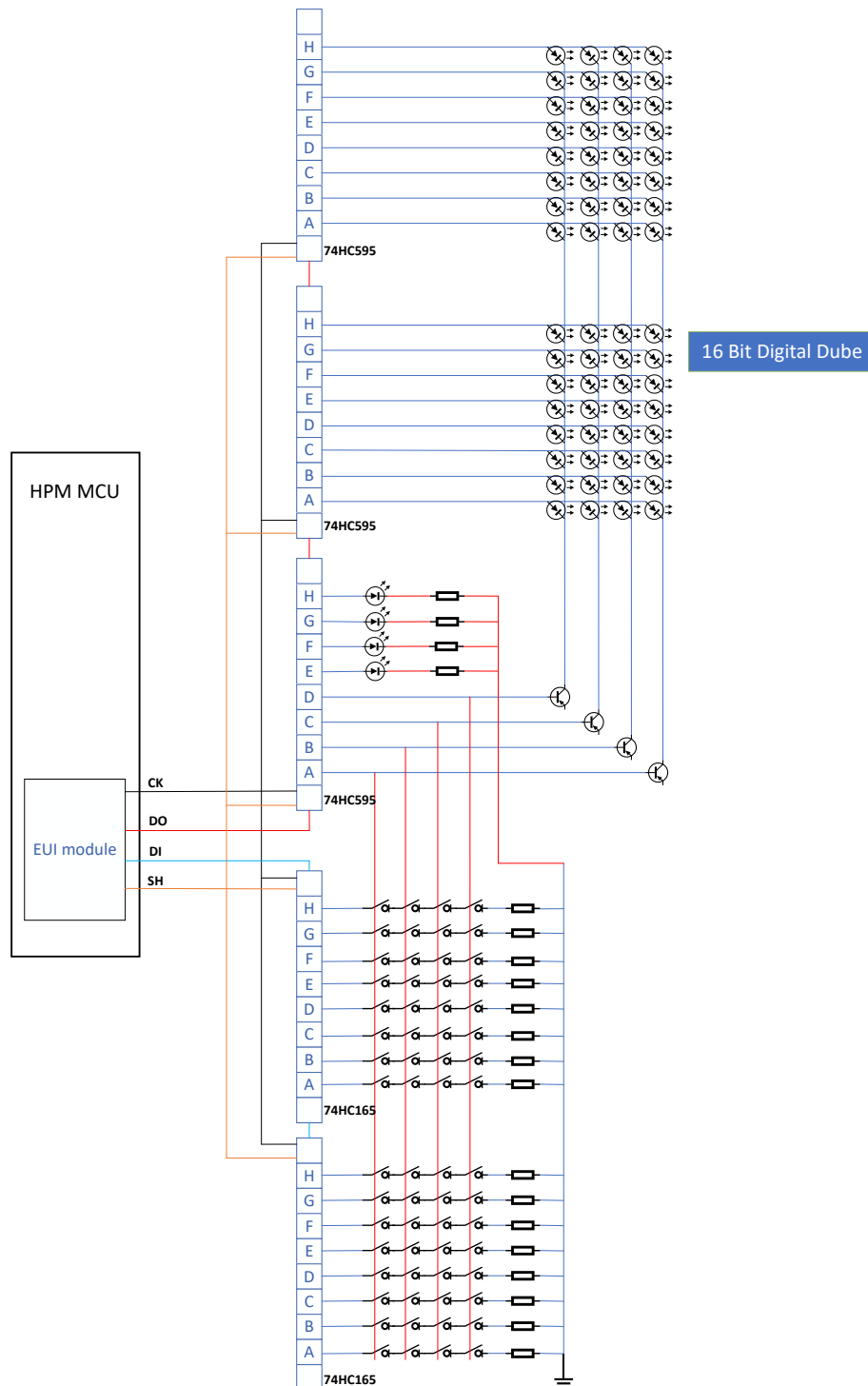


Figure 24: Schematic Diagram of EUI Driving 16-Segment Digital Tubes and Receiving from Shift Register

24.3.1 EUI Port Interface Description

Signal Name	I/O	Signal Description
DI	I	Serial Data Input signal
CK	O	Serial Data Output Clock

Signal Name	I/O	Signal Description
DO	O	Serial Data Output signal
SH	O	Data Latch Clock signal line

Table 150: EUI Port Interface Description

24.3.2 Output Mode and Display

The EUI module sends data on the falling edge of the clock, and external shift registers can receive data on the rising edge. The EUI module can send out 24 bits of data, which is 24-bit data. When the EUI module is connected to digital tubes, the output data includes display data and column select data. Based on the set display mode, the width of the data and the column select data can be changed. For example, setting the display mode to 8 * 8 mode in the [DISP_MODE] bit field of the CTRL register (offset address 0x000) will generate 8-bit data and 8-bit column select data. If the display mode is set to 16 * 4, it will generate 16-bit data and 4-bit column select data. The number of clock cycles per bit is configured in the [CLK_DIV] bit field of the DIV_CFG register (offset address 0x004).

24.3.3 Drive Timing Parameters

The EUI module outputs serial data to drive digital tubes, with timing parameters such as output period (CLKO_TIME), slot time (SLOT_TEME), hold time (HOLD_TIME), display time (DISP_TIME), and loop time (LOOP_TIME). The related schematic diagram is shown below.

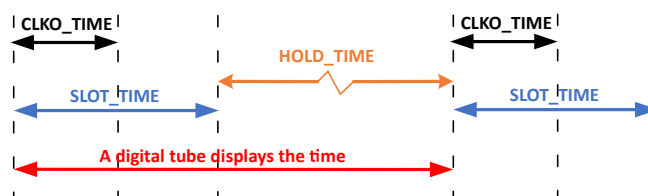


Figure 25: Schematic Diagram of EUI Digital Tube Drive Timing Parameters

Output Period (CLKO_TIME) is the clock period for sending data outward, derived by dividing the EUI module's system clock. The default output period is 1 us. Users can configure it in the [CLK_DIV] bit field of the DIV_CFG register to obtain the desired output period.

Slot Time (SLOT_TEME) consists of 32 output clock periods. Within these 32 clock cycles, the first 24 are used for data transmission, and the remaining time is for external digital tube data storage and control signal generation, etc.

Hold Time (HOLD_TIME) uses slot time as its unit. The default hold time is set to 0, so data is transmitted every slot time. Users can configure the required hold time in the [HOLD_TIME] bit field of the DIV_CFG register (offset address 0x004) as needed, which will increase the interval between sending data outward.

Display Time (DISP_TIME) is the time required to refresh the external LEDs and to fully detect a key press (on-state), with a default setting of 1 ms. The formula for display time is as follows, where N represents the number of digital tubes.

$$\text{DISP\_TIME} = (\text{HOLD\_TIME} + 1) * \text{SLOT\_TEME} * N \quad (1)$$

Key Debounce Time (FILTER_TIME) uses display time as its unit, with a default setting of 500 us in 16 * 4 mode (with only 4 column select signals).

Loop Time (LOOP_TIME) is the display time multiplied by the number of digital tube groups. For example, the loop time for 8 groups is the product of 16 and the display time.

The display of digital tubes driven by the EUI module can be configured according to user needs, in the [DB_CTRL] bit field of the DAB_CTR register (offset address 0x00c). For example, if the data for each group is A and B, when displaying A, data A itself can be output, and when displaying B, data B itself can also be output. Additionally, configurations can display $A \& B$, $A | B$, $A \wedge B$.

24.3.4 Receiving Serial Data

Use the CK port of the EUI as the clock for the shift register, and connect the shift register's data output to the DI pin of the EUI module, thus allowing the EUI module to receive serial data input from the external shift register.

24.3.5 Interrupt Function Description

The EUI module supports zone key-on exception and independent key-on exception interrupts. When it is detected that the key press on-state in a zone has not changed for longer than the debounce time, a zone key-on exception interrupt is triggered. When it is detected that the independent key on-state has not changed for longer than the debounce time, an independent key-on exception interrupt is triggered.

Note: After power-on, if the time during which the data remains 0 exceeds the debounce time, an interrupt will also be triggered once.

24.3.6 Register Configuration Guide

1. Driving External Shift Registers to Drive Digital Tubes

1. Configure the [CLKO_ADJ] bit field of the DIV_CFG register (offset address 0x004) as needed to perform internal frequency division in the EUI, obtaining the output clock frequency for external data;
2. Configure the [HOLD_TIME] and [CLK_DIV] bit fields of the DIV_CFG register (offset address 0x004) as needed to set the hold time and obtain the desired output clock frequency;
3. Optionally, configure the filter length in the [FILT_LEN] bit field of the PAD_CFG register (offset address 0x008) to eliminate jitter;
4. As needed, write the output data to the [DATA_POL] bit field of the POL_CFG register (offset address 0x00C) (if polarity inversion is required, also configure the [SH_POL] bit field);
5. As needed, enable the required interrupt functions in the IRQ_EN register (offset address 0x024);
6. As needed, configure the display mode in the [DISP_MODE] bit field of the CTRL register (offset address 0x000), and then enable the EUI to send data out via the ENABLE bit field;

2. Receiving External Shift Register Input

1. Configure the [CLKO_ADJ] bit field of the DIV_CFG register (offset address 0x004) as needed to perform

internal frequency division in the EUI, obtaining the base frequency for output data;

2. Configure the [HOLD_TIME] and [CLK_DIV] bit fields of the DIV_CFG register (offset address 0x004) as needed to set the hold time and obtain the desired output clock frequency;

3. Optionally, configure the filter length in the [FILT_LEN] bit field of the PAD_CFG register (offset address 0x008) to eliminate jitter;

4. Enable the [IRQ_EREAS] bit field of the IRQ_EN register (offset address 0x024);

5. Configure the display mode in the [DISP_MODE] bit field of the CTRL register (offset address 0x000), and then enable the EUI in the [ENABLE] bit field;

6. After an interrupt is triggered, the received data can be read from the DATA_IN_0, DATA_IN1, DATA_IN_2, DATA_IN_3, DATA_IN_4, DATA_IN_5, DATA_IN_6, DATA_IN_7 registers;

24.4 EUI Registers

24.4.1 Register Overview

The list of EUI registers is as follows:

EUI0 base address: 0xF0038000

EUI1 base address: 0xF003C000

Offset	Register Name	Description	Reset Value
0x0000	CTRL	Control Register	0x00000000
0x0004	DIV_CFG		0x00000000
0x0008	PAD_CFG		0x00000000
0x000C	POL_CFG		0x00000000
0x0010	OUT_CFG		0x00000000
0x0014	IN_CFG		0x00000000
0x0020	IRQ_STS		0x00000000
0x0024	IRQ_EN		0x00000000
0x0040	DATA_IN0		0x00000000
0x0044	DATA_IN1		0x00000000
0x0048	DATA_IN2		0x00000000
0x004C	DATA_IN3		0x00000000
0x0050	DATA_IN4		0x00000000
0x0054	DATA_IN5		0x00000000
0x0058	DATA_IN6		0x00000000
0x005C	DATA_IN7		0x00000000
0x0080	DAB_CTRL		0x00000000
0x0090	DCD_CTRL		0x00000000
0x00A0	DEF_CTRL		0x00000000
0x00B0	DGH_CTRL		0x00000000
0x0100	DA_00		0x00000000
0x0104	DA_01		0x00000000

Offset	Register Name	Description	Reset Value
0x0140	DB_00		0x00000000
0x0144	DB_01		0x00000000
0x0180	DC_00		0x00000000
0x0184	DC_01		0x00000000
0x01C0	DD_00		0x00000000
0x01C4	DD_01		0x00000000
0x0200	DE_00		0x00000000
0x0204	DE_01		0x00000000
0x0240	DF_00		0x00000000
0x0244	DF_01		0x00000000
0x0280	DG_00		0x00000000
0x0284	DG_01		0x00000000
0x02C0	DH_00		0x00000000
0x02C4	DH_01		0x00000000

Table 151: EUI Register List

24.4.2 Register Details

The detailed description of EUI registers is as follows:

24.4.3 CTRL (0x0)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLR_OUTPUT	CLR_DATA_IN	RSVD																										COL_LOOP_MODE	DISP_MODE	ENABLE	
RW	WO	N/A																										RW	RW	RW	
0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	

CTRL [31:0]

Bits	Name	Description
31	CLR_OUTPUT	write 1 to send 24bit 0 to data output, Auto clear after clear finishd; only can be set when clock_enable(in eut_state) is 0
30	CLR_DATA_IN	write 1 to clear data_in register to all 0. only can be set when clock_enable(in eut_state) is 0
2	COL_LOOP_MODE	0: all configured LED on at same time in one column; 1: only one LED on in one column, need

Bits	Name	Description
1	DISP_MODE	0 for 8x8(8bit data,8bit select, need 2 external 595 chip, 1 for data, 1 for select) 1 for 16x4(16bit data, 4bit select, need 3 external 595 chip, 2 for data, 4bit for select, the other 4bit can be used as dedicate led control)
0	ENABLE	set to start block; clr to stop(hw will finish current loop)

CTRL Bits

24.4.4 DIV_CFG (0x4)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								CLKO_ADJ				HOLD_TIME				CLK_DIV															
N/A								RW				RW				RW															
x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DIV_CFG [31:0]

Bits	Name	Description
23-20	CLKO_ADJ	used to generate pulse signal for internal slow clock base, such as data change time. default 2 for data change 3 system clock cycles after negedge of coutput clock
19-16	HOLD_TIME	for display hold time, in unit of slot time(32 output clock cycles) shift 16 or 24 cycles in one slot time, then hold output for hold_time slots it need N*(hold_time+1) slot to display all LEDs, N is LED column select signal bits, valid value is 1 to 8 (default is 1024us in 8x8 mode, 512 us in 16x4 mode)
15-0	CLK_DIV	output clock divider of system clock. 0 for div2, 1 for div4; default 0x63 for 1MHz output clock at 200MHz system clock

DIV_CFG Bits

24.4.5 PAD_CFG (0x8)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																				FILT_LEN											
N/A																				RW											
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	

PAD_CFG [31:0]

Bits	Name	Description
11-0	FILT_LEN	switch filter length, use to remove jitter. In unit of disp_time. will assert interrupt if scan input data stable for more than this time. Default 5ms in 8x8 mode

PAD_CFG Bits

24.4.6 POL_CFG (0xC)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD				DI_POL	DO_POL	CK_POL	SH_POL	DATA_POL																							
N/A				RW	RW	RW	RW	RW																							
x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

POL_CFG [31:0]

Bits	Name	Description
27	DI_POL	polarity at pad_do_eut_in
26	DO_POL	polarity at pad_do_eut_out
25	CK_POL	polarity at pad_do_eut_ck
24	SH_POL	polarity at pad_do_eut_sh
23-0	DATA_POL	output polarity, set to invert output data at output shift register. NOTE: high 16bits are used in 8x8 mode since shift is always 24 cycles

POL_CFG Bits

24.4.7 OUT_CFG (0x10)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD					DO_CFG		DO_VAL																								

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
N/A					RW			RW																							
x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OUT_CFG [31:0]

Bits	Name	Description
26-24	DO_CFG	define how many dedicate led controls in 8x8 mode: 0: no dedicate led control, use 8bits(A~H) for led line select 1: do_val[7]/H is use as dedicate led control, 7bits(A~G) for led line select 2: do_val[7:6]/GH are use as dedicate led control, 6bits(A~F) for led line select ... 6: do_val[7:2]/C~H are use as dedicate led control, 2bit/AB for led line select 7: 24 bits are used as dedicate led control(do_val[23:0] is used) in 16x4 mode, if set to 0~4, 4bit(A~D)for led line select others are same as 8x8 mode
23-0	DO_VAL	dedicate led control values, this value will override output data(after do_pol)

OUT_CFG Bits

24.4.8 IN_CFG (0x14)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DI_CFG																DI_VAL															
RW																RO															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IN_CFG [31:0]

Bits	Name	Description
31-16	DI_CFG	set for input used as dedicate switch, will use dedicate filter logic and interrupt logic
15-0	DI_VAL	16bit input data

IN_CFG Bits

24.4.9 IRQ_STS (0x20)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																														IRQ_SINGLE_STS	IRQ_AREA_STS
N/A																														W1C	W1C
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0

IRQ_STS [31:0]

Bits	Name	Description
1	IRQ_SINGLE_STS	
0	IRQ_AREA_STS	

IRQ_STS Bits

24.4.10 IRQ_EN (0x24)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																														IRQ_SINGLE_EN	IRQ_AREA_EN
N/A																														RW	RW
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0

IRQ_EN [31:0]

Bits	Name	Description
1	IRQ_SINGLE_EN	
0	IRQ_AREA_EN	

IRQ_EN Bits

24.4.11 DATA_IN0 (0x40)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																DATA															
N/A																RO															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DATA_IN0 [31:0]

Bits	Name	Description
15-0	DATA	read value when select bit 0(0x01 for output value in 8x8 mode)

DATA_IN0 Bits

24.4.12 DATA_IN1 (0x44)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																DATA															
N/A																RO															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DATA_IN1 [31:0]

Bits	Name	Description
15-0	DATA	

DATA_IN1 Bits

24.4.13 DATA_IN2 (0x48)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																DATA															
N/A																RO															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DATA_IN2 [31:0]

Bits	Name	Description
15-0	DATA	

DATA_IN2 Bits

24.4.14 DATA_IN3 (0x4C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																DATA															
N/A																RO															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DATA_IN3 [31:0]

Bits	Name	Description
15-0	DATA	

DATA_IN3 Bits

24.4.15 DATA_IN4 (0x50)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																DATA															
N/A																RO															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DATA_IN4 [31:0]

Bits	Name	Description
15-0	DATA	

DATA_IN4 Bits

24.4.16 DATA_IN5 (0x54)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																DATA															
N/A																RO															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DATA_IN5 [31:0]

Bits	Name	Description
15-0	DATA	

DATA_IN5 Bits

24.4.17 DATA_IN6 (0x58)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																DATA															
N/A																RO															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DATA_IN6 [31:0]

Bits	Name	Description
15-0	DATA	

DATA_IN6 Bits

24.4.18 DATA_IN7 (0x5C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																DATA															
N/A																RO															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DATA_IN7 [31:0]

Bits	Name	Description
15-0	DATA	

DATA_IN7 Bits

24.4.19 DAB_CTRL (0x80)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DB_CTRL		RSVD							LOOPAB								PRDB							PRDA							
RW		N/A							RW								RW							RW							
0	0	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

DAB_CTRL [31:0]

Bits	Name	Description
31-30	DB_CTRL	00: output data B directly; 01: A&B; 10: A B; 11: A^B
23-16	LOOPAB	display AB loop display data A for prda, then B for prdB, then lopp AB; then display C, D(loop CD), EF, GH, then back to AB 0 for no loop, just display once
15-8	PRDB	display data B time(in unit of 16*disp_time), default 65ms(65536us)
7-0	PRDA	display data A time(in unit of 16*disp_time) dab_ctrl.prda must be none-zero value, others can be set to 0 if no data

DAB_CTRL Bits

24.4.20 DCD_CTRL (0x90)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DD_CTRL		RSVD							LOOPCD							PRDD							PRDC								
RW		N/A							RW							RW							RW								
0	0	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

DCD_CTRL [31:0]

Bits	Name	Description
31-30	DD_CTRL	
23-16	LOOPCD	
15-8	PRDD	
7-0	PRDC	

DCD_CTRL Bits

24.4.21 DEF_CTRL (0xA0)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DF_CTRL		RSVD							LOOPEF							PRDF							PRDE								
RW		N/A							RW							RW							RW								
0	0	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

DEF_CTRL [31:0]

Bits	Name	Description
31-30	DF_CTRL	
23-16	LOOPEF	
15-8	PRDF	
7-0	PRDE	

DEF_CTRL Bits

24.4.22 DGH_CTRL (0xB0)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DH_CTRL		RSVD							LOOPGH							PRDH							PRDG								
RW		N/A							RW							RW							RW								
0	0	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

DGH_CTRL [31:0]

Bits	Name	Description
31-30	DH_CTRL	
23-16	LOOPGH	
15-8	PRDH	
7-0	PRDG	

DGH_CTRL Bits

24.4.23 DA_00 (0x100)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DA_00 [31:0]

Bits	Name	Description
31-0	DATA	

DA_00 Bits

24.4.24 DA_01 (0x104)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DA_01 [31:0]

Bits	Name	Description
31-0	DATA	

DA_01 Bits

24.4.25 DB_00 (0x140)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DB_00 [31:0]

Bits	Name	Description
31-0	DATA	

DB_00 Bits

24.4.26 DB_01 (0x144)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DB_01 [31:0]

Bits	Name	Description
31-0	DATA	

DB_01 Bits

24.4.27 DC_00 (0x180)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DC_00 [31:0]

Bits	Name	Description
31-0	DATA	

DC_00 Bits

24.4.28 DC_01 (0x184)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DC_01 [31:0]

Bits	Name	Description
31-0	DATA	

DC_01 Bits

24.4.29 DD_00 (0x1C0)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DD_00 [31:0]

Bits	Name	Description
31-0	DATA	

DD_00 Bits

24.4.30 DD_01 (0x1C4)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DD_01 [31:0]

Bits	Name	Description
31-0	DATA	

DD_01 Bits

24.4.31 DE_00 (0x200)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DE_00 [31:0]

Bits	Name	Description
31-0	DATA	

DE_00 Bits

24.4.32 DE_01 (0x204)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DE_01 [31:0]

Bits	Name	Description
31-0	DATA	

DE_01 Bits

24.4.33 DF_00 (0x240)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DF_00 [31:0]

Bits	Name	Description
31-0	DATA	

DF_00 Bits

24.4.34 DF_01 (0x244)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DF_01 [31:0]

Bits	Name	Description
31-0	DATA	

DF_01 Bits

24.4.35 DG_00 (0x280)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DG_00 [31:0]

Bits	Name	Description
31-0	DATA	

DG_00 Bits

24.4.36 DG_01 (0x284)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DG_01 [31:0]

Bits	Name	Description
31-0	DATA	

DG_01 Bits

24.4.37 DH_00 (0x2C0)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DH_00 [31:0]

Bits	Name	Description
31-0	DATA	

DH_00 Bits

24.4.38 DH_01 (0x2C4)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DH_01 [31:0]

Bits	Name	Description
31-0	DATA	

DH_01 Bits

25 Logic Observation Module LOBS

This section describes the main features and functions of LOBS.

25.1 Function Description

All PAD do/oe/di signals will be sent to lobs. When certain expected conditions are met, the value of do/oe/di of the specified PAD signal will be stored.

25.2 Feature Summary

This chapter describes the main features of LOBS:

- 12 Signal Groups with a configurable width of 128
Lobs has 12 sets of signal inputs, each set of 128 bits, and the unused high-bit tie0, so in fact, only 96 bits at most in each group are valid
 . All these signals can be used as comparison signals or signals to be collected

grp0	PA31_DI	PA31_OE	PA31_DO	...	PA00_DI	PA00_OE	PA00_DO
grp1	PB31_DI	PB31_OE	PB31_DO	...	PB00_DI	PB00_OE	PB00_DO
grp2	PC31_DI	PC31_OE	PC31_DO	...	PC00_DI	PC00_OE	PC00_DO
grp3	PD31_DI	PD31_OE	PD31_DO	...	PD00_DI	PD00_OE	PD00_DO
grp4	PE05_DI	PE05_OE	PE05_DO	...	PE00_DI	PE00_OE	PE00_DO
grp5	PF31_DI	PF31_OE	PF31_DO	...	PF00_DI	PF00_OE	PF00_DO
grp6	PV15_DI	PV15_OE	PV15_DO	...	PV00_DI	PV00_OE	PV00_DO
grp7	PW21_DI	PW21_OE	PW21_DO	...	PW00_DI	PW00_OE	PW00_DO
grp8	PX07_DI	PX07_OE	PX07_DO	...	PX00_DI	PX00_OE	PX00_DO
grp9	PY07_DI	PY07_OE	PY07_DO	...	PY00_DI	PY00_OE	PY00_DO

The py pin needs to be configured with PIOC

- Programmable Trigger States
Five programmable trigger modes that can jump back and forth to when trigger conditions are met Each mode only supports a single trigger condition and does not support multiple trigger conditions
- 32 bit counter in each Trigger States
Each trigger mode contains a 32-bit counter
- Trace Signal to system bus by AHB32
Store the value of the signal to be collected into the system memory through the ahb bus. The width of the signal to be collected is 128 bits. Only a few bits of are not supported.
can be set to the start address and stop address stored in the system bus If the amount of data to be stored exceeds the set address range, it will loop back to the original address and continue to store
.
- Internal FIFO
Built-in buffer prevents loss of acquisition signals caused by the system bus being occupied. You can read out the full status through the register or output interrupt
The FIFO input is 128 bits, and a 128-bit data has a total of 16 bytes. The data arrangement format is as follows:

data[7:0] is header information. The bit2-bit4 flags are currently in that state, and the other bits are temporarily useless data[31:8] is counting information data[127:32] is data collection information

- Data Loss

LOBS does not store the signals to be collected into system memory under each cycle, resulting in memory waste. Here, different loss intervals can be set according to different AHB bus burst configurations

When hburst=4, it is recommended to configure

every 4 pieces of data When hburst=8, it is recommended to configure

every 5 pieces of data When hburst=16, it is recommended to configure

every 6 pieces of data

When shburst=8/16, a burst is incomplete but there is no valid data. At this time, the incomplete operation

will be made up. There is a register that can read out the real end address of the sampling data storage, and the artificially completed deadbeef will not be regarded as the sampling data

25.3 Configuration Notes

This chapter describes how to configure LOBS:

- When does the trigger condition satisfy

The comparison between the value of the comparison signal and the value set in the register is successful (trigger signal comparison mode)

The count value of the counter matches the setting value (trigger count comparison mode)

- After selecting a mode, how to select the comparison signal and the signal to be collected

Select the group

where the comparison signal and sampling signal are located through the register SIGSEL For the group where the comparison signal is located It can only be selected from the group specified in the current state, that is, it can be selected through the register SIGSEL. At most 4bit

For the group where the sampling signal is located,

If the SEL signal of STREAMCTRL is set to 0, it means that the sampling signal is also selected from the group specified in the current state, that is, it can be selected through the register SIGSEL. At most 128bit

If the SEL signal of STREAMCTRL is set to 1, it means that the sampling signal is not selected from the currently specified group. At this time, the sampling signal is selected through GRPSELA and GRPENAL. Up to 8bit

These 8bits can be generated from 2 different groups, and each group can select up to 4bit

You can also choose from the same signal group, and you can choose at most 8bit

- When can the signal to be collected be stored in system memory

The signals to be collected before/after/before + after meeting the trigger conditions can be stored in the system memory

LOBS is as follows:

LOBS base address: 0xF00DC000

Offset	Register Name	Description	Reset Value
0x0000	CTRL	Control Register	0x00000000
0x0004	IRQ_EN	Interrupt Enable Register	0x00000000

Offset	Register Name	Description	Reset Value
0x0008	IRQ_STS	Interrupt Status Register	0x00000000
0x000C	STREAMCTRL	Stream Control Register	0x00000C27
0x0010	PTACTION	Pre-trigger Action Register	0x00000000
0x0014	STARTADDR	Start Address Register	0x00000000
0x0018	ENDADDR	End Address Register	0x00000000
0x0020	CTSR	Current Trigger State Register	0x00000001
0x0024	CCVR	Current Counter Value Register	0x00000000
0x0028	CAVR	Current Action Value Register	0x00000000
0x0030	FIFOSTATE	Fifo State Register	0x00000000
0x0034	FINALADDR	Final Address Register	0x00000000
0x0040	TRIGADDR1	Trigger Address1	0x00000000
0x0044	TRIGADDR2	Trigger Address1	0x00000000
0x0048	TRIGADDR3	Trigger Address1	0x00000000
0x004C	TRIGADDR4	Trigger Address1	0x00000000
0x0050	GRPSELA	Group Select Register	0x00000000
0x0054	GRPENA	Group Enable Register	0x00000000
0x0058	SIGSELA1	Signal Select1 Register	0x00000000
0x0060	SIGSELA2	Signal Select2 Register	0x00000000
0x0098	SIGENA	Signal Enable Register	0x00000000
0x0100	STATE[0][SIGSEL]	Signal Select Register	0x00000000
0x0104	STATE[0][TRIGCTRL]	Trigger Control Register	0x00000000
0x0108	STATE[0][NEXTSTATE]	Next State Register	0x00000000
0x010C	STATE[0][ACTION]	Action Register	0x00000000
0x0120	STATE[0][COUNTCOMP]	Counter Compare Register	0x00000000
0x0130	STATE[0][EXTMASK]	External Mask Register	0x00000000
0x0134	STATE[0][EXTCOMP]	External Compare Register	0x00000000
0x0140	STATE[0][SIGMASK]	Signal Mask Register	0x00000000
0x0144	STATE[0][COMPEN]	Compare Enable register	0x00000000
0x0180	STATE[0][SIGCOMP0]	Signal Compare Register0	0x00000000
0x0184	STATE[0][SIGCOMP1]	Signal Compare Register1	0x00000000
0x0188	STATE[0][SIGCOMP2]	Signal Compare Register2	0x00000000
0x018C	STATE[0][SIGCOMP3]	Signal Compare Register3	0x00000000
0x0200	STATE[1][SIGSEL]	Signal Select Register	0x00000000
0x0204	STATE[1][TRIGCTRL]	Trigger Control Register	0x00000000
0x0208	STATE[1][NEXTSTATE]	Next State Register	0x00000000
0x020C	STATE[1][ACTION]	Action Register	0x00000000
0x0220	STATE[1][COUNTCOMP]	Counter Compare Register	0x00000000
0x0230	STATE[1][EXTMASK]	External Mask Register	0x00000000
0x0234	STATE[1][EXTCOMP]	External Compare Register	0x00000000
0x0240	STATE[1][SIGMASK]	Signal Mask Register	0x00000000

Offset	Register Name	Description	Reset Value
0x0244	STATE[1][COMPEN]	Compare Enable register	0x00000000
0x0280	STATE[1][SIGCOMP0]	Signal Compare Register0	0x00000000
0x0284	STATE[1][SIGCOMP1]	Signal Compare Register1	0x00000000
0x0288	STATE[1][SIGCOMP2]	Signal Compare Register2	0x00000000
0x028C	STATE[1][SIGCOMP3]	Signal Compare Register3	0x00000000
0x0300	STATE[2][SIGSEL]	Signal Select Register	0x00000000
0x0304	STATE[2][TRIGCTRL]	Trigger Control Register	0x00000000
0x0308	STATE[2][NEXTSTATE]	Next State Register	0x00000000
0x030C	STATE[2][ACTION]	Action Register	0x00000000
0x0320	STATE[2][COUNTCOMP]	Counter Compare Register	0x00000000
0x0330	STATE[2][EXTMASK]	External Mask Register	0x00000000
0x0334	STATE[2][EXTCOMP]	External Compare Register	0x00000000
0x0340	STATE[2][SIGMASK]	Signal Mask Register	0x00000000
0x0344	STATE[2][COMPEN]	Compare Enable register	0x00000000
0x0380	STATE[2][SIGCOMP0]	Signal Compare Register0	0x00000000
0x0384	STATE[2][SIGCOMP1]	Signal Compare Register1	0x00000000
0x0388	STATE[2][SIGCOMP2]	Signal Compare Register2	0x00000000
0x038C	STATE[2][SIGCOMP3]	Signal Compare Register3	0x00000000
0x0400	STATE[3][SIGSEL]	Signal Select Register	0x00000000
0x0404	STATE[3][TRIGCTRL]	Trigger Control Register	0x00000000
0x0408	STATE[3][NEXTSTATE]	Next State Register	0x00000000
0x040C	STATE[3][ACTION]	Action Register	0x00000000
0x0420	STATE[3][COUNTCOMP]	Counter Compare Register	0x00000000
0x0430	STATE[3][EXTMASK]	External Mask Register	0x00000000
0x0434	STATE[3][EXTCOMP]	External Compare Register	0x00000000
0x0440	STATE[3][SIGMASK]	Signal Mask Register	0x00000000
0x0444	STATE[3][COMPEN]	Compare Enable register	0x00000000
0x0480	STATE[3][SIGCOMP0]	Signal Compare Register0	0x00000000
0x0484	STATE[3][SIGCOMP1]	Signal Compare Register1	0x00000000
0x0488	STATE[3][SIGCOMP2]	Signal Compare Register2	0x00000000
0x048C	STATE[3][SIGCOMP3]	Signal Compare Register3	0x00000000
0x0500	STATE[4][SIGSEL]	Signal Select Register	0x00000000
0x0504	STATE[4][TRIGCTRL]	Trigger Control Register	0x00000000
0x0508	STATE[4][NEXTSTATE]	Next State Register	0x00000000
0x050C	STATE[4][ACTION]	Action Register	0x00000000
0x0520	STATE[4][COUNTCOMP]	Counter Compare Register	0x00000000
0x0530	STATE[4][EXTMASK]	External Mask Register	0x00000000
0x0534	STATE[4][EXTCOMP]	External Compare Register	0x00000000
0x0540	STATE[4][SIGMASK]	Signal Mask Register	0x00000000
0x0544	STATE[4][COMPEN]	Compare Enable register	0x00000000

Offset	Register Name	Description	Reset Value
0x0580	STATE[4][SIGCOMP0]	Signal Compare Register0	0x00000000
0x0584	STATE[4][SIGCOMP1]	Signal Compare Register1	0x00000000
0x0588	STATE[4][SIGCOMP2]	Signal Compare Register2	0x00000000
0x058C	STATE[4][SIGCOMP3]	Signal Compare Register3	0x00000000
0x0FB0	LAR	Lock Access Register	0x00000000

Table 152: LOBS Register List

25.4 LOBS register description

GPIO is as follows:

25.4.1 CTRL (0x0)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD		RSVD																												RUN	
N/A		N/A																												RW	
0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0

CTRL [31:0]

Bits	Name	Description
0	RUN	Run control. 0 LOBS disabled. Register programming permitted. 1 LOBS enabled

CTRL Bits

25.4.2 IRQ_EN (0x4)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																														FULL_EN	FINAL_EN
N/A																														RW	RW
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IRQ_EN [31:0]

Bits	Name	Description
1	FULL_EN	FIFO overflow status interrupt enable
0	FINAL_EN	final status interrupt enable

Bits	Name	Description

IRQ_EN Bits

25.4.3 IRQ_STS (0x8)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																														FULL_STS	FINAL_STS
N/A																														W1C	W1C
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IRQ_STS [31:0]

Bits	Name	Description
1	FULL_STS	FIFO overflow status interrupt status
0	FINAL_STS	final status interrupt status

IRQ_STS Bits

25.4.4 STREAMCTRL (0xC)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																	SAMPLE_INCR					SEL	RSVD	SAMPLE				BURST			
N/A																	RW					RW	N/A	RW				RW			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	x	0	1	0	0	1	1	1

STREAMCTRL [31:0]

Bits	Name	Description
14-9	SAMPLE_INCR	Sample Rate high extension bits, the high 6 bits of the SAMPLE bit field, and the combined value with the SAMPLE bit field are as follows: 4 take one every 5 5 take one every 6 6 take one every 7 9 take one every 10 49 take one every 50 99 take one every 100 499 take one every 500 999 take one every 1000 others not supported
8	SEL	Signal Group Select 0 128bit from one group 1 from 2 groups, 4bit in each group
6-3	SAMPLE	Sample Rate 4 take one every 5 5 take one every 6 6 take one every 7
2-0	BURST	Burst Cfg 3b011 Incr4 3b101 Incr8 3b111 Incr16

STREAMCTRL Bits

25.4.5 PTACTION (0x10)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																												TRACE	RSVD		
N/A																												RW	N/A		
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

PTACTION [31:0]

Bits	Name	Description
3	TRACE	Enables trace

PTACTION Bits

25.4.6 STARTADDR (0x14)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

STARTADDR [31:0]

Bits	Name	Description
31-0	ADDR	Start address

STARTADDR Bits

25.4.7 ENDADDR (0x18)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENDADDR [31:0]

Bits	Name	Description
31-0	ADDR	End address

ENDADDR Bits

25.4.8 CTSR (0x20)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FINALSTATE	RSVD																										CTSR				
RO	N/A																										RO				
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

CTSR [31:0]

Bits	Name	Description
31	FINALSTATE	0 LOBS is still tracing. 1 Indicates that the LOBS has stopped advancing Trigger States and stopped trace. FINALSTATE can be set by TRIGCTRL.COUNTBRK reaching the final loop count, or by programming NEXTSTATEto zero
4-0	CTSR	Reads current Trigger State. This is a one-hot encoded field. When CTRL.RUN: 0 RAZ. 1 Returns current Trigger State. If FINALSTATE is 1, then the CTSR field gives the Trigger State when FINALSTATE became 1

CTSR Bits

25.4.9 CCVR (0x24)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCVR																															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CCVR [31:0]

Bits	Name	Description
31-0	CCVR	Returns the counter value when the CTSR was last read. If the CTSR has never been read, then the value in the CCVR is undefined

CCVR Bits

25.4.10 CAVR (0x28)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																												TRACE	RSVD		
N/A																												RO	N/A		
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

CAVR [31:0]

Bits	Name	Description
3	TRACE	Trace active. 0b0 Trace is not active. 0b1 Trace is active.

CAVR Bits

25.4.11 FIFOSTATE (0x30)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																												FULL	EMPTY		
N/A																												RO	RO		
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

FIFOSTATE [31:0]

Bits	Name	Description
1	FULL	FIFO full
0	EMPTY	FIFO empty

FIFOSTATE Bits

25.4.12 FINALADDR (0x34)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

FINALADDR [31:0]

Bits	Name	Description
31-0	ADDR	Final address

FINALADDR Bits

25.4.13 TRIGADDR1 (0x40)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																															
RW																															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TRIGADDR1 [31:0]

Bits	Name	Description
31-0	ADDR	Trigger address1

TRIGADDR1 Bits

25.4.14 TRIGADDR2 (0x44)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																ADDR															
																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TRIGADDR2 [31:0]

Bits	Name	Description
31-0	ADDR	Trigger address2

TRIGADDR2 Bits

25.4.15 TRIGADDR3 (0x48)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																ADDR															
																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TRIGADDR3 [31:0]

Bits	Name	Description
31-0	ADDR	Trigger address3

TRIGADDR3 Bits

25.4.16 TRIGADDR4 (0x4C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																ADDR															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TRIGADDR4 [31:0]

Bits	Name	Description
31-0	ADDR	Trigger address4

TRIGADDR4 Bits

25.4.17 GRPSELA (0x50)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								NUM2				NUM1			
N/A																								RW				RW			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GRPSELA [31:0]

Bits	Name	Description
7-4	NUM2	Select sample group number2
3-0	NUM1	Select sample group number1

GRPSELA Bits

25.4.18 GRPENA (0x54)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																										EN2	EN1				
N/A																										RW	RW				
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

GRPENA [31:0]

Bits	Name	Description
1	EN2	Enable sample group number2
0	EN1	Enable sample group number1

GRPENA Bits

25.4.19 SIGSELA1 (0x58)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NUM4								NUM3								NUM2								NUM1							
RW								RW								RW								RW							
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SIGSELA1 [31:0]

Bits	Name	Description
31-24	NUM4	Select sample signal bit number4 in first group
23-16	NUM3	Select sample signal bit number3 in first group
15-8	NUM2	Select sample signal bit number2 in first group
7-0	NUM1	Select sample signal bit number1 in first group

SIGSELA1 Bits

25.4.20 SIGSELA2 (0x60)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NUM4								NUM3								NUM2								NUM1							
RW								RW								RW								RW							
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SIGSELA2 [31:0]

Bits	Name	Description
31-24	NUM4	Select sample signal bit number4 in second group
23-16	NUM3	Select sample signal bit number3 in second group
15-8	NUM2	Select sample signal bit number2 in second group
7-0	NUM1	Select sample signal bit number1 in second group

SIGSELA2 Bits

25.4.21 SIGENA (0x98)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								EN2				EN1			
N/A																								RW				RW			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SIGENA [31:0]

Bits	Name	Description
7-4	EN2	Enable sample signal number2
3-0	EN1	Enable sample signal number1

SIGENA Bits

25.4.22 STATE[SIGSEL] (0x100 + 0x100 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
RSVD																				EN																
N/A																				RW																
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						

STATE[SIGSEL] [31:0]

Bits	Name	Description
11-0	EN	Selects Signal Group. 0x1 Selects Signal Group 0. 0x2 Selects Signal Group 1. 0x4 Selects Signal Group 2. 0x8 Selects Signal Group 3. 0x10 Selects Signal Group 4. 0x20 Selects Signal Group 5. 0x40 Selects Signal Group 6. 0x80 Selects Signal Group 7. 0x100 Selects Signal Group 8. 0x200 Selects Signal Group 9. 0x400 Selects Signal Group 10. 0x800 Selects Signal Group 11.

STATE[SIGSEL] Bits

25.4.23 STATE[TRIGCTRL] (0x104 + 0x100 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																						RSVD		TRACE		RSVD		COMPSEL	COMP		
N/A																						N/A		RW		N/A		RW	RW		
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x	0	0	x	x	0	0	0	0

STATE[TRIGCTRL] [31:0]

Bits	Name	Description
7-6	TRACE	Trace capture control. 0b10 Trace is captured every ELACLK cycle. others Reserved.
3	COMPSEL	Comparison mode. Acts as both a counter enable and a select for the comparison mode. 0b0 Disable counters and select Trigger Signal Comparison mode. 0b1 Enable counters and select Trigger Counter Comparison mode.
2-0	COMP	Trigger Signal Comparison type select. 0b000 Trigger Signal Comparisons disabled. The enabled counters count clocks immediately after the Trigger State has been entered and generate a programmable Output Action and transition to the next Trigger State when the Counter Compare Register count is reached, that is when a Trigger Counter Comparison match occurs. 0b001 Compare type is equal (==). 0b010 Compare type is greater than (>). 0b011 Compare type is greater than or equal (>=). 0b101 Compare type is not equal (!=). 0b110 Compare type is less than (<). 0b111 Compare type is less than or equal (<=).

STATE[TRIGCTRL] Bits

25.4.24 STATE[NEXTSTATE] (0x108 + 0x100 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																												NEXTSTATE			
N/A																												RW			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

STATE[NEXTSTATE] [31:0]

Bits	Name	Description
4-0	NEXTSTATE	Selects the next state to move to after the Trigger Condition has been met in the current state. 0x0 Do not change state. This is the final Trigger State. 0x1 Selects Trigger State 0. 0x2 Selects Trigger State 1. 0x4 Selects Trigger State 2. 0x8 Selects Trigger State 3. 0x10 Selects Trigger State 4.

STATE[NEXTSTATE] Bits

25.4.25 STATE[ACTION] (0x10C + 0x100 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																												TRACE	RSVD		
N/A																												RW	N/A		
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

STATE[ACTION] [31:0]

Bits	Name	Description
3	TRACE	Trace active. 0b0 Trace disable. 0b1 Trace enable.

STATE[ACTION] Bits

25.4.26 STATE[COUNTCOMP] (0x120 + 0x100 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

STATE[COUNTCOMP] [31:0]

Bits	Name	Description
31-0	VALUE	A value that, when reached in the associated up-counter for this Trigger State, causes a Trigger Counter Comparison match to occur.

STATE[COUNTCOMP] Bits

25.4.27 STATE[EXTMASK] (0x130 + 0x100 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

STATE[EXTMASK] [31:0]

Bits	Name	Description
31-0	VALUE	External Mask

STATE[EXTMASK] Bits

25.4.28 STATE[EXTCOMP] (0x134 + 0x100 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

STATE[EXTCOMP] [31:0]

Bits	Name	Description
31-0	VALUE	External Compare

STATE[EXTCOMP] Bits

25.4.29 STATE[SIGMASK] (0x140 + 0x100 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NUM3								NUM2								NUM1								NUM0							
RW								RW								RW								RW							
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

STATE[SIGMASK] [31:0]

Bits	Name	Description
31-24	NUM3	Select compare signal number3
23-16	NUM2	Select compare signal number2
15-8	NUM1	Select compare signal number1
7-0	NUM0	Select compare signal number0

Bits	Name	Description

STATE[SIGMASK] Bits

25.4.30 STATE[COMPEN] (0x144 + 0x100 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																										COMPLOGIC	EN				
N/A																										RW	RW				
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	

STATE[COMPEN] [31:0]

Bits	Name	Description
4	COMPLOGIC	compare logic 0 and : compare signals must match simultaneously,then trigger 1 or : compare signals only need 1 signal match,then trigger
3-0	EN	Select compare signal number0-3

STATE[COMPEN] Bits

25.4.31 STATE[SIGCOMP0] (0x180 + 0x100 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
VALUE0																																	
RW																																	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			

STATE[SIGCOMP0] [31:0]

Bits	Name	Description
31-0	VALUE0	Compare golden value for Signal Group signals[31:0].

STATE[SIGCOMP0] Bits

25.4.32 STATE[SIGCOMP1] (0x184 + 0x100 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE1																															
RW																															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

STATE[SIGCOMP1] [31:0]

Bits	Name	Description
31-0	VALUE1	Compare golden value for Signal Group signals[63:32].

STATE[SIGCOMP1] Bits

25.4.33 STATE[SIGCOMP2] (0x188 + 0x100 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																VALUE2															
																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

STATE[SIGCOMP2] [31:0]

Bits	Name	Description
31-0	VALUE2	Compare golden value for Signal Group signals[95:64].

STATE[SIGCOMP2] Bits

25.4.34 STATE[SIGCOMP3] (0x18C + 0x100 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																VALUE3															
																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

STATE[SIGCOMP3] [31:0]

Bits	Name	Description
31-0	VALUE3	Compare golden value for Signal Group signals[127:96].

STATE[SIGCOMP3] Bits

25.4.35 LAR (0xFB0)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LAR [31:0]

Bits	Name	Description
31-0	VALUE	Lock Access Value

LAR Bits

26 Memory Overview

This chapter describes the internal memory and external storage peripheral related interfaces of this product.

26.1 Internal SRAM

This product integrates a total of 544 KB of internal RAM, including the processor's local memory and general memory.

26.1.1 Local Memory

local memory LM includes instruction local memory ILM and data local memory DLM. It refers to memory that can operate at the same clock frequency as the core and can be accessed by the core with zero waiting cycles. The RISC-V CPU0 on this product is configured with 256 KB of local memory. The local memory LM can be accessed directly by the processor or by other masters on the chip through the processor's bus from the interface, which corresponds to LM and LM_SLV on the system address map.

ILD and DLM are allocated as follows:

- ILM0, instruction local memory for RISC-V CPU0, 128KB;
- DLM0, data local memory for RISC-V CPU0, 128KB;

26.1.2 Universal Memory

This product supports the following universal memory:

- AXI SRAM0, 256KB, located on the AXI bus;
- AHB SRAM, 32KB, located on the AHB bus;

26.2 General Purpose Register

This product contains a general purpose register (GPR) with a certain capacity, which allows users to store data for special purposes.

- Power management domain general registers PGPR0 and PGPR1, each with a capacity of 64 Byte, are located in the power management domain. Data can be saved when power goes down in the system's power domain. And is not affected by system power domain reset. Individual registers of may be occupied by ROM. Please refer to the ROM chapter for specific occupation.
- The PDGO module of the power management domain provides 10 32-Bit GPR registers to save data in shutdown mode.

PGPR is as follows:

PGPR0 base address: 0xF4138000

PGPR1 base address: 0xF413C000

Offset	Register Name	Description	Reset Value
0x0000	PMIC_GPR00	Generic control	0x00000000
0x0004	PMIC_GPR01	Generic control	0x00000000
0x0008	PMIC_GPR02	Generic control	0x00000000
0x000C	PMIC_GPR03	Generic control	0x00000000
0x0010	PMIC_GPR04	Generic control	0x00000000
0x0014	PMIC_GPR05	Generic control	0x00000000
0x0018	PMIC_GPR06	Generic control	0x00000000
0x001C	PMIC_GPR07	Generic control	0x00000000
0x0020	PMIC_GPR08	Generic control	0x00000000
0x0024	PMIC_GPR09	Generic control	0x00000000
0x0028	PMIC_GPR10	Generic control	0x00000000
0x002C	PMIC_GPR11	Generic control	0x00000000
0x0030	PMIC_GPR12	Generic control	0x00000000
0x0034	PMIC_GPR13	Generic control	0x00000000
0x0038	PMIC_GPR14	Generic control	0x00000000
0x003C	PMIC_GPR15	Generic control	0x00000000

Table 153: PGPR Register List

PGPR is described in detail as follows:

26.2.1 PMIC_GPR00 (0x0)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPR																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PMIC_GPR00 [31:0]

Bits	Name	Description
31-0	GPR	Generic control

PMIC_GPR00 Bits

26.2.2 PMIC_GPR01 (0x4)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPR																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PMIC_GPR01 [31:0]

Bits	Name	Description
31-0	GPR	Generic control

PMIC_GPR01 Bits

26.2.3 PMIC_GPR02 (0x8)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPR																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PMIC_GPR02 [31:0]

Bits	Name	Description
31-0	GPR	Generic control

PMIC_GPR02 Bits

26.2.4 PMIC_GPR03 (0xC)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPR																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PMIC_GPR03 [31:0]

Bits	Name	Description
31-0	GPR	Generic control

PMIC_GPR03 Bits

26.2.5 PMIC_GPR04 (0x10)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPR																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PMIC_GPR04 [31:0]

Bits	Name	Description
31-0	GPR	Generic control

PMIC_GPR04 Bits

26.2.6 PMIC_GPR05 (0x14)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPR																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PMIC_GPR05 [31:0]

Bits	Name	Description
31-0	GPR	Generic control

PMIC_GPR05 Bits

26.2.7 PMIC_GPR06 (0x18)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPR																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PMIC_GPR06 [31:0]

Bits	Name	Description
31-0	GPR	Generic control

PMIC_GPR06 Bits

26.2.8 PMIC_GPR07 (0x1C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPR																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PMIC_GPR07 [31:0]

Bits	Name	Description
31-0	GPR	Generic control

PMIC_GPR07 Bits

26.2.9 PMIC_GPR08 (0x20)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPR																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PMIC_GPR08 [31:0]

Bits	Name	Description
31-0	GPR	Generic control

PMIC_GPR08 Bits

26.2.10 PMIC_GPR09 (0x24)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPR																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PMIC_GPR09 [31:0]

Bits	Name	Description
31-0	GPR	Generic control

PMIC_GPR09 Bits

26.2.11 PMIC_GPR10 (0x28)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPR																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PMIC_GPR10 [31:0]

Bits	Name	Description
31-0	GPR	Generic control

PMIC_GPR10 Bits

26.2.12 PMIC_GPR11 (0x2C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPR																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PMIC_GPR11 [31:0]

Bits	Name	Description
31-0	GPR	Generic control

PMIC_GPR11 Bits

26.2.13 PMIC_GPR12 (0x30)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPR																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PMIC_GPR12 [31:0]

Bits	Name	Description
31-0	GPR	Generic control

PMIC_GPR12 Bits

26.2.14 PMIC_GPR13 (0x34)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPR																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PMIC_GPR13 [31:0]

Bits	Name	Description
31-0	GPR	Generic control

PMIC_GPR13 Bits

26.2.15 PMIC_GPR14 (0x38)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPR																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PMIC_GPR14 [31:0]

Bits	Name	Description
31-0	GPR	Generic control

PMIC_GPR14 Bits

26.2.16 PMIC_GPR15 (0x3C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPR																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

PMIC_GPR15 [31:0]

Bits	Name	Description
31-0	GPR	Generic control

PMIC_GPR15 Bits

26.3 Programmable parallel port PPI

programmable parallel port PPI is used to transfer data to parallel port peripherals. The CPU or system dma directly accesses the peripherals by accessing memory. The parallel port protocol can be freely defined through programming.

- supports up to 4 devices, and each device can be individually configured with address range, port bit width (8, 16, 32 bits are supported) and command sequence
- supports 4 CS ports, 4 byte address data ports, 4 byte valid ports, 8-bit control lines, and 1-bit clock line
- has a total of 64 command words. Each device can flexibly configure the required command words

26.4 Serial Bus Controller XPI

Serial Bus Controller XPI supports connecting various serially accessible external memories, and can also be interconnected with some devices that support serial bus.

XPI supports external memory including serial NOR Flash, serial NAND Flash, serial PSRAM, etc. It also supports HyperBus devices such as HyperRAM and HyperFlash.

XPI supports online code execution of XIP, and can realize online decryption of code and data through the online decryption module EXIP, that is, data and code can be stored in NOR Flash in the form of ciphertext.

26.5 Read Only Memory ROM

This product supports 128KB internal read-only memory ROM. The

ROM stores the startup code, Flashloader and some peripheral drivers of this product.

26.6 One-time Programmable Memory OTP

This product supports 4096-bit fuses as on-chip one-time programmable memory OTP.

OTP is used to store the following information:

- Product's unique serial number Unique ID
- product startup configuration information
- product security configuration information
- Part of the security key.
- Calibration data of on-chip analog modules stored when the product is shipped from the factory
- General data for users

The management of one-time programmable memory OTP is realized by the OTP controller. Users can read and write OTP through the OTP controller. In addition, the OTP controller is also responsible for implementing read protection and write protection of OTP.

On this product, users need to turn on LDOOTP when programming the OTP fuse array, and must turn off LDOOTP after programming is completed.

27 OTP and OTP Controller

This chapter introduces the main functions and features of the on-chip one-time programmable memory OTP and OTP controller.

27.1 Feature Summary

The features of the on-chip one-time programmable memory OTP are as follows:

- OTP is implemented based on a fuse array, each bit in the fuse array can be programmed once and cannot be changed after programming
- The OTP fuse array contains 128 words, each word contains 32 bits of fuses, with a total capacity of 4096 bits
- OTP words are divided into the following different types according to functions and features:
 - IDENTITY: Chip factory information, UUID and other information are classified as IDENTITY type. Such information is programmed by the original manufacturer, and the public key HASH for secure boot is also classified as IDENTITY type, which can be programmed by customers
 - SECURE: OTP words such as chip life cycle and OTP hardware lock bits are classified as SECURE type
 - SECRET: Various keys, such as Debug Key, EXIP KEK, FMK of the key manager are classified as SECRET type
 - GENERAL: General data that users can freely program is classified as GENERAL type. Control OTP words reserved for BOOT ROM are also classified as GENERAL type
- Supports 32-bit OTP hardware lock bits, each lock bit corresponds to 4 OTP words, combined with the shadow lock bits and fuse lock bits of the OTP controller to realize OTP read and program protection

27.1.1 OTP Controller Feature Summary

The management of the one-time programmable memory OTP of this product is implemented by the OTP controller. Users can realize read and program operations on OTP through the OTP controller. In addition, the OTP controller is also responsible for implementing OTP read protection and write protection.

The OTP controller is divided into 2 parts in the system address mapping:

- Main body part (named OTP in the memory mapping table)
- Shadow part (named OTPSHW in the memory mapping table)

The features of the OTP controller, as well as the different configurations of the main body part (OTP) and the shadow part (OTPSHW) are as follows:

- Supports 128 shadow registers, corresponding to 128 words of OTP
 - The main body part (OTP) contains shadow registers 0~15, corresponding to OTP words 0~15
 - The shadow part (OTPSHW) contains shadow registers 16~127, corresponding to OTP words 16~127
- Shadow lock registers: each OTP word corresponds to 2 lock bits, the shadow lock registers can configure read and write protection for shadow registers
 - The main body part (OTP) contains shadow lock register 0, corresponding to OTP words 0~15
 - The shadow part (OTPSHW) contains shadow lock registers 1~7, corresponding to OTP words

16~127

- Supports 128 fuse registers, corresponding to 128 words of OTP, the fuse register is the direct read and write interface of OTP
 - The main body part (OTP) contains fuse registers 0~127, corresponding to OTP words 0~127
 - The shadow part (OTPSHW) does not support fuse registers
- Fuse lock registers: each OTP word corresponds to 2 lock bits, the fuse lock registers can configure read and program protection for the fuse array
 - The main body part (OTP) contains fuse lock registers 0~7, corresponding to OTP words 0~127
 - The shadow part (OTPSHW) does not support fuse lock registers
- Supports OTP operation engine, supports OTP read, write, and OTP shadow register reload, supports interrupt generation
 - The operation engine of the main body part (OTP) supports OTP words 0~127
 - The shadow part (OTPSHW) does not support the operation engine

27.2 OTP Controller Function Description

27.2.1 OTP Shadow Registers

The OTP controller contains 128 shadow registers, each shadow register corresponds to one word of OTP. The OTP controller will load all data of the OTP fuse array into the shadow registers when the chip is released from reset.

Users can also manually reload specific OTP words to the corresponding shadow registers through the OTP operation engine, please refer to [Section 27.4](#) for details.

If a shadow register is not configured with read or write protection, users can read and modify the value of the shadow register arbitrarily. Note that some system configuration related control bits stored in OTP actually take effect after being loaded into the shadow register. Therefore, modifying the OTP shadow register can directly modify the corresponding configuration without programming OTP. For example, setting the `DEBUG_DISABLE` bit of the OTP shadow register to 1 can immediately close the debug port.

The OTP shadow registers include a set of shadow lock registers. The lock registers are used to configure read protection, write protection of shadow registers, and whether to allow shadow registers to reload the corresponding word value from the OTP fuse array.

Each OTP shadow register (32-bit) corresponds to 2 lock control bits. The definition of the LOCK bit field is as follows:

- 2'b00: Unlocked, no restriction on user reading and writing shadow registers
- 2'b01: Single lock, users can read shadow registers, but are not allowed to write shadow registers, and are not allowed to modify the LOCK bit itself
- 2'b10: Unlocked, no restriction on user reading and writing shadow registers, and modification of the LOCK bit itself is not allowed
- 2'b11: Double lock, users are not allowed to read or write shadow registers, and modification of the LOCK bit itself is not allowed

For shadow registers corresponding to different types of OTP words, the actual read protection, write and reload protection effects are affected by both the shadow lock register and the OTP hardware lock bit. For specific

effects, please refer to [Section 27.5](#).

27.3 OTP Fuse Registers

The OTP controller contains 128 fuse registers, each fuse register corresponds to one word of OTP. The fuse register is essentially the read and write interface of the OTP fuse array. Reading the fuse register can directly read the value of the corresponding word of the OTP fuse array bypassing the OTP shadow register.

To prevent user misoperation, the fuse register contains an unlock register UNLOCK. Writing 0x4F50454E (corresponding to the ASCII string "OPEN") to the unlock register can unlock the write protection of the fuse register. After unlocking, writing to the fuse register can directly start programming the corresponding OTP word. If the user writes any value other than 0x4F50454E to the unlock register, the write protection of the fuse register will be activated directly.

The OTP fuse registers include a set of fuse lock registers. The lock registers can configure read protection and program protection for the corresponding words of the fuse array.

Each OTP word (32-bit) corresponds to 2 lock control bits. The definition of the LOCK bit field is as follows:

- 2'b00: Unlocked, no restriction on user reading and writing fuse registers
- 2'b01: Single lock, users can read fuses, but are not allowed to program fuses, and are not allowed to modify the LOCK bit itself
- 2'b10: Unlocked, no restriction on user reading and writing fuse registers, and modification of the LOCK bit itself is not allowed
- 2'b11: Double lock, users are not allowed to read or program fuses, and modification of the LOCK bit itself is not allowed

For different types of words in the OTP fuse array, the actual read protection and program protection effects are affected by both the fuse lock register and the OTP hardware lock bit. For specific effects, please refer to [Section 27.5](#).

27.4 OTP Operation Engine

The OTP controller supports direct reading and writing of the OTP fuse array by reading and writing fuse registers. However, if the processor directly reads and writes the fuse register, the read and write delay of the fuse array itself is long, which may block the processor from working. Therefore, the OTP controller supports a set of operation engines. Users can realize reading and programming of the OTP fuse array through the operation engine, and can also reload the OTP fuse array into the corresponding shadow registers.

The steps to read OTP through the operation engine are as follows:

- Write the sub-address of the target word to the ADDR register, that is, 0 represents Word 0, 1 represents Word 1, and so on.
- Write 0x52454144 (corresponding to the ASCII string "READ") to the CMD register
- Wait for the $INT\_FLAG[READ]$ flag bit to be set to 1. Read the value of the target word from the DATA register

The steps to program OTP through the operation engine are as follows:

- Write the sub-address of the target word to the ADDR register, that is, 0 represents Word 0, 1 represents Word 1, and so on.
- Write the programming value of the target word to the DATA register

- Write 0x424C4F57 (corresponding to the ASCII string "BLOW") to the CMD register
- Wait for the $INT\_FLAG[WRITE]$ flag bit to be set to 1. Users can reload OTP shadow registers through the operation engine, which supports 4 reload regions:
 - **REGION[LOAD_REGION0]**, reload OTP Word 0 ~ Word7 to SHAWD0 ~ SHADOW7
 - **REGION[LOAD_REGION1]**, reload OTP Word 8 ~ Word15 to SHAWD7 ~ SHADOW15
 - **REGION[LOAD_REGION2]**, reload OTP Word 16 ~ Word127 to SHAWD16 ~ SHADOW127
 - **REGION[LOAD_REGION3]**, user-defined start and end word addresses of OTP words

Users can set bits 0~3 of the **LOAD_REQ** register to start the reload of **LOAD_REGION0**~3.

Note that before starting **REGION[LOAD_REGION3]**, users need to configure the **START** bit field of the **REGION[LOAD_REGION3]** register and the **STOP** bit field of the **REGION[LOAD_REGION3]** register to specify a continuous segment of OTP words. The **START** bit field defines the first OTP word address to be reloaded, and the **STOP** bit field defines the end word address of the reload, that is, reload OTP Word (START) ~ OTP Word (STOP -1) to SHAWD (START) ~ SHADOW(STOP -1).

Users can query the reload process through the **LOAD_COMP** register. When the reload of the corresponding reload region is completed, the flag bit is set to 1.

Note that the OTP operation engine's read, write and reload operations on OTP are restricted by OTP hardware lock bits, shadow register lock bits and fuse register lock bits. If the corresponding OTP word or OTP shadow register is protected by read or write protection, the read, write and reload operations of the OTP operation engine will not succeed.

27.5 OTP Read and Write Protection

The OTP controller supports configuring different read and write protection policies for different types of words in OTP. The actual read and write protection effects of the OTP fuse array, fuse registers, and shadow registers are affected by OTP hardware lock bits, shadow register lock bits and fuse register lock bits at the same time.

Word 0 of the OTP fuse array is the 32-bit hardware lock bit **HARD_LOCK**[31 : 0]. *Programming **HARD_LOCK** can apply read or write protection to the OTP fuse array.*

Note that OTP supports 32-bit hardware lock bits **HARD_LOCK**[31 : 0], each bit corresponds to 4 words of OTP. The lock bit of OTP fuse register bit shadow register.

All OTP data can be divided into the following 4 types by word. After setting the corresponding hardware lock bit for different types of OTP words, the effects of read protection or write protection are different:

- IDENTITY
- SECURE
- SECRET
- GENERAL

Users can refer to the OTP mapping table to understand the type attribution of each OTP word.

The OTP read and write protection policies are summarized as follows:

- Effects of shadow register lock bits
 - When the lock control bit of the shadow register is set to unlocked, the shadow register can be read and written freely, or the value of the OTP array can be reloaded to the shadow register
 - When the lock control bit of the shadow register is set to single lock, the shadow register is write-protected, and it is not allowed to reload the value of the OTP array to the shadow register
 - When the lock control bit of the shadow register is set to double lock, the shadow register is

read-protected and write-protected, and it is not allowed to reload the value of the OTP array to the shadow register

- Effects of fuse register lock bits
 - When the lock control bit of the fuse register is set to unlocked, the fuse register can be read and written freely
 - When the lock control bit of the fuse register is set to single lock, the fuse array is write-protected, that is, it is not allowed to program the fuse array by writing the fuse register or through the OTP operation engine
 - When the lock control bit of the fuse register is set to double lock, the fuse array is read-protected and write-protected, that is, it is not allowed to read or program the fuse array through the fuse register or the OTP operation engine
- Effects of OTP hardware lock bits
 - For IDENTITY type OTP words, when the corresponding hardware lock bit is 1, only write protection is applied, that is, it is allowed to read the OTP word through the fuse register or the OTP operation engine, and also allowed to read the shadow register corresponding to the OTP word. But it is not allowed to program the OTP word through the fuse register or the OTP operation engine, nor is it allowed to write or reload the shadow register corresponding to the OTP word.
 - For SECURE type OTP words, when the corresponding hardware lock bit is 1, only write protection is applied, that is, it is allowed to read the OTP word through the fuse register or the OTP operation engine, and also allowed to read the shadow register corresponding to the OTP word. But it is not allowed to program the OTP word through the fuse register or the OTP operation engine, nor is it allowed to write or reload the shadow register corresponding to the OTP word. Note that the shadow register corresponding to the product life cycle in the SECURE type OTP words is an exception, allowing users to write the shadow register to advance the current life cycle (set the product life cycle bit from 0 to 1), but not allowing life cycle rollback (clear the product life cycle bit from 1 to 0)
 - For SECRET type OTP words, when the corresponding hardware lock bit is 1, read protection and write protection are applied at the same time, that is, it is not allowed to read the OTP word through the fuse register or the OTP operation engine, nor is it allowed to read the shadow register corresponding to the OTP word. It is also not allowed to program the OTP word through the fuse register or the OTP operation engine, nor is it allowed to write or reload the shadow register corresponding to the OTP word.
 - For GENERAL type OTP words, when the corresponding hardware lock bit is 1, only write protection is applied to the fuse array, that is, it is allowed to read the OTP word through the fuse register or the OTP operation engine, and also allowed to read the shadow register corresponding to the OTP word. It is not allowed to program the OTP word through the fuse register or the OTP operation engine, but it is allowed to write or reload the shadow register corresponding to the OTP word.
- For a word in the OTP array, if the read protection or program protection configured by the hardware lock bit and the fuse lock bit conflicts, the protection always takes effect. Similarly, for an OTP shadow register, if the read protection or write protection configured by the hardware lock bit and the shadow lock bit conflicts, the protection always takes effect.

The following table summarizes the read and write protection effects of different types of OTP words:

IDENTITY Type	Hardware Lock	Fuse Single Lock	Fuse Double Lock	Shadow Single Lock	Shadow Double Lock
Read Fuse	Y	Y	N	Y	Y
Write Fuse	N	N	N	Y	Y
Shadow Register Reload	Y	Y	Y	N	N
Read Shadow Register	Y	Y	Y	Y	N
Write Shadow Register	N	Y	Y	N	N

Table 154: Summary of Read and Write Protection for IDENTITY Type OTP Words

SECURE Type	Hardware Lock	Fuse Single Lock	Fuse Double Lock	Shadow Single Lock	Shadow Double Lock
Read Fuse	Y	Y	N	Y	Y
Write Fuse	N	N	N	Y	Y
Shadow Register Reload	Y	Y	Y	N	N
Read Shadow Register	Y	Y	Y	Y	N
Write Shadow Register	N	Y	Y	N	N

Table 155: Summary of Read and Write Protection for SECURE Type OTP Words

SECRET Type	Hardware Lock	Fuse Single Lock	Fuse Double Lock	Shadow Single Lock	Shadow Double Lock
Read Fuse	N	Y	N	Y	Y
Write Fuse	N	N	N	Y	Y
Shadow Register Reload	Y	Y	Y	N	N
Read Shadow Register	N	Y	Y	Y	N
Write Shadow Register	N	Y	Y	N	N

Table 156: Summary of Read and Write Protection for SECRET Type OTP Words

GENERAL Type	Hardware Lock	Fuse Single Lock	Fuse Double Lock	Shadow Single Lock	Shadow Double Lock
Read Fuse	Y	Y	N	Y	Y
Write Fuse	N	N	N	Y	Y
Shadow Register Reload	Y	Y	Y	N	N
Read Shadow Register	Y	Y	Y	Y	N
Write Shadow Register	Y	Y	Y	N	N

Table 157: Summary of Read and Write Protection for GENERAL Type OTP Words

27.6 OTP Controller Registers

The OTP register list is as follows:

OTP base address: 0xF3158000

Offset	Register Name	Description	Reset Value
0x0000	SHADOW[SHADOW000]	Fuse shadow registers	0x00000000
0x0004	SHADOW[SHADOW001]	Fuse shadow registers	0x00000000
0x0008	SHADOW[SHADOW002]	Fuse shadow registers	0x00000000
0x000C	SHADOW[SHADOW003]	Fuse shadow registers	0x00000000
0x0010	SHADOW[SHADOW004]	Fuse shadow registers	0x00000000
0x0014	SHADOW[SHADOW005]	Fuse shadow registers	0x00000000
0x0018	SHADOW[SHADOW006]	Fuse shadow registers	0x00000000
0x001C	SHADOW[SHADOW007]	Fuse shadow registers	0x00000000
0x0020	SHADOW[SHADOW008]	Fuse shadow registers	0x00000000
0x0024	SHADOW[SHADOW009]	Fuse shadow registers	0x00000000
0x0028	SHADOW[SHADOW010]	Fuse shadow registers	0x00000000
0x002C	SHADOW[SHADOW011]	Fuse shadow registers	0x00000000
0x0030	SHADOW[SHADOW012]	Fuse shadow registers	0x00000000
0x0034	SHADOW[SHADOW013]	Fuse shadow registers	0x00000000
0x0038	SHADOW[SHADOW014]	Fuse shadow registers	0x00000000
0x003C	SHADOW[SHADOW015]	Fuse shadow registers	0x00000000
0x0040	SHADOW[SHADOW016]	Fuse shadow registers	0x00000000
0x0044	SHADOW[SHADOW017]	Fuse shadow registers	0x00000000
0x0048	SHADOW[SHADOW018]	Fuse shadow registers	0x00000000
0x004C	SHADOW[SHADOW019]	Fuse shadow registers	0x00000000
0x0050	SHADOW[SHADOW020]	Fuse shadow registers	0x00000000
0x0054	SHADOW[SHADOW021]	Fuse shadow registers	0x00000000
0x0058	SHADOW[SHADOW022]	Fuse shadow registers	0x00000000

Offset	Register Name	Description	Reset Value
0x005C	SHADOW[SHADOW023]	Fuse shadow registers	0x00000000
0x0060	SHADOW[SHADOW024]	Fuse shadow registers	0x00000000
0x0064	SHADOW[SHADOW025]	Fuse shadow registers	0x00000000
0x0068	SHADOW[SHADOW026]	Fuse shadow registers	0x00000000
0x006C	SHADOW[SHADOW027]	Fuse shadow registers	0x00000000
0x0070	SHADOW[SHADOW028]	Fuse shadow registers	0x00000000
0x0074	SHADOW[SHADOW029]	Fuse shadow registers	0x00000000
0x0078	SHADOW[SHADOW030]	Fuse shadow registers	0x00000000
0x007C	SHADOW[SHADOW031]	Fuse shadow registers	0x00000000
0x0080	SHADOW[SHADOW032]	Fuse shadow registers	0x00000000
0x0084	SHADOW[SHADOW033]	Fuse shadow registers	0x00000000
0x0088	SHADOW[SHADOW034]	Fuse shadow registers	0x00000000
0x008C	SHADOW[SHADOW035]	Fuse shadow registers	0x00000000
0x0090	SHADOW[SHADOW036]	Fuse shadow registers	0x00000000
0x0094	SHADOW[SHADOW037]	Fuse shadow registers	0x00000000
0x0098	SHADOW[SHADOW038]	Fuse shadow registers	0x00000000
0x009C	SHADOW[SHADOW039]	Fuse shadow registers	0x00000000
0x00A0	SHADOW[SHADOW040]	Fuse shadow registers	0x00000000
0x00A4	SHADOW[SHADOW041]	Fuse shadow registers	0x00000000
0x00A8	SHADOW[SHADOW042]	Fuse shadow registers	0x00000000
0x00AC	SHADOW[SHADOW043]	Fuse shadow registers	0x00000000
0x00B0	SHADOW[SHADOW044]	Fuse shadow registers	0x00000000
0x00B4	SHADOW[SHADOW045]	Fuse shadow registers	0x00000000
0x00B8	SHADOW[SHADOW046]	Fuse shadow registers	0x00000000
0x00BC	SHADOW[SHADOW047]	Fuse shadow registers	0x00000000
0x00C0	SHADOW[SHADOW048]	Fuse shadow registers	0x00000000
0x00C4	SHADOW[SHADOW049]	Fuse shadow registers	0x00000000
0x00C8	SHADOW[SHADOW050]	Fuse shadow registers	0x00000000
0x00CC	SHADOW[SHADOW051]	Fuse shadow registers	0x00000000
0x00D0	SHADOW[SHADOW052]	Fuse shadow registers	0x00000000
0x00D4	SHADOW[SHADOW053]	Fuse shadow registers	0x00000000
0x00D8	SHADOW[SHADOW054]	Fuse shadow registers	0x00000000
0x00DC	SHADOW[SHADOW055]	Fuse shadow registers	0x00000000
0x00E0	SHADOW[SHADOW056]	Fuse shadow registers	0x00000000
0x00E4	SHADOW[SHADOW057]	Fuse shadow registers	0x00000000
0x00E8	SHADOW[SHADOW058]	Fuse shadow registers	0x00000000
0x00EC	SHADOW[SHADOW059]	Fuse shadow registers	0x00000000
0x00F0	SHADOW[SHADOW060]	Fuse shadow registers	0x00000000
0x00F4	SHADOW[SHADOW061]	Fuse shadow registers	0x00000000
0x00F8	SHADOW[SHADOW062]	Fuse shadow registers	0x00000000

Offset	Register Name	Description	Reset Value
0x00FC	SHADOW[SHADOW063]	Fuse shadow registers	0x00000000
0x0100	SHADOW[SHADOW064]	Fuse shadow registers	0x00000000
0x0104	SHADOW[SHADOW065]	Fuse shadow registers	0x00000000
0x0108	SHADOW[SHADOW066]	Fuse shadow registers	0x00000000
0x010C	SHADOW[SHADOW067]	Fuse shadow registers	0x00000000
0x0110	SHADOW[SHADOW068]	Fuse shadow registers	0x00000000
0x0114	SHADOW[SHADOW069]	Fuse shadow registers	0x00000000
0x0118	SHADOW[SHADOW070]	Fuse shadow registers	0x00000000
0x011C	SHADOW[SHADOW071]	Fuse shadow registers	0x00000000
0x0120	SHADOW[SHADOW072]	Fuse shadow registers	0x00000000
0x0124	SHADOW[SHADOW073]	Fuse shadow registers	0x00000000
0x0128	SHADOW[SHADOW074]	Fuse shadow registers	0x00000000
0x012C	SHADOW[SHADOW075]	Fuse shadow registers	0x00000000
0x0130	SHADOW[SHADOW076]	Fuse shadow registers	0x00000000
0x0134	SHADOW[SHADOW077]	Fuse shadow registers	0x00000000
0x0138	SHADOW[SHADOW078]	Fuse shadow registers	0x00000000
0x013C	SHADOW[SHADOW079]	Fuse shadow registers	0x00000000
0x0140	SHADOW[SHADOW080]	Fuse shadow registers	0x00000000
0x0144	SHADOW[SHADOW081]	Fuse shadow registers	0x00000000
0x0148	SHADOW[SHADOW082]	Fuse shadow registers	0x00000000
0x014C	SHADOW[SHADOW083]	Fuse shadow registers	0x00000000
0x0150	SHADOW[SHADOW084]	Fuse shadow registers	0x00000000
0x0154	SHADOW[SHADOW085]	Fuse shadow registers	0x00000000
0x0158	SHADOW[SHADOW086]	Fuse shadow registers	0x00000000
0x015C	SHADOW[SHADOW087]	Fuse shadow registers	0x00000000
0x0160	SHADOW[SHADOW088]	Fuse shadow registers	0x00000000
0x0164	SHADOW[SHADOW089]	Fuse shadow registers	0x00000000
0x0168	SHADOW[SHADOW090]	Fuse shadow registers	0x00000000
0x016C	SHADOW[SHADOW091]	Fuse shadow registers	0x00000000
0x0170	SHADOW[SHADOW092]	Fuse shadow registers	0x00000000
0x0174	SHADOW[SHADOW093]	Fuse shadow registers	0x00000000
0x0178	SHADOW[SHADOW094]	Fuse shadow registers	0x00000000
0x017C	SHADOW[SHADOW095]	Fuse shadow registers	0x00000000
0x0180	SHADOW[SHADOW096]	Fuse shadow registers	0x00000000
0x0184	SHADOW[SHADOW097]	Fuse shadow registers	0x00000000
0x0188	SHADOW[SHADOW098]	Fuse shadow registers	0x00000000
0x018C	SHADOW[SHADOW099]	Fuse shadow registers	0x00000000
0x0190	SHADOW[SHADOW100]	Fuse shadow registers	0x00000000
0x0194	SHADOW[SHADOW101]	Fuse shadow registers	0x00000000
0x0198	SHADOW[SHADOW102]	Fuse shadow registers	0x00000000

Offset	Register Name	Description	Reset Value
0x019C	SHADOW[SHADOW103]	Fuse shadow registers	0x00000000
0x01A0	SHADOW[SHADOW104]	Fuse shadow registers	0x00000000
0x01A4	SHADOW[SHADOW105]	Fuse shadow registers	0x00000000
0x01A8	SHADOW[SHADOW106]	Fuse shadow registers	0x00000000
0x01AC	SHADOW[SHADOW107]	Fuse shadow registers	0x00000000
0x01B0	SHADOW[SHADOW108]	Fuse shadow registers	0x00000000
0x01B4	SHADOW[SHADOW109]	Fuse shadow registers	0x00000000
0x01B8	SHADOW[SHADOW110]	Fuse shadow registers	0x00000000
0x01BC	SHADOW[SHADOW111]	Fuse shadow registers	0x00000000
0x01C0	SHADOW[SHADOW112]	Fuse shadow registers	0x00000000
0x01C4	SHADOW[SHADOW113]	Fuse shadow registers	0x00000000
0x01C8	SHADOW[SHADOW114]	Fuse shadow registers	0x00000000
0x01CC	SHADOW[SHADOW115]	Fuse shadow registers	0x00000000
0x01D0	SHADOW[SHADOW116]	Fuse shadow registers	0x00000000
0x01D4	SHADOW[SHADOW117]	Fuse shadow registers	0x00000000
0x01D8	SHADOW[SHADOW118]	Fuse shadow registers	0x00000000
0x01DC	SHADOW[SHADOW119]	Fuse shadow registers	0x00000000
0x01E0	SHADOW[SHADOW120]	Fuse shadow registers	0x00000000
0x01E4	SHADOW[SHADOW121]	Fuse shadow registers	0x00000000
0x01E8	SHADOW[SHADOW122]	Fuse shadow registers	0x00000000
0x01EC	SHADOW[SHADOW123]	Fuse shadow registers	0x00000000
0x01F0	SHADOW[SHADOW124]	Fuse shadow registers	0x00000000
0x01F4	SHADOW[SHADOW125]	Fuse shadow registers	0x00000000
0x01F8	SHADOW[SHADOW126]	Fuse shadow registers	0x00000000
0x01FC	SHADOW[SHADOW127]	Fuse shadow registers	0x00000000
0x0200	SHADOW_LOCK[LOCK00]	Fuse shadow lock	0x00000000
0x0204	SHADOW_LOCK[LOCK01]	Fuse shadow lock	0x00000000
0x0208	SHADOW_LOCK[LOCK02]	Fuse shadow lock	0x00000000
0x020C	SHADOW_LOCK[LOCK03]	Fuse shadow lock	0x00000000
0x0210	SHADOW_LOCK[LOCK04]	Fuse shadow lock	0x00000000
0x0214	SHADOW_LOCK[LOCK05]	Fuse shadow lock	0x00000000
0x0218	SHADOW_LOCK[LOCK06]	Fuse shadow lock	0x00000000
0x021C	SHADOW_LOCK[LOCK07]	Fuse shadow lock	0x00000000
0x0400	FUSE[FUSE000]	Fuse Array	0x00000000
0x0404	FUSE[FUSE001]	Fuse Array	0x00000000
0x0408	FUSE[FUSE002]	Fuse Array	0x00000000
0x040C	FUSE[FUSE003]	Fuse Array	0x00000000
0x0410	FUSE[FUSE004]	Fuse Array	0x00000000
0x0414	FUSE[FUSE005]	Fuse Array	0x00000000
0x0418	FUSE[FUSE006]	Fuse Array	0x00000000

Offset	Register Name	Description	Reset Value
0x041C	FUSE[FUSE007]	Fuse Array	0x00000000
0x0420	FUSE[FUSE008]	Fuse Array	0x00000000
0x0424	FUSE[FUSE009]	Fuse Array	0x00000000
0x0428	FUSE[FUSE010]	Fuse Array	0x00000000
0x042C	FUSE[FUSE011]	Fuse Array	0x00000000
0x0430	FUSE[FUSE012]	Fuse Array	0x00000000
0x0434	FUSE[FUSE013]	Fuse Array	0x00000000
0x0438	FUSE[FUSE014]	Fuse Array	0x00000000
0x043C	FUSE[FUSE015]	Fuse Array	0x00000000
0x0440	FUSE[FUSE016]	Fuse Array	0x00000000
0x0444	FUSE[FUSE017]	Fuse Array	0x00000000
0x0448	FUSE[FUSE018]	Fuse Array	0x00000000
0x044C	FUSE[FUSE019]	Fuse Array	0x00000000
0x0450	FUSE[FUSE020]	Fuse Array	0x00000000
0x0454	FUSE[FUSE021]	Fuse Array	0x00000000
0x0458	FUSE[FUSE022]	Fuse Array	0x00000000
0x045C	FUSE[FUSE023]	Fuse Array	0x00000000
0x0460	FUSE[FUSE024]	Fuse Array	0x00000000
0x0464	FUSE[FUSE025]	Fuse Array	0x00000000
0x0468	FUSE[FUSE026]	Fuse Array	0x00000000
0x046C	FUSE[FUSE027]	Fuse Array	0x00000000
0x0470	FUSE[FUSE028]	Fuse Array	0x00000000
0x0474	FUSE[FUSE029]	Fuse Array	0x00000000
0x0478	FUSE[FUSE030]	Fuse Array	0x00000000
0x047C	FUSE[FUSE031]	Fuse Array	0x00000000
0x0480	FUSE[FUSE032]	Fuse Array	0x00000000
0x0484	FUSE[FUSE033]	Fuse Array	0x00000000
0x0488	FUSE[FUSE034]	Fuse Array	0x00000000
0x048C	FUSE[FUSE035]	Fuse Array	0x00000000
0x0490	FUSE[FUSE036]	Fuse Array	0x00000000
0x0494	FUSE[FUSE037]	Fuse Array	0x00000000
0x0498	FUSE[FUSE038]	Fuse Array	0x00000000
0x049C	FUSE[FUSE039]	Fuse Array	0x00000000
0x04A0	FUSE[FUSE040]	Fuse Array	0x00000000
0x04A4	FUSE[FUSE041]	Fuse Array	0x00000000
0x04A8	FUSE[FUSE042]	Fuse Array	0x00000000
0x04AC	FUSE[FUSE043]	Fuse Array	0x00000000
0x04B0	FUSE[FUSE044]	Fuse Array	0x00000000
0x04B4	FUSE[FUSE045]	Fuse Array	0x00000000
0x04B8	FUSE[FUSE046]	Fuse Array	0x00000000

Offset	Register Name	Description	Reset Value
0x04BC	FUSE[FUSE047]	Fuse Array	0x00000000
0x04C0	FUSE[FUSE048]	Fuse Array	0x00000000
0x04C4	FUSE[FUSE049]	Fuse Array	0x00000000
0x04C8	FUSE[FUSE050]	Fuse Array	0x00000000
0x04CC	FUSE[FUSE051]	Fuse Array	0x00000000
0x04D0	FUSE[FUSE052]	Fuse Array	0x00000000
0x04D4	FUSE[FUSE053]	Fuse Array	0x00000000
0x04D8	FUSE[FUSE054]	Fuse Array	0x00000000
0x04DC	FUSE[FUSE055]	Fuse Array	0x00000000
0x04E0	FUSE[FUSE056]	Fuse Array	0x00000000
0x04E4	FUSE[FUSE057]	Fuse Array	0x00000000
0x04E8	FUSE[FUSE058]	Fuse Array	0x00000000
0x04EC	FUSE[FUSE059]	Fuse Array	0x00000000
0x04F0	FUSE[FUSE060]	Fuse Array	0x00000000
0x04F4	FUSE[FUSE061]	Fuse Array	0x00000000
0x04F8	FUSE[FUSE062]	Fuse Array	0x00000000
0x04FC	FUSE[FUSE063]	Fuse Array	0x00000000
0x0500	FUSE[FUSE064]	Fuse Array	0x00000000
0x0504	FUSE[FUSE065]	Fuse Array	0x00000000
0x0508	FUSE[FUSE066]	Fuse Array	0x00000000
0x050C	FUSE[FUSE067]	Fuse Array	0x00000000
0x0510	FUSE[FUSE068]	Fuse Array	0x00000000
0x0514	FUSE[FUSE069]	Fuse Array	0x00000000
0x0518	FUSE[FUSE070]	Fuse Array	0x00000000
0x051C	FUSE[FUSE071]	Fuse Array	0x00000000
0x0520	FUSE[FUSE072]	Fuse Array	0x00000000
0x0524	FUSE[FUSE073]	Fuse Array	0x00000000
0x0528	FUSE[FUSE074]	Fuse Array	0x00000000
0x052C	FUSE[FUSE075]	Fuse Array	0x00000000
0x0530	FUSE[FUSE076]	Fuse Array	0x00000000
0x0534	FUSE[FUSE077]	Fuse Array	0x00000000
0x0538	FUSE[FUSE078]	Fuse Array	0x00000000
0x053C	FUSE[FUSE079]	Fuse Array	0x00000000
0x0540	FUSE[FUSE080]	Fuse Array	0x00000000
0x0544	FUSE[FUSE081]	Fuse Array	0x00000000
0x0548	FUSE[FUSE082]	Fuse Array	0x00000000
0x054C	FUSE[FUSE083]	Fuse Array	0x00000000
0x0550	FUSE[FUSE084]	Fuse Array	0x00000000
0x0554	FUSE[FUSE085]	Fuse Array	0x00000000
0x0558	FUSE[FUSE086]	Fuse Array	0x00000000

Offset	Register Name	Description	Reset Value
0x055C	FUSE[FUSE087]	Fuse Array	0x00000000
0x0560	FUSE[FUSE088]	Fuse Array	0x00000000
0x0564	FUSE[FUSE089]	Fuse Array	0x00000000
0x0568	FUSE[FUSE090]	Fuse Array	0x00000000
0x056C	FUSE[FUSE091]	Fuse Array	0x00000000
0x0570	FUSE[FUSE092]	Fuse Array	0x00000000
0x0574	FUSE[FUSE093]	Fuse Array	0x00000000
0x0578	FUSE[FUSE094]	Fuse Array	0x00000000
0x057C	FUSE[FUSE095]	Fuse Array	0x00000000
0x0580	FUSE[FUSE096]	Fuse Array	0x00000000
0x0584	FUSE[FUSE097]	Fuse Array	0x00000000
0x0588	FUSE[FUSE098]	Fuse Array	0x00000000
0x058C	FUSE[FUSE099]	Fuse Array	0x00000000
0x0590	FUSE[FUSE100]	Fuse Array	0x00000000
0x0594	FUSE[FUSE101]	Fuse Array	0x00000000
0x0598	FUSE[FUSE102]	Fuse Array	0x00000000
0x059C	FUSE[FUSE103]	Fuse Array	0x00000000
0x05A0	FUSE[FUSE104]	Fuse Array	0x00000000
0x05A4	FUSE[FUSE105]	Fuse Array	0x00000000
0x05A8	FUSE[FUSE106]	Fuse Array	0x00000000
0x05AC	FUSE[FUSE107]	Fuse Array	0x00000000
0x05B0	FUSE[FUSE108]	Fuse Array	0x00000000
0x05B4	FUSE[FUSE109]	Fuse Array	0x00000000
0x05B8	FUSE[FUSE110]	Fuse Array	0x00000000
0x05BC	FUSE[FUSE111]	Fuse Array	0x00000000
0x05C0	FUSE[FUSE112]	Fuse Array	0x00000000
0x05C4	FUSE[FUSE113]	Fuse Array	0x00000000
0x05C8	FUSE[FUSE114]	Fuse Array	0x00000000
0x05CC	FUSE[FUSE115]	Fuse Array	0x00000000
0x05D0	FUSE[FUSE116]	Fuse Array	0x00000000
0x05D4	FUSE[FUSE117]	Fuse Array	0x00000000
0x05D8	FUSE[FUSE118]	Fuse Array	0x00000000
0x05DC	FUSE[FUSE119]	Fuse Array	0x00000000
0x05E0	FUSE[FUSE120]	Fuse Array	0x00000000
0x05E4	FUSE[FUSE121]	Fuse Array	0x00000000
0x05E8	FUSE[FUSE122]	Fuse Array	0x00000000
0x05EC	FUSE[FUSE123]	Fuse Array	0x00000000
0x05F0	FUSE[FUSE124]	Fuse Array	0x00000000
0x05F4	FUSE[FUSE125]	Fuse Array	0x00000000
0x05F8	FUSE[FUSE126]	Fuse Array	0x00000000

Offset	Register Name	Description	Reset Value
0x05FC	FUSE[FUSE127]	Fuse Array	0x00000000
0x0600	FUSE_LOCK[LOCK00]	Fuse lock	0x00000000
0x0604	FUSE_LOCK[LOCK01]	Fuse lock	0x00000000
0x0608	FUSE_LOCK[LOCK02]	Fuse lock	0x00000000
0x060C	FUSE_LOCK[LOCK03]	Fuse lock	0x00000000
0x0610	FUSE_LOCK[LOCK04]	Fuse lock	0x00000000
0x0614	FUSE_LOCK[LOCK05]	Fuse lock	0x00000000
0x0618	FUSE_LOCK[LOCK06]	Fuse lock	0x00000000
0x061C	FUSE_LOCK[LOCK07]	Fuse lock	0x00000000
0x0800	UNLOCK	UNLOCK	0x00000000
0x0804	DATA	DATA	0x00000000
0x0808	ADDR	ADDR	0x00000000
0x080C	CMD	CMD	0x00000000
0x0A00	LOAD_REQ	LOAD Request	0x00000007
0x0A04	LOAD_COMP	LOAD complete	0x00000007
0x0A20	REGION[LOAD_REGION0]	LOAD region	0x00000800
0x0A24	REGION[LOAD_REGION1]	LOAD region	0x00001008
0x0A28	REGION[LOAD_REGION2]	LOAD region	0x00000010
0x0A2C	REGION[LOAD_REGION3]	LOAD region	0x00000000
0x0C00	INT_FLAG	interrupt flag	0x00000000
0x0C04	INT_EN	interrupt enable	0x00000000

Table 158: OTP Register List

27.7 OTP Controller Register Details

The detailed description of OTP registers is as follows:

27.7.1 SHADOW (0x0 + 0x4 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																SHADOW															
																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SHADOW [31:0]

Bits	Name	Description
31-0	SHADOW	shadow register of fuse for pmic area for PMIC, index valid for 0-15, for SOC index valid for 16-128

SHADOW Bits

27.7.2 SHADOW_LOCK (0x200 + 0x4 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LOCK																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SHADOW_LOCK [31:0]

Bits	Name	Description
31-0	LOCK	lock for pmic part shadow registers, 2 bits per 32 bit word, lock behavior is different between different fuse types 00: not locked 01: soft locked 10: not locked, and cannot lock in future 11: double locked

SHADOW_LOCK Bits

27.7.3 FUSE (0x400 + 0x4 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FUSE																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FUSE [31:0]

Bits	Name	Description
31-0	FUSE	fuse array, valid in PMIC part only read operation will read out value in fuse array write operation will update fuse array value(please make sure fuse is unlocked and 2.5V power is ready)

FUSE Bits

27.7.4 FUSE_LOCK (0x600 + 0x4 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LOCK																															
RW																															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FUSE_LOCK [31:0]

Bits	Name	Description
31-0	LOCK	lock for fuse array, 2 bits per 32 bit word, lock behavior is different between different fuse types 00: not locked 01: soft locked 10: not locked, and cannot lock in future 11: double locked

FUSE_LOCK Bits

27.7.5 UNLOCK (0x800)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																UNLOCK															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

UNLOCK [31:0]

Bits	Name	Description
31-0	UNLOCK	unlock word for fuse array operation write "OPEN" to unlock fuse array, write any other value will lock write to fuse. Please make sure 24M crystal is running and 2.5V LDO working properly

UNLOCK Bits

27.7.6 DATA (0x804)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																DATA															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DATA [31:0]

Bits	Name	Description
31-0	DATA	data register for non-blocking access this register hold dat read from fuse array or data to by programmed to fuse array

DATA Bits

27.7.7 ADDR (0x808)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								ADDR							
N/A																								RW							
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	

ADDR [31:0]

Bits	Name	Description
6-0	ADDR	word address to be read or write

ADDR Bits

27.7.8 CMD (0x80C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMD																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CMD [31:0]

Bits	Name	Description
31-0	CMD	command to access fure array "BLOW" will update fuse word at ADDR to value hold in DATA "READ" will fetch fuse value in at ADDR to DATA register

CMD Bits

27.7.9 LOAD_REQ (0xA00)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																REQUEST															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
N/A																												RW			
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	1	1	1

LOAD_REQ [31:0]

Bits	Name	Description
3-0	REQUEST	reload request for 4 regions bit0: region0 bit1: region1 bit2: region2 bit3: region3

LOAD_REQ Bits

27.7.10 LOAD_COMP (0xA04)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																												COMPLETE			
N/A																												RW			
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	1	1	1

LOAD_COMP [31:0]

Bits	Name	Description
3-0	COMPLETE	reload complete sign for 4 regions bit0: region 0 bit1: region1 bit2: region2 bit3: region3

LOAD_COMP Bits

27.7.11 REGION (0xA20 + 0x4 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																STOP						RSVD	START								
N/A																RW						N/A	RW								
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	x	0	0	0	0	0	0	0

REGION [31:0]

Bits	Name	Description
14-8	STOP	stop address of load region, fuse word at end address will NOT be reloaded region0: fixed at 8 region1: fixed at 16 region2: fixed at 0, region3: usrer configurable
6-0	START	start address of load region, fuse word at start address will be reloaded region0: fixed at 0 region1: fixed at 8 region2: fixed at 16, region3: usrer configurable

REGION Bits

27.7.12 INT_FLAG (0xC00)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																												WRITE	READ	LOAD	
N/A																												RW	RW	RW	
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	

INT_FLAG [31:0]

Bits	Name	Description
2	WRITE	fuse write flag, write 1 to clear 0: fuse is not written or writing 1: value in DATA register is programmed into fuse
1	READ	fuse read flag, write 1 to clear 0: fuse is not read or reading 1: fuse value is put in DATA register
0	LOAD	fuse load flag, write 1 to clear 0: fuse is not loaded or loading 1: fuse loaded

INT_FLAG Bits

27.7.13 INT_EN (0xC04)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																WRITE			READ			LOAD									

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
N/A																													RW	RW	RW
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0

INT_EN [31:0]

Bits	Name	Description
2	WRITE	fuse write interrupt enable 0: fuse write interrupt is not enable 1: fuse write interrupt is enable
1	READ	fuse read interrupt enable 0: fuse read interrupt is not enable 1: fuse read interrupt is enable
0	LOAD	fuse load interrupt enable 0: fuse load interrupt is not enable 1: fuse load interrupt is enable

INT_EN Bits

28 Expanded Serial Peripheral Interface XPI

This chapter describes the main features and functions of Expanded Serial Peripheral Interface XPI.

28.1 Feature Summary

Expanded Serial Peripheral Interface XPI supports connecting various external memory devices, and can also communicate with some serial bus devices.

XPI supports external memory devices including serial NOR Flash, serial NAND Flash, serial PSRAM, etc., and also supports HyperBus devices such as HyperRAM, HyperFlash.

The features of XPI are as follows:

- supports 1/2/4/8 bit data mode;
- Two IO ports CA and CB, each port has 2 CS. XPI can connect four external devices with 4bit data width or two external devices with 8bit data width;
- supports both SDR and DDR;
- supports direct bus addressing access to external memory devices;
- supports Quad-SPI/Octal-SPI serial NOR Flash;
- supports serial NAND Flash;
- supports HyperBus, HyperRAM and HyperFlash;
- supports Quad-SPI/Octal-SPI PSRAM.

28.2 Function Description

This chapter describes the functions of Expanded Serial Peripheral Interface XPI.

28.2.1 External device connection configuration

XPI supports connecting various external memory devices, it can be configured into: external port independent mode and external port combination mode; shortened to independent mode and combination mode.

When XPI is set to independent mode, the pins of port CA and port CB are as follows:

- port CA: CA_CS0, CA_CS1, CA_SCLK, CA_DQS, CA_D[3:0];
- port CB: CB_CS0, CB_CS1, CB_SCLK, CB_DQS, CB_D[3:0];

In independent mode, XPI can connect four external devices (such as nor flash):

- Device 1: CA_CS0, CA_SCLK, CA_DQS, CA_D[3:0];
- Device 2: CA_CS1, CA_SCLK, CA_DQS, CA_D[3:0];
- Device 3: CB_CS0, CB_SCLK, CB_DQS, CB_D[3:0];
- Device 4: CB_CS1, CB_SCLK, CB_DQS, CB_D[3:0];

When XPI is set to combination mode, it can connect two memory devices with 8bit data width, such as Octal-SPI NOR Flash or Octal-SPI PSRAM.

- Device 1: CA_CS0, CA_SCLK, CA_DQS, CA_D[3:0], CB_D[3:0];
- Device 2: CB_CS0, CB_SCLK, CB_DQS, CA_D[3:0], CB_D[3:0];

The data bits wide supported by XPI: Classic SPI mode, Dual-SPI mode, Quad-SPI mode and Octal-SPI mode.

- Classic SPI mode, D0 is MOSI(Master OUT Slave IN), and D1 is MISO(Master IN Slave OUT);
- Dual-SPI mode, D0 and D1 are used as 2-wire bidirectional data signals;
- Qual-SPI mode, D0 ~ D3 is used as a 4-line bidirectional data signal;
- Octal-SPI mode, D0 ~ D3 on port A and D0 ~ D3 on port B are used as 8-wire bidirectional data signals.

The data transfer mode of XPI can be configured to SDR(Single Data Rate) mode or DDR(Double Data Rate) mode;SDR mode is aligned with the rising edge of the clock during data transfer; DDR mode transmits data on the rising edge and falling edge of the clock. Which data transfer mode during transfer can be determined by the access instructions in XPI's INSTGRP.

28.2.2 XPI address mapping

XPI supports connecting up to 4 external serial memory devices, differentiated by address. When the access address falls within a memory device address range, the external device will be selected through the corresponding CS signal. The user configures the address range of each external device in bytes through the FLSHSZ bitfile in the CR0 register of each external device.

- CA_CS0, address range is $0 \sim DEVSIZE[0].SIZE\_KB$
- CA_CS1, address range is $DEVSIZE[0].SIZE\_KB \sim DEVSIZE[0].SIZE\_KB + DEVSIZE[1].SIZE\_KB$
- CB_CS0, address range is $DEVSIZE[0].SIZE\_KB + DEVSIZE[1].SIZE\_KB \sim DEVSIZE[0].SIZE\_KB + DEVSIZE[1].SIZE\_KB + DEVSIZE[2].SIZE\_KB$
- CB_CS1, address range is $DEVSIZE[0].SIZE\_KB + DEVSIZE[1].SIZE\_KB + DEVSIZE[2].SIZE\_KB \sim DEVSIZE[0].SIZE\_KB + DEVSIZE[1].SIZE\_KB + DEVSIZE[2].SIZE\_KB + DEVSIZE[3].SIZE\_KB$

28.2.3 External Device Address Translation

XPI supports bus addressing ranges.The master device on the bus can directly access external devices connected to XPI through the addresses mapped by the XPI system memory. At this time, XPI will automatically remove the base address of XPI on the system memory mapping table from the received system address, and only access the external device using the address offset.

XPI supports subdivision of addresses into row addresses and column addresses. When the DEVATTR[x].CAS bitfile is non-0, XPI supports sending row addresses and column addresses to external devices respectively. Otherwise all addresses are sent as row addresses.

28.2.4 instruction list

XPI supports configuring the format of instructions, addresses, and data to external memory through the instruction table. XPI supports 16 groups of instruction tables, each group of instruction table supports up to 8 instructions, each instruction being 16 bits long.

Instructions can configure a specific stage of communication between XPI and external devices, such as instruction segment, address segment, Dummy segment, data read/write segment, etc. Several instructions are combined to a complete XPI read or write operation.

The 16-bit instruction format is as follows:

- bit [15], when it is 0, data is transferred in SDR mode, and when it is 1, data is transferred in DDR mode;
- bit [14], reserved bit;
- bit [13:10], opcode ;

- bit [9:8], data bits *numpad* ;
- bit [7:0], *operand* .

Instruction list supports the following opcodes:

- 4'h1, send CMD, the sent CMD value is equal to operand;
- 4'h2, send row address, operand represents the address length, in bits;
- 4'h3, send column address, operand represents the address length, in bits;
- 4'h4, send 1-bit MODE, MODE is equal to operand[0], and the number of data bits can only be 1 bit;
- 4'h5, send 2-bit MODE, MODE is equal to operand[1:0], and the number of data bits cannot exceed 2 bits;
- 4'h6, send 4-bit MODE, MODE is equal to operand[3:0], and the number of data bits cannot exceed 4 bits;
- 4'h7, send 8-bit MODE, MODE is equal to operand[7:0];
- 4'h8, write data;
- 4'h9, read data;
- 4'hC, send Dummy cycle. When sending Dummy, XPI does not drive the data line. The number of Dummy cycles is equal to the operand;
- 4'hD, sends Dummy RWDS, similar to Dummy, for HyperBus devices. During a read operation, if RWDS is high, The number of Dummy cycles is $(operand \times 4 - 1)$, otherwise it is $(operand \times 2 - 1)$; During a write operation, if RWDS is high, the number of Dummy cycles is $(operand \times 4 - 2)$,Otherwise it is $(operand \times 2 - 2)$;
- 4'h0, stop executing instructions.

Instruction table supports the following data bit numbers *numpad*, which represents the number of data lines used by the instruction:

- 2'h0, 1-bit data line transmits data;
- 2'h1, 2-bit data line transmits data;
- 2'h2, 4-bit data line transfers data;
- 2'h3, 8-bit data line transfers data;

When XPI accesses through bus addressing, it determines which instruction in the instruction table to execute the corresponding read or write operation according to the configuration of the DEVMACR[x] register.

When XPI accesses through register instructions, the instructions specified in the instruction table are executed according to the configuration of the APBINSR register.

28.2.5 Bus addressed access

XPI supports bus addressed access. Bus addressed access means that the master device on the bus, such as a processor, DMA, etc., uses XPI as a bus slave device and directly accesses the external device connected to XPI according to the address assigned to XPI on the memory map table. XPI executes code online through bus-addressed access.

Any read or write access by the bus master on the system to the XPI address means issuing a bus addressing access request to the XPI. When XPI receives such a request, it converts the request into read or write access to the external device. Conversion is achieved by executing one or more instructions in the instruction list.

Users need to edit the instruction table in advance and use instructions to read or write external devices.

Then configure the WRINSGRPIND bitfile of register DEVMACR[x] to get the instruction index to be executed when a bus-addressed access write operation is received.

28.2.6 register instruction access

XPI supports register instruction access. Users can edit the instruction table and execute any instruction in the instruction table by configuring the registers of XPI.

After the user edits the instruction table, the user specifies the index of the instruction to be executed in the instruction table through the PINSGRPIND bitfile in the APBINSR register. Then set the TRG bitfile in the APBCMD register to 1 to trigger the execution of the instruction.

When operating external devices, the read data is stored in the register RX buffer, and software can directly read the buffer through the PRX register. When writing to external devices, the data to be sent must be written into the TX buffer in advance, and software can directly write to the TX buffer through the PTX register.

When an XPI register instruction accesses an external device, the target address is stored in the DAR bitfile in the APBDAR register.

28.2.7 XPI Time Characteristics

XPI supports adjusting the time characteristics of IO input and output.

The setup time and hold time between the CSx and SCLK can be adjusted through the TCSS bitfile and TCSH bitfile in the DEVATTR[x] register.

- SDR mode, the setup time is $TCSS + 0.5$ cycles, and the hold time is $TCSH$ cycles;
- DDR mode, the setup time is $TCSS + 0.5$ cycles, and the hold time is $TCSH + 0.5$ cycles.

XPI supports configuring the interval between the CSx signal flipping from active to inactive and the next time it flips to active again. User can configure the DEVATTR[x] register. The interval is as following.

$CSINTVAL \times 1$ clock cycles, CSINTVU = 1'b0;

$CSINTVAL \times 256$ clock cycles, CSINTVU = 1'b1.

XPI supports sampling options for setting data, and can be configured through the RXCLKSRC bit in the GCR0 register.

- 2'b00, send clock, internal loopback;
- 2'b01, looped back through the DQS pin of the chip;
- 2'b10, reserved for standby;
- 2'b11, the DQS signal provided by external device.

28.2.8 interrupt

XPI supports generating the following interrupt requests:

- register instruction access completed;
- register instruction access timeout;
- bus addressing access timeout;
- register instruction access fail;
- bus addressing access fail;
- the number of valid data in the RX buffer for register instruction access reaches or exceeds the threshold;

- the number of valid data in the TX buffer for register instruction access reaches or falls below the threshold;
- Bus timeout.

28.3 XPI Configuration and Use

It is recommended that users initialize and configure XPI by calling the ROM API of this product.

For details of the ROM API of this product, please refer to [Section 18.7](#) .

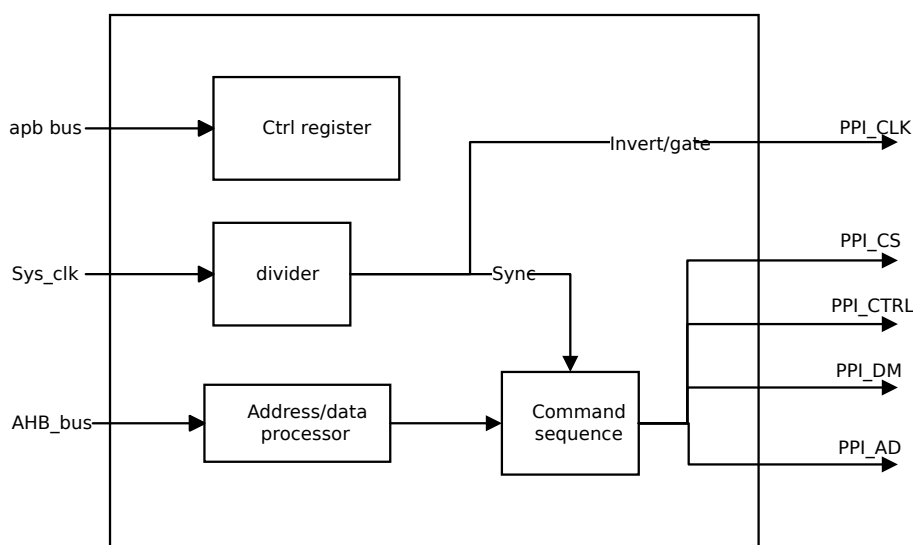
29 Programmable parallel port PPI

This chapter introduces the main functions and features of the Programmable Parallel Interface PPI(Programmable Parallel Interface).

29.1 Overview

PPI module can flexibly configure the chip's internal system bus read and write operations into parallel external interface read and write timing signals to complete data exchange with chip peripherals. PPI can be connected directly to external RAM, Flash, programmable logic device, or some other simple device that just serves as a slave.

29.2 PPI Architecture Diagram



29.3 External Interface Pin Description

Table 159: PPI pin description

Pin	Direction	Function Description
PPI_CS[3:0]	Output	Peripheral Selection
PPI_AD[31:0]	Input Output	Address or Data
PPI_DM[3:0]	Output	Valid Data Byte Select Signal
PPI_CTRL[7:0]	Input Output	Control Signal Line
PPI_CLK	Output	External Output Clock

29.4 Feature Summary

- supports 4 CS pins, and each CS can be configured separately: address interval (base address requires 1MB alignment, and a maximum address span of 128MB), data port bit width (support 8/16/32 bits) and configurable command word sequence (start/end index of the command word)
- supports 32-bit data/address multiplexing pins and supports the ability to flexibly configure these pins in bytes for transmitting address or data

- supports 4 valid data selection pins, corresponding to whether each byte in the data bus is valid or not. These pins correspond to the bytes in the data/address multiplexing pins that are configured as data functions
- supports 8 control pins, which can flexibly configure the timing relationship between each other and configure the input and output directions of pins
- supports 1 clock output pin, which is obtained by dividing the system clock. The maximum frequency is 16, and the clock is always output or output following the sequence of command words.
- 32-bit AHB data transmission port, supporting burst transmission
- 32-bit APB configuration register port

29.5 Function Description

This module is used to transfer data with parallel port peripherals. After the configuration is completed, the CPU or system dma can directly access the peripherals by accessing memory through the AHB bus. For each AHB data transfer, on the chip interface, it can be cut into several segments, each segment being described by a command word. For some devices, the first access and subsequent access will be different, such as devices that support burst transmission, so each device can be configured with a different sequence of first access or subsequent access command words.

29.5.1 PPI_AD[31:0] Address data allocation

external interface is a configurable address data multiplexing bus. Several optional configurations are provided below:

- Time-sharing multiplexing of 32-bit addresses and 32-bit data
- Time-sharing multiplexing of 32-bit addresses and 16-bit data
- Time-sharing multiplexing of 32-bit addresses and 8-bit data
- 16-bit address +16-bit data non-multiplexed
- 24-bit address +8-bit data non-multiplexed

Note: In time-sharing multiplexing mode, sometimes it is necessary to add an address latch externally to the chip. You can select PPI_CTRL to use it as the address latch enable.

29.5.2 Read and write cycle settings

PPI is driven by a command word sequence, with a maximum sequence length of 64. Each time the system bus initiates a read/write access, PPI drives the corresponding pins in turn according to the command word sequence order to generate the target timing. Read and write are described by different command sequences. The start and end sequence numbers of the read command word need to be configured in the CS[CFG3] register, and the start and end sequence numbers of the write command word need to be configured in the CS[CFG4] register. The configurable items for each command word are as follows:

- Duration (number of system clock cycles, minimum one system clock cycle, maximum 256 system clock cycles)
- Whether the PPI_CLK pin is output at this command time (valid in clk_gate mode, which is the clock after dividing the system clock)
- PPI_CS pin outputs the polarity of the level at this command time
- PPI_AD pin function (data/address, 1 byte as a group)

- PPI_AD pin direction (input/output, 1 byte in group)
- PPI_AD pin and system data/address mapping
- PPI_CTRL Control Line Related Functions

29.5.2.1 Read Cycle

Taking the peripheral hanging on the CS0 pin as an example, the CS0 [CFG0~ CFG4] register group needs to be set as needed to select the system address range corresponding to the peripheral, select the number of control lines required to control the peripheral, control line polarity during idle, read/write start and end command words, whether clock output is required, etc. In this example, three control lines are selected for the address latch function (PPI_CTRL[0], low level when idle), the slave read enable function (PPI_CTRL[1], high level when idle), and the slave write enable function (PPI_CTRL[2], high level when idle). Take configuring four command words to form a continuous read command sequence as an example, and the read sequence shown in the following figure is composed:

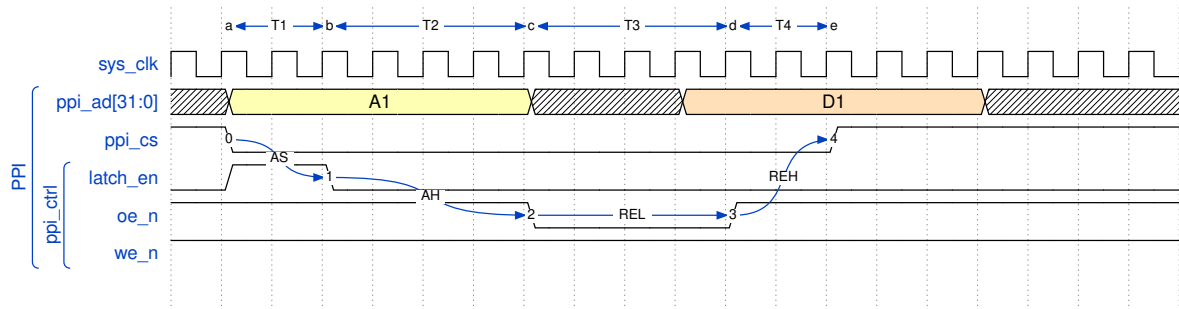
The first command word cycle (T1) is set to 2 system clock cycles in duration as the address establishment latch time (AS). During T1 time, the pins used as address bytes in PPI_AD[31:0] are configured to function as address and direction as output. At the same time, you can configure the effective polarity of PPI_CS[0] output (active low level, that is, PPI_CS[0] gets a falling edge at the beginning of the T1 cycle), and set the effective polarity of PPI_CTRL[0] output (active high level, that is, PPI_CTRL[0] gets a rising edge at the beginning of the T1 cycle). PPI_CTRL Other control signals remain output at invalid levels.

The second command word cycle (T2) is set to a duration of 4 system clock cycles as the address retention time (AH). During T2 time, keep the pin that serves as an address byte in PPI_AD[31:0] as an address and the direction is output. The active polarity of the PPI_CS[0] output is maintained while configuring the inactive polarity of the output of the control pin PPI_CTRL[0](that is, PPI_CTRL[0] gets a falling edge at the beginning of the T2 cycle). PPI_CTRL Other control signals remain output at invalid levels.

The third command word cycle (T3) is set to a duration of 4 system clock cycles as the read data time (REL). During the T3 time, the pins used as data bytes in PPI_AD[31:0] are configured to function as data and the direction is input (that is, the address data multiplexing byte in PPI_AD[31:0] is changed from output address to receive data at the beginning of the T3 cycle). Keep the valid polarity of the PPI_CS[0] output and keep the invalid polarity of the address latch control pin PPI_CTRL[0] output. At the same time, set the effective polarity of the output from the slave device read enable control pin PPI_CTRL[1](active low level, that is, PPI_CTRL[1] gets a falling edge at the beginning of the T3 cycle). Keep other control signal output invalid levels.

The fourth command word cycle (T4) is set to 2 system clock cycles in duration as the read sequence end time (REH). During T4 time, keep the pin function as data bytes in PPI_AD[31:0] as data and the direction is input. Keep the valid polarity of the PPI_CS[0] output and keep the invalid polarity of the address latch control pin PPI_CTRL[0] output. At the same time, the invalid polarity of the output from the device read enable control pin PPI_CTRL[1] is set (that is, PPI_CTRL[1] gets a rising edge at the beginning of the T4 cycle). Keep other control signal output invalid levels. PPI collects data from PPI_AD[31:0]

on the last rising edge of the system clock in the T4 cycle

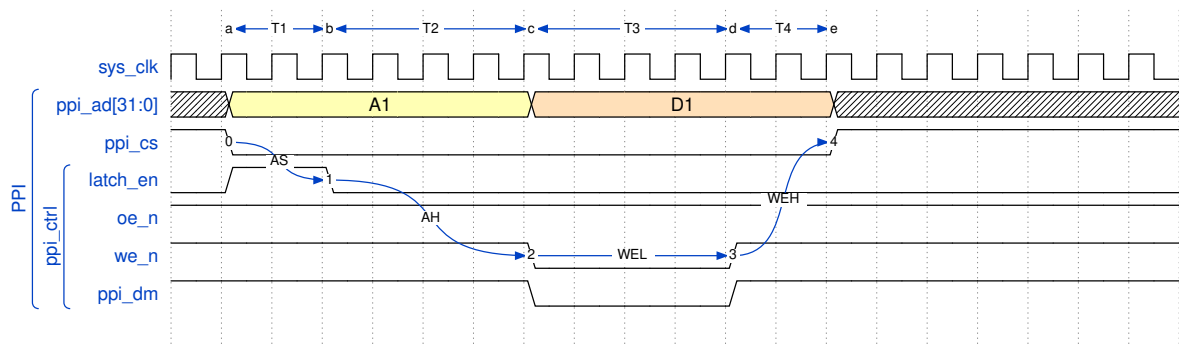


29.5.2.2 Write Cycle

The control timings of write cycles and read cycles are almost the same. The main difference is:

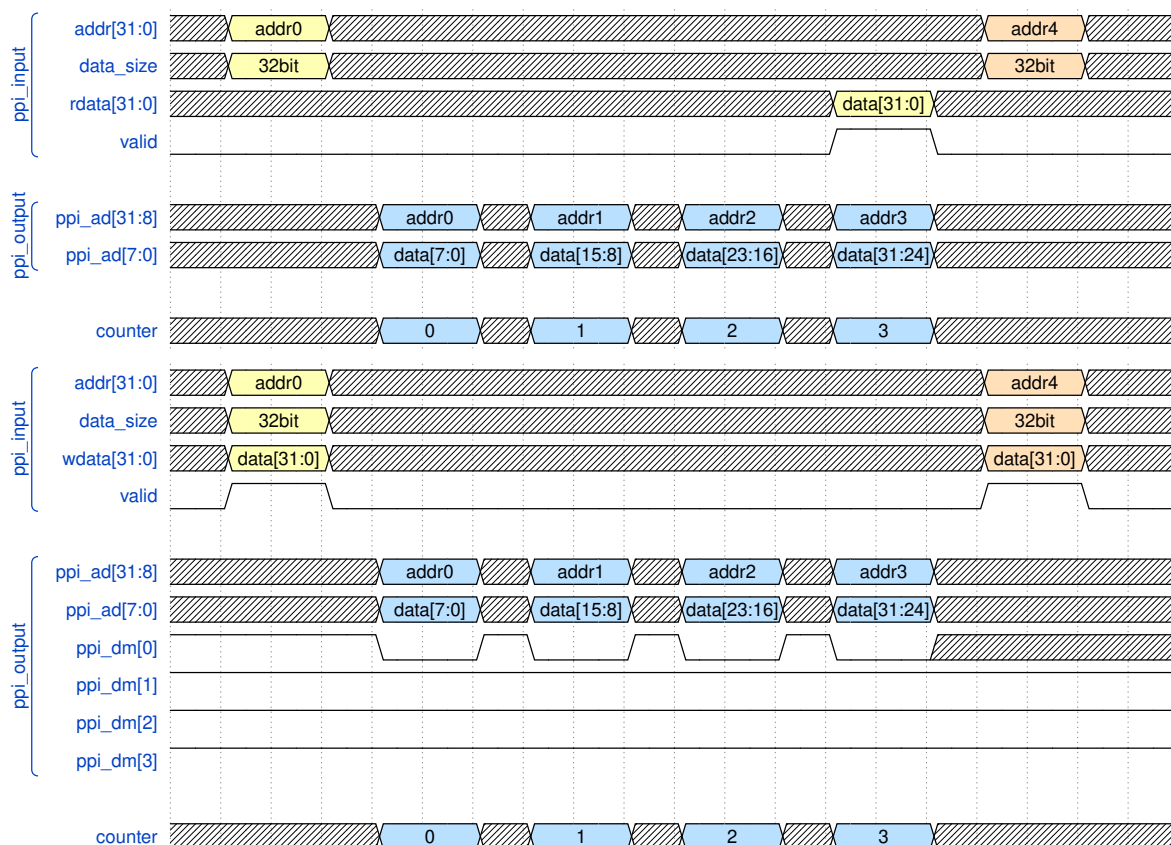
- Configure the output of an invalid level from the device read enable control pin PPI_CTRL[1], and configure the output of an effective level from the device write enable control pin PPI_CTRL[2]
- PPI_AD[31:0] The pin direction used as data bytes is output
- PPI_DM[3:0] pins output the effective polarity following the effective data byte. This pin change is determined by PPI internal logic based on the effective data byte, and does not need to follow the command word configuration.

The PPI output pin write timing relationship obtained from a sequence of four command words is shown in the following figure:



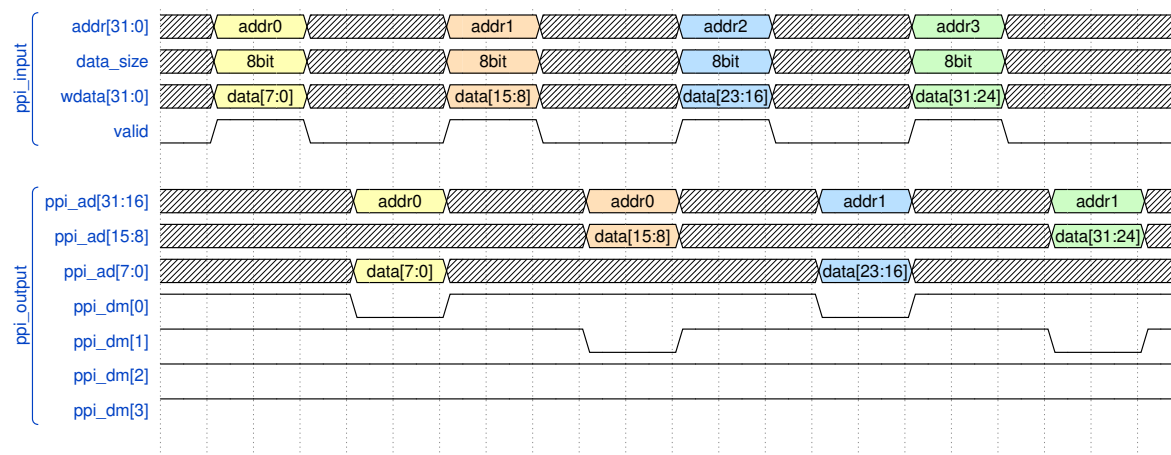
29.5.2.3 chip bus read and write bits are wider than peripheral data bits wide

For example, the chip's internal bus read and write data bit width is 32 bits, but the connected peripheral data bit width is only 8 bits. At this time, one internal bus read or write will completely execute the configured command sequence four times. The time interval between the end of the current execution command sequence and the start of the next command sequence is configurable (CS[CFG2].INTER_CMD_DLY) and defaults to one system clock cycle. Each address and data change is completed independently by PPI and no additional configuration is required. The schematic diagram of reading and writing timing split is shown in the following figure.



29.5.2.4 The read and write bit width of the chip bus is smaller than the peripheral data bit width

For example, the read and write data bit width of the internal bus of the chip is 8 bits, but the data bit width of the connected peripheral is 16 bits. At this time, an internal bus read and write will only execute the configured command sequence once. When writing data, the pin PPI_DM[3:0] that indicates whether each byte in the corresponding data bus is valid will change accordingly, as shown in the figure.



29.5.3 Clock Output

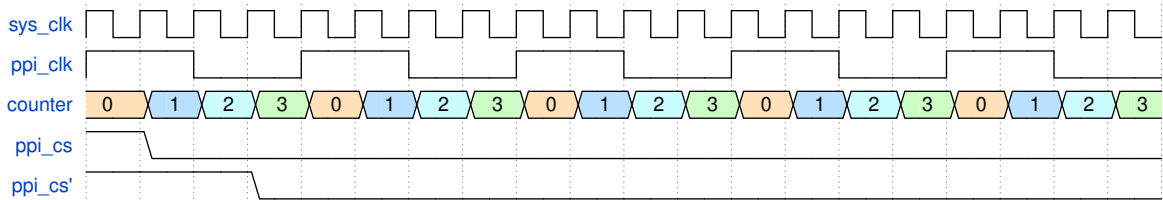
When the slave device needs an external clock, PPI_CLK can be enabled and PPI_CLK can be output either always or following the command word according to your needs. The PPI_CLK pin clock is obtained by dividing the frequency of the system clock, with a maximum of 16. The high and low level duty cycles can be configured,

and the relevant configuration is in the CLKPIN_CFG register.

also requires configuring the synchronization relationship between PPI_CLK and the command sequence. When synchronization is enabled (CS[CFG2].CS_SYNC_EN), the first command word of the command sequence will start executing when the system clock divider counter counts the synchronization configuration value (CS[CFG2].SYNC_CLK_SEL), thereby controlling the timing relationship between all PPI_CS pins, PPI_CTRL pins, and PPI_AD pins and PPI_CLK pins.

For example, if the system clock is divided by four (CLKPIN_CFG[CYCLE]=3) and output, the internal frequency division counter counts from 0 to 3 and then starts counting from 0. At this time, CLKPIN_CFG[HIGH]=3 (that is, the frequency division counter starts outputting a high level after it reaches 3), and CLKPIN_CFG[LOW]=1 (that is, the frequency division counter starts outputting a low level after it reaches 1).

If CS[CFG2].SYNC_CLK_SEL is set to 0, if the first command word is to output the PPI_CS valid value, the PPI_CS pin is valid will occur when the division counter counts to 0. If CS[CFG2].SYNC_CLK_SEL is set to 2, the PPI_CS pin is valid will occur when the division counter counts to 2, as shown in the following figure.



29.5.4 Address mapping conversion relationship

PPI corresponds to the chip's internal address from 0xF800_0000 to 0xFFFF_FFFF. The address range of peripheral addresses on the chip's memory map can be configured. When the data read and write access address on the bus matches the configuration of the CS[CFG0] register, the corresponding CS configuration register group will be triggered to generate the target read and write timing to access the slave device connected to the corresponding CS.

- CS0 [CFG0] register to configure the external device corresponding to the peripheral chip select signal PPI_CS[0]
- CS1 [CFG0] register, which configures the external device corresponding to the peripheral chip select signal PPI_CS[1]
- CS2[CFG0] register, which configures the external device corresponding to the peripheral chip select signal PPI_CS[2]
- CS3 [CFG0] register, which configures the external device corresponding to the peripheral chip select signal PPI_CS[3]

address translation is related to the peripheral data bit width (data_size), and provides a mask of the upper 16-bit addresses for unused upper address PPI. Configure the CS[CFG1] register, and the specific conversion is as follows:

- data_size =32bit, { Haddr[31:16] & mask[15:0], Haddr[15:0] } is shifted to the right by 2 bits;
- data_size =16bit, { Haddr[31:16] & mask[15:0], Haddr[15:0] } is shifted to the right by 1 bit;
- data_size =8bit, { Haddr[31:16] & mask[15:0], Haddr[15:0] } is shifted to the right by 0bit;

29.5.5 interrupt

PPI interrupt source: When the transmission time of an AHB (including burst) exceeds the configured system clock number (tm_cfg), it will be considered a timeout, and the interrupt status is reflected in the corresponding flag bit in the status register IRQ_STS.

29.6 Programmable Parallel Port Register

29.6.1 register description

programmable parallel port is as follows:

PPI base address: 0xF00CC000

Offset	Register Name	Description	Reset Value
0x0000	GLB_CFG	glb_cfg	0x00000000
0x0004	PAD_CFG	pad_cfg	0x0F000000
0x0014	CLKPIN_CFG	clkpin_cfg	0x00000000
0x0018	TM_CFG	tm_cfg	0x00000000
0x0020	IRQ_STS	irq_sts	0x00000000
0x0024	IRQ_EN	irq_en	0x00000000
0x0040	CS[0][CFG0]	cfg0	0x00000000
0x0044	CS[0][CFG1]	cfg1	0xFFFF0000
0x0048	CS[0][CFG2]	cfg2	0x00000000
0x004C	CS[0][CFG3]	cfg3	0x00000000
0x0050	CS[0][CFG4]	cfg4	0x3F3F3F3F
0x0060	CS[1][CFG0]	cfg0	0x00000000
0x0064	CS[1][CFG1]	cfg1	0xFFFF0000
0x0068	CS[1][CFG2]	cfg2	0x00000000
0x006C	CS[1][CFG3]	cfg3	0x00000000
0x0070	CS[1][CFG4]	cfg4	0x3F3F3F3F
0x0080	CS[2][CFG0]	cfg0	0x00000000
0x0084	CS[2][CFG1]	cfg1	0xFFFF0000
0x0088	CS[2][CFG2]	cfg2	0x00000000
0x008C	CS[2][CFG3]	cfg3	0x00000000
0x0090	CS[2][CFG4]	cfg4	0x3F3F3F3F
0x00A0	CS[3][CFG0]	cfg0	0x00000000
0x00A4	CS[3][CFG1]	cfg1	0xFFFF0000

Offset	Register Name	Description	Reset Value
0x00A8	CS[3][CFG2]	cfg2	0x00000000
0x00AC	CS[3][CFG3]	cfg3	0x00000000
0x00B0	CS[3][CFG4]	cfg4	0x3F3F3F3F
0x0400	CMD[0][CMD_CFG]	cmd_cfg	0x00000000
0x0404	CMD[0][AD_CFG]	ad_cfg	0x00000000
0x0408	CMD[0][CTRL_CFG]	ctrl_cfg	0x00000000
0x0410	CMD[1][CMD_CFG]	cmd_cfg	0x00000000
0x0414	CMD[1][AD_CFG]	ad_cfg	0x00000000
0x0418	CMD[1][CTRL_CFG]	ctrl_cfg	0x00000000
0x0420	CMD[2][CMD_CFG]	cmd_cfg	0x00000000
0x0424	CMD[2][AD_CFG]	ad_cfg	0x00000000
0x0428	CMD[2][CTRL_CFG]	ctrl_cfg	0x00000000
0x0430	CMD[3][CMD_CFG]	cmd_cfg	0x00000000
0x0434	CMD[3][AD_CFG]	ad_cfg	0x00000000
0x0438	CMD[3][CTRL_CFG]	ctrl_cfg	0x00000000
0x0440	CMD[4][CMD_CFG]	cmd_cfg	0x00000000
0x0444	CMD[4][AD_CFG]	ad_cfg	0x00000000
0x0448	CMD[4][CTRL_CFG]	ctrl_cfg	0x00000000
0x0450	CMD[5][CMD_CFG]	cmd_cfg	0x00000000
0x0454	CMD[5][AD_CFG]	ad_cfg	0x00000000
0x0458	CMD[5][CTRL_CFG]	ctrl_cfg	0x00000000
0x0460	CMD[6][CMD_CFG]	cmd_cfg	0x00000000
0x0464	CMD[6][AD_CFG]	ad_cfg	0x00000000
0x0468	CMD[6][CTRL_CFG]	ctrl_cfg	0x00000000
0x0470	CMD[7][CMD_CFG]	cmd_cfg	0x00000000
0x0474	CMD[7][AD_CFG]	ad_cfg	0x00000000
0x0478	CMD[7][CTRL_CFG]	ctrl_cfg	0x00000000
0x0480	CMD[8][CMD_CFG]	cmd_cfg	0x00000000
0x0484	CMD[8][AD_CFG]	ad_cfg	0x00000000
0x0488	CMD[8][CTRL_CFG]	ctrl_cfg	0x00000000
0x0490	CMD[9][CMD_CFG]	cmd_cfg	0x00000000
0x0494	CMD[9][AD_CFG]	ad_cfg	0x00000000
0x0498	CMD[9][CTRL_CFG]	ctrl_cfg	0x00000000
0x04A0	CMD[10][CMD_CFG]	cmd_cfg	0x00000000
0x04A4	CMD[10][AD_CFG]	ad_cfg	0x00000000
0x04A8	CMD[10][CTRL_CFG]	ctrl_cfg	0x00000000
0x04B0	CMD[11][CMD_CFG]	cmd_cfg	0x00000000
0x04B4	CMD[11][AD_CFG]	ad_cfg	0x00000000
0x04B8	CMD[11][CTRL_CFG]	ctrl_cfg	0x00000000

Offset	Register Name	Description	Reset Value
0x04C0	CMD[12][CMD_CFG]	cmd_cfg	0x00000000
0x04C4	CMD[12][AD_CFG]	ad_cfg	0x00000000
0x04C8	CMD[12][CTRL_CFG]	ctrl_cfg	0x00000000
0x04D0	CMD[13][CMD_CFG]	cmd_cfg	0x00000000
0x04D4	CMD[13][AD_CFG]	ad_cfg	0x00000000
0x04D8	CMD[13][CTRL_CFG]	ctrl_cfg	0x00000000
0x04E0	CMD[14][CMD_CFG]	cmd_cfg	0x00000000
0x04E4	CMD[14][AD_CFG]	ad_cfg	0x00000000
0x04E8	CMD[14][CTRL_CFG]	ctrl_cfg	0x00000000
0x04F0	CMD[15][CMD_CFG]	cmd_cfg	0x00000000
0x04F4	CMD[15][AD_CFG]	ad_cfg	0x00000000
0x04F8	CMD[15][CTRL_CFG]	ctrl_cfg	0x00000000
0x0500	CMD[16][CMD_CFG]	cmd_cfg	0x00000000
0x0504	CMD[16][AD_CFG]	ad_cfg	0x00000000
0x0508	CMD[16][CTRL_CFG]	ctrl_cfg	0x00000000
0x0510	CMD[17][CMD_CFG]	cmd_cfg	0x00000000
0x0514	CMD[17][AD_CFG]	ad_cfg	0x00000000
0x0518	CMD[17][CTRL_CFG]	ctrl_cfg	0x00000000
0x0520	CMD[18][CMD_CFG]	cmd_cfg	0x00000000
0x0524	CMD[18][AD_CFG]	ad_cfg	0x00000000
0x0528	CMD[18][CTRL_CFG]	ctrl_cfg	0x00000000
0x0530	CMD[19][CMD_CFG]	cmd_cfg	0x00000000
0x0534	CMD[19][AD_CFG]	ad_cfg	0x00000000
0x0538	CMD[19][CTRL_CFG]	ctrl_cfg	0x00000000
0x0540	CMD[20][CMD_CFG]	cmd_cfg	0x00000000
0x0544	CMD[20][AD_CFG]	ad_cfg	0x00000000
0x0548	CMD[20][CTRL_CFG]	ctrl_cfg	0x00000000
0x0550	CMD[21][CMD_CFG]	cmd_cfg	0x00000000
0x0554	CMD[21][AD_CFG]	ad_cfg	0x00000000
0x0558	CMD[21][CTRL_CFG]	ctrl_cfg	0x00000000
0x0560	CMD[22][CMD_CFG]	cmd_cfg	0x00000000
0x0564	CMD[22][AD_CFG]	ad_cfg	0x00000000
0x0568	CMD[22][CTRL_CFG]	ctrl_cfg	0x00000000
0x0570	CMD[23][CMD_CFG]	cmd_cfg	0x00000000
0x0574	CMD[23][AD_CFG]	ad_cfg	0x00000000
0x0578	CMD[23][CTRL_CFG]	ctrl_cfg	0x00000000
0x0580	CMD[24][CMD_CFG]	cmd_cfg	0x00000000
0x0584	CMD[24][AD_CFG]	ad_cfg	0x00000000
0x0588	CMD[24][CTRL_CFG]	ctrl_cfg	0x00000000
0x0590	CMD[25][CMD_CFG]	cmd_cfg	0x00000000

Offset	Register Name	Description	Reset Value
0x0594	CMD[25][AD_CFG]	ad_cfg	0x00000000
0x0598	CMD[25][CTRL_CFG]	ctrl_cfg	0x00000000
0x05A0	CMD[26][CMD_CFG]	cmd_cfg	0x00000000
0x05A4	CMD[26][AD_CFG]	ad_cfg	0x00000000
0x05A8	CMD[26][CTRL_CFG]	ctrl_cfg	0x00000000
0x05B0	CMD[27][CMD_CFG]	cmd_cfg	0x00000000
0x05B4	CMD[27][AD_CFG]	ad_cfg	0x00000000
0x05B8	CMD[27][CTRL_CFG]	ctrl_cfg	0x00000000
0x05C0	CMD[28][CMD_CFG]	cmd_cfg	0x00000000
0x05C4	CMD[28][AD_CFG]	ad_cfg	0x00000000
0x05C8	CMD[28][CTRL_CFG]	ctrl_cfg	0x00000000
0x05D0	CMD[29][CMD_CFG]	cmd_cfg	0x00000000
0x05D4	CMD[29][AD_CFG]	ad_cfg	0x00000000
0x05D8	CMD[29][CTRL_CFG]	ctrl_cfg	0x00000000
0x05E0	CMD[30][CMD_CFG]	cmd_cfg	0x00000000
0x05E4	CMD[30][AD_CFG]	ad_cfg	0x00000000
0x05E8	CMD[30][CTRL_CFG]	ctrl_cfg	0x00000000
0x05F0	CMD[31][CMD_CFG]	cmd_cfg	0x00000000
0x05F4	CMD[31][AD_CFG]	ad_cfg	0x00000000
0x05F8	CMD[31][CTRL_CFG]	ctrl_cfg	0x00000000
0x0600	CMD[32][CMD_CFG]	cmd_cfg	0x00000000
0x0604	CMD[32][AD_CFG]	ad_cfg	0x00000000
0x0608	CMD[32][CTRL_CFG]	ctrl_cfg	0x00000000
0x0610	CMD[33][CMD_CFG]	cmd_cfg	0x00000000
0x0614	CMD[33][AD_CFG]	ad_cfg	0x00000000
0x0618	CMD[33][CTRL_CFG]	ctrl_cfg	0x00000000
0x0620	CMD[34][CMD_CFG]	cmd_cfg	0x00000000
0x0624	CMD[34][AD_CFG]	ad_cfg	0x00000000
0x0628	CMD[34][CTRL_CFG]	ctrl_cfg	0x00000000
0x0630	CMD[35][CMD_CFG]	cmd_cfg	0x00000000
0x0634	CMD[35][AD_CFG]	ad_cfg	0x00000000
0x0638	CMD[35][CTRL_CFG]	ctrl_cfg	0x00000000
0x0640	CMD[36][CMD_CFG]	cmd_cfg	0x00000000
0x0644	CMD[36][AD_CFG]	ad_cfg	0x00000000
0x0648	CMD[36][CTRL_CFG]	ctrl_cfg	0x00000000
0x0650	CMD[37][CMD_CFG]	cmd_cfg	0x00000000
0x0654	CMD[37][AD_CFG]	ad_cfg	0x00000000
0x0658	CMD[37][CTRL_CFG]	ctrl_cfg	0x00000000
0x0660	CMD[38][CMD_CFG]	cmd_cfg	0x00000000
0x0664	CMD[38][AD_CFG]	ad_cfg	0x00000000

Offset	Register Name	Description	Reset Value
0x0668	CMD[38][CTRL_CFG]	ctrl_cfg	0x00000000
0x0670	CMD[39][CMD_CFG]	cmd_cfg	0x00000000
0x0674	CMD[39][AD_CFG]	ad_cfg	0x00000000
0x0678	CMD[39][CTRL_CFG]	ctrl_cfg	0x00000000
0x0680	CMD[40][CMD_CFG]	cmd_cfg	0x00000000
0x0684	CMD[40][AD_CFG]	ad_cfg	0x00000000
0x0688	CMD[40][CTRL_CFG]	ctrl_cfg	0x00000000
0x0690	CMD[41][CMD_CFG]	cmd_cfg	0x00000000
0x0694	CMD[41][AD_CFG]	ad_cfg	0x00000000
0x0698	CMD[41][CTRL_CFG]	ctrl_cfg	0x00000000
0x06A0	CMD[42][CMD_CFG]	cmd_cfg	0x00000000
0x06A4	CMD[42][AD_CFG]	ad_cfg	0x00000000
0x06A8	CMD[42][CTRL_CFG]	ctrl_cfg	0x00000000
0x06B0	CMD[43][CMD_CFG]	cmd_cfg	0x00000000
0x06B4	CMD[43][AD_CFG]	ad_cfg	0x00000000
0x06B8	CMD[43][CTRL_CFG]	ctrl_cfg	0x00000000
0x06C0	CMD[44][CMD_CFG]	cmd_cfg	0x00000000
0x06C4	CMD[44][AD_CFG]	ad_cfg	0x00000000
0x06C8	CMD[44][CTRL_CFG]	ctrl_cfg	0x00000000
0x06D0	CMD[45][CMD_CFG]	cmd_cfg	0x00000000
0x06D4	CMD[45][AD_CFG]	ad_cfg	0x00000000
0x06D8	CMD[45][CTRL_CFG]	ctrl_cfg	0x00000000
0x06E0	CMD[46][CMD_CFG]	cmd_cfg	0x00000000
0x06E4	CMD[46][AD_CFG]	ad_cfg	0x00000000
0x06E8	CMD[46][CTRL_CFG]	ctrl_cfg	0x00000000
0x06F0	CMD[47][CMD_CFG]	cmd_cfg	0x00000000
0x06F4	CMD[47][AD_CFG]	ad_cfg	0x00000000
0x06F8	CMD[47][CTRL_CFG]	ctrl_cfg	0x00000000
0x0700	CMD[48][CMD_CFG]	cmd_cfg	0x00000000
0x0704	CMD[48][AD_CFG]	ad_cfg	0x00000000
0x0708	CMD[48][CTRL_CFG]	ctrl_cfg	0x00000000
0x0710	CMD[49][CMD_CFG]	cmd_cfg	0x00000000
0x0714	CMD[49][AD_CFG]	ad_cfg	0x00000000
0x0718	CMD[49][CTRL_CFG]	ctrl_cfg	0x00000000
0x0720	CMD[50][CMD_CFG]	cmd_cfg	0x00000000
0x0724	CMD[50][AD_CFG]	ad_cfg	0x00000000
0x0728	CMD[50][CTRL_CFG]	ctrl_cfg	0x00000000
0x0730	CMD[51][CMD_CFG]	cmd_cfg	0x00000000
0x0734	CMD[51][AD_CFG]	ad_cfg	0x00000000
0x0738	CMD[51][CTRL_CFG]	ctrl_cfg	0x00000000

Offset	Register Name	Description	Reset Value
0x0740	CMD[52][CMD_CFG]	cmd_cfg	0x00000000
0x0744	CMD[52][AD_CFG]	ad_cfg	0x00000000
0x0748	CMD[52][CTRL_CFG]	ctrl_cfg	0x00000000
0x0750	CMD[53][CMD_CFG]	cmd_cfg	0x00000000
0x0754	CMD[53][AD_CFG]	ad_cfg	0x00000000
0x0758	CMD[53][CTRL_CFG]	ctrl_cfg	0x00000000
0x0760	CMD[54][CMD_CFG]	cmd_cfg	0x00000000
0x0764	CMD[54][AD_CFG]	ad_cfg	0x00000000
0x0768	CMD[54][CTRL_CFG]	ctrl_cfg	0x00000000
0x0770	CMD[55][CMD_CFG]	cmd_cfg	0x00000000
0x0774	CMD[55][AD_CFG]	ad_cfg	0x00000000
0x0778	CMD[55][CTRL_CFG]	ctrl_cfg	0x00000000
0x0780	CMD[56][CMD_CFG]	cmd_cfg	0x00000000
0x0784	CMD[56][AD_CFG]	ad_cfg	0x00000000
0x0788	CMD[56][CTRL_CFG]	ctrl_cfg	0x00000000
0x0790	CMD[57][CMD_CFG]	cmd_cfg	0x00000000
0x0794	CMD[57][AD_CFG]	ad_cfg	0x00000000
0x0798	CMD[57][CTRL_CFG]	ctrl_cfg	0x00000000
0x07A0	CMD[58][CMD_CFG]	cmd_cfg	0x00000000
0x07A4	CMD[58][AD_CFG]	ad_cfg	0x00000000
0x07A8	CMD[58][CTRL_CFG]	ctrl_cfg	0x00000000
0x07B0	CMD[59][CMD_CFG]	cmd_cfg	0x00000000
0x07B4	CMD[59][AD_CFG]	ad_cfg	0x00000000
0x07B8	CMD[59][CTRL_CFG]	ctrl_cfg	0x00000000
0x07D0	CMD[61][CMD_CFG]	cmd_cfg	0x00000000
0x07D4	CMD[61][AD_CFG]	ad_cfg	0x00000000
0x07D8	CMD[61][CTRL_CFG]	ctrl_cfg	0x00000000
0x07E0	CMD[62][CMD_CFG]	cmd_cfg	0x00000000
0x07E4	CMD[62][AD_CFG]	ad_cfg	0x00000000
0x07E8	CMD[62][CTRL_CFG]	ctrl_cfg	0x00000000
0x07F0	CMD[63][CMD_CFG]	cmd_cfg	0x00000000
0x07F4	CMD[63][AD_CFG]	ad_cfg	0x00000000
0x07F8	CMD[63][CTRL_CFG]	ctrl_cfg	0x00000000

Table 160: PPI Register List

timer are described in detail as follows:

29.6.2 GLB_CFG (0x0)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																N/A															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

GLB_CFG [31:0]

Bits	Name	Description
1	PAD_OUT_REG_ENJ	0: register output, one cycle delay; 1: direct output, no delay but may have timing issue
0	SOFT_RESET	software reset

GLB_CFG Bits

29.6.3 PAD_CFG (0x4)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD				CS_IDLE_ST				RSVD								CTRL_PAD_OE								CTRL_PAD_POL							
N/A				RW				N/A								RW								RW							
x	x	x	x	1	1	1	1	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PAD_CFG [31:0]

Bits	Name	Description
27-24	CS_IDLE_ST	cs pin idle state, default high for active low
15-8	CTRL_PAD_OE	the pad output enable signal. 0 for IN; 1 for OUT. NOTE: for unused pads, set both ctrl_pad_oe and ctrl_pad_pol to 0
7-0	CTRL_PAD_POL	for OUT pad: 0: output the value in cmd 1: output reversed value in cmd for IN pad, defines the signal active value, when ctrl_cfg.io_cfg is set, will wait the active value for ready(generally read or write ready)

PAD_CFG Bits

29.6.4 CLKPIN_CFG (0x14)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD				CYCLE				RSVD				HIGH				RSVD				LOW				RSVD		INVERT	RSVD			AON	EN
N/A				RW				N/A				RW				N/A				RW				N/A		RW	N/A			RW	RW
x	x	x	x	0	0	0	0	x	x	x	x	0	0	0	0	x	x	x	x	0	0	0	0	x	x	0	x	x	x	0	0

CLKPIN_CFG [31:0]

Bits	Name	Description
27-24	CYCLE	there will be a system counter run from 0 to cycle, clk output will be set to high when counter is clk_high, and low when counter is clk_low. The output will be system clock if cycle is 0. All 4 CS share same clock configuration(one clock pin with configured frequency). different CS can be assert at different counter value.
19-16	HIGH	clock high numer
11-8	LOW	clock low number
5	INVERT	set to invert clock output
1	AON	0: use clk_gate in cmd sequence for whether output clock 1: always enable clock output;
0	EN	set to enable clock logic

CLKPIN_CFG Bits

29.6.5 TM_CFG (0x18)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD															TM_EN	RSVD				TM_CFG											
N/A															RW	N/A				RW											
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0

TM_CFG [31:0]

Bits	Name	Description
16	TM_EN	timeout enable. if enabled, then if each AHB transfer time exceed tm_cfg clock cycles, will assert irq
11-0	TM_CFG	timeout value, max 20us at 200MHz clock

TM_CFG Bits

29.6.6 IRQ_STS (0x20)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																IRQ_TMOUT_STS															
N/A																RW1C															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0

IRQ_STS [31:0]

Bits	Name	Description
0	IRQ_TMOUT_STS	tiemout interrupt status, write 1 to clear

IRQ_STS Bits

29.6.7 IRQ_EN (0x24)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																IRQ_TMOUT_EN															
N/A																RW															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0

IRQ_EN [31:0]

Bits	Name	Description
0	IRQ_TMOUT_EN	timeout interrupt enable

IRQ_EN Bits

29.6.8 CS[CFG0] (0x40 + 0x20 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD				ADDR_END												RSVD				ADDR_START											
N/A				RW												N/A				RW											
x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0

CS[CFG0] [31:0]

Bits	Name	Description
27-16	ADDR_END	
11-0	ADDR_START	addr_start and addr_end config the address slot for CS0, use high 12bit, the minimun slot is 1Mbyte(addr_start==addr_end)

CS[CFG0] Bits

29.6.9 CS[CFG1] (0x44 + 0x20 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR_MASK																RSVD										ADDR_SHIFT					
RW																N/A										RW					
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

CS[CFG1] [31:0]

Bits	Name	Description
31-16	ADDR_MASK	the high AHB address will AND with {cs0_mask[15:0], 16'hFFFF}, shift right with addr_shift, then output as real address.
3-0	ADDR_SHIFT	gennerally should be configured according to port size, 0 for 8bit; 1 for 16bit; 2 for 32bit;

CS[CFG1] Bits

29.6.10 CS[CFG2] (0x48 + 0x20 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD			CS_SYNC_EN	RSVD				SYNC_CLK_SEL				RSVD					READY_IN_SEL		RSVD			INTER_CMD_DLY				DM_POLARITY	PORT_SIZE		ENABLE		
N/A			RW	N/A				RW				N/A					RW	N/A			RW				RW	RW		RW			
x	x	x	0	x	x	x	x	0	0	0	0	x	x	x	x	x	x	x	0	x	x	x	x	0	0	0	0	0	0	0	0

CS[CFG2] [31:0]

Bits	Name	Description
28	CS_SYNC_EN	set to enable CS pin sync with clock counter. Clr if use async mode(no clk pin), or not care the CS start time with clk pin
23-20	SYNC_CLK_SEL	CS assert at when clk_div_cnt equal to sync_clk_sel
12	READY_IN_SEL	0: use two stage sync; 1: use one stage sync
7-4	INTER_CMD_DLY	set to none-zero value, will add delay between each command sequence for burst cmd, or splited transfer cmd sequence(such as transfer 32bit on 16bit port), CS will be de-assert during the delay.
3	DM_POLARITY	dm polarity
2-1	PORT_SIZE	00-8bit; 01-16bit; 10-32bit; 11-reserved
0	ENABLE	CS enable

CS[CFG2] Bits

29.6.11 CS[CFG3] (0x4C + 0x20 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																															
x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CS[CFG3] [31:0]

Bits	Name	Description
29-24	RCMD_END1	sequential read cmd end index
21-16	RCMD_START1	sequential read cmd start index
13-8	RCMD_END0	first read cmd end index
5-0	RCMD_START0	first read cmd start index

CS[CFG3] Bits

29.6.12 CS[CFG4] (0x50 + 0x20 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD		WCMD_END1						RSVD		WCMD_START1						RSVD		WCMD_END0						RSVD		WCMD_START0					
N/A		RW						N/A		RW						N/A		RW						N/A		RW					
x	x	1	1	1	1	1	1	x	x	1	1	1	1	1	1	x	x	1	1	1	1	1	1	x	x	1	1	1	1	1	1

CS[CFG4] [31:0]

Bits	Name	Description
29-24	WCMD_END1	sequential write cmd end index
21-16	WCMD_START1	sequential write cmd start index
13-8	WCMD_END0	first write cmd end index
5-0	WCMD_START0	first write cmd start index

CS[CFG4] Bits

29.6.13 CMD[CMD_CFG] (0x400 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD														CS_VAL	CLK_GATE	RSVD						CYCLE_NUM									
N/A														RW	RW	N/A						RW									
x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0

CMD[CMD_CFG] [31:0]

Bits	Name	Description
17	CS_VAL	cs value in current cmd
16	CLK_GATE	the clock gate enable signal, set to output clock signal
7-0	CYCLE_NUM	cmd clock cycles

CMD[CMD_CFG] Bits

29.6.14 CMD[AD_CFG] (0x404 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD														DIR3	AD_SEL3	BYTE_SEL3	DIR2	AD_SEL2	BYTE_SEL2	DIR1	AD_SEL1	BYTE_SEL1	DIR0	AD_SEL0	BYTE_SEL0	DIR0	AD_SEL0	BYTE_SEL0	DIR0	AD_SEL0	BYTE_SEL0
N/A														RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

CMD[AD_CFG] [31:0]

Bits	Name	Description
15	DIR3	
14	AD_SEL3	
13-12	BYTE_SEL3	
11	DIR2	
10	AD_SEL2	
9-8	BYTE_SEL2	
7	DIR1	
6	AD_SEL1	
5-4	BYTE_SEL1	
3	DIR0	0 for OUT; 1 for IN
2	AD_SEL0	0 for data; 1 for address.
1-0	BYTE_SEL0	select one of the 4 bytes(11 for 31:24, 10 for 23:16, 01 for 15:8, 00 for 7:0)

CMD[AD_CFG] Bits

29.6.15 CMD[CTRL_CFG] (0x408 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD			IO_CFG7		RSVD		IO_CFG6		RSVD		IO_CFG5		RSVD		IO_CFG4		RSVD		IO_CFG3		RSVD		IO_CFG2		RSVD		IO_CFG1		RSVD		IO_CFG0
N/A			RW		N/A		RW		N/A		RW		N/A		RW		N/A		RW		N/A		RW		N/A		RW		N/A		RW
x	x	x	0	x	x	x	0	x	x	x	0	x	x	x	0	x	x	x	0	x	x	x	0	x	x	x	0	x	x	x	0

CMD[CTRL_CFG] [31:0]

Bits	Name	Description
28	IO_CFG7	
24	IO_CFG6	
20	IO_CFG5	
16	IO_CFG4	
12	IO_CFG3	
8	IO_CFG2	
4	IO_CFG1	
0	IO_CFG0	for OUT, it defines the output value(0 or 1); for IN, it defines whether to wait for ready(ready polarity is defined in ctrl_pad_pol)

CMD[CTRL_CFG] Bits

30 Algorithm Acceleration Overview

This chapter introduces the TFA and CRC modules.

30.1 Transcendental Function Accelerator (TFA)

This product is equipped with a function calculation accelerator. The supported functions and their input ranges are shown in [Table 161](#):

Function Name	Function Formula	Independent Variable Range
INV	$y = \frac{1}{x}$	$(-\infty, \infty)$, -NaN ¹ , +NaN
SQRT	$y = \sqrt{x}$	$(-\infty, \infty)$, -NaN, +NaN
SIN	$y = \sin(2\pi x)$	$(-\infty, \infty)$, -NaN, +NaN
COS	$y = \cos(2\pi x)$	$(-\infty, \infty)$, -NaN, +NaN
IEXP2	$y = 2^x$	$(-\infty, \infty)$, -NaN, +NaN
LOG2	$y = \log_2 x$	$(-\infty, \infty)$, -NaN, +NaN
ATANPU	$y = \frac{\arctan(x)}{2\pi}$	$[-1, 1]$, -NaN, +NaN, $-\infty$, $+\infty$
INVSQRT	$y = \frac{1}{\sqrt{x}}$	$(-\infty, \infty)$, -NaN, +NaN

¹ NaN: Not a Number

Table 161: Operators Supported by TFA

Users can directly call the APIs provided by the supporting software for use.

30.2 Cyclic Redundancy Check (CRC)

This product integrates 1 CRC module, which supports multi-channel cyclic redundancy check with arbitrary polynomials.

31 Cyclic Redundancy Detection CRC

This chapter describes the functions and features of cyclic redundancy detection CRC.

31.1 Feature Summary

Cyclic Redundancy Detection CRC are as follows:

- supports independent calculation of 8 channels
- supports programmable CRC algorithms (can support all CRC algorithms with the highest power less than or equal to 32)
- supports per-byte, double-byte, and full-word transfers
- supports quick configuration of 15 common CRC algorithms

31.2 Function Description

Cyclic Redundancy Detection CRC module serves as a slave device on the system APB bus. Users can use bus configuration to input code stream and read CRC check code. The CRC module can input the code stream to be verified into the CHN[n][DATA] register according to the user-defined CRC algorithm, and the corresponding CRC check code can be obtained from CHN[n][result]. The CRC module provides 8 independent calculation channels, each channel can independently configure its CRC algorithm and complete the calculation independently. The input code stream supports byte, double byte, and full-word transmission, and also supports the code stream configured to flip order by byte or bit. Each channel supports fast pre-configuration of 15 common CRC algorithms.

CRC algorithm configuration needs to be carried out before CRC is used, which can be quickly completed by configuring each algorithm register in detail or configuring pre-configured registers. Support discontinuous transmission of code stream, and the calculation results of discontinuous transmission of code stream and continuous transmission are consistent. The calculation result in CHN[n][result] is always the check code of the currently accumulated transmitted code stream.

31.3 CRC algorithm configuration

- CHN[n][POLY]: Expression of the CRC algorithm polynomial. For example, the polynomial corresponding to CRC-32 is expressed as 0x04C11DB7. If the CRC polynomial is lower than 32 bits, this configuration register can be written to 0 in the high order.
- CHN[n][INIT_DATA]: CRC initial value.
- CHN [n][XOROUT]: The exclusive OR value of the CRC output value, that is, the final check code is obtained after the CRC calculation result is XORed with this parameter.
- CHN [n][MISC_SETTING]: Miscellaneous configuration of CRC, including: POLY_WIDTH(length of CRC check code), REV_IN(whether to flip the code stream bit-by-bit), REV_OUT(whether to flip the output result bit-by-bit), BYTE_REV(whether to flip the input code stream bit-by-byte).
- CHN[n][PRE_SET]: CRC fast pre-configuration register. The priority of the CRC algorithm corresponding to the pre-configuration register is higher than the configuration of the above-mentioned detailed algorithm parameter register, that is, when the user configures the pre-configuration register, if any previously configured algorithm parameters will be overwritten by the corresponding pre-configured parameters.

31.4 CRC register

CRC is as follows:

CRC base address: 0xF0080000

Offset	Register Name	Description	Reset Value
0x0000	CHN[0][PRE_SET]	0 pre set for crc setting	0x00000000
0x0004	CHN[0][CLR]	chn0 clear crc result and setting	0x00000000
0x0008	CHN[0][POLY]	chn0 poly	0x00000000
0x000C	CHN[0][INIT_DATA]	chn0 init_data	0x00000000
0x0010	CHN[0][XOROUT]	chn0 xorout	0x00000000
0x0014	CHN[0][MISC_SETTING]	chn0 misc_setting	0x00000000
0x0018	CHN[0][DATA]	chn0 data	0x00000000
0x001C	CHN[0][RESULT]	chn0 result	0x00000000
0x0040	CHN[1][PRE_SET]	1 pre set for crc setting	0x00000000
0x0044	CHN[1][CLR]	chn1 clear crc result and setting	0x00000000
0x0048	CHN[1][POLY]	chn1 poly	0x00000000
0x004C	CHN[1][INIT_DATA]	chn1 init_data	0x00000000
0x0050	CHN[1][XOROUT]	chn1 xorout	0x00000000
0x0054	CHN[1][MISC_SETTING]	chn1 misc_setting	0x00000000
0x0058	CHN[1][DATA]	chn1 data	0x00000000
0x005C	CHN[1][RESULT]	chn1 result	0x00000000
0x0080	CHN[2][PRE_SET]	2 pre set for crc setting	0x00000000
0x0084	CHN[2][CLR]	chn2 clear crc result and setting	0x00000000
0x0088	CHN[2][POLY]	chn2 poly	0x00000000
0x008C	CHN[2][INIT_DATA]	chn2 init_data	0x00000000
0x0090	CHN[2][XOROUT]	chn2 xorout	0x00000000
0x0094	CHN[2][MISC_SETTING]	chn2 misc_setting	0x00000000
0x0098	CHN[2][DATA]	chn2 data	0x00000000
0x009C	CHN[2][RESULT]	chn2 result	0x00000000
0x00C0	CHN[3][PRE_SET]	3 pre set for crc setting	0x00000000
0x00C4	CHN[3][CLR]	chn3 clear crc result and setting	0x00000000
0x00C8	CHN[3][POLY]	chn3 poly	0x00000000
0x00CC	CHN[3][INIT_DATA]	chn3 init_data	0x00000000
0x00D0	CHN[3][XOROUT]	chn3 xorout	0x00000000
0x00D4	CHN[3][MISC_SETTING]	chn3 misc_setting	0x00000000
0x00D8	CHN[3][DATA]	chn3 data	0x00000000
0x00DC	CHN[3][RESULT]	chn3 result	0x00000000
0x0100	CHN[4][PRE_SET]	4 pre set for crc setting	0x00000000
0x0104	CHN[4][CLR]	chn4 clear crc result and setting	0x00000000
0x0108	CHN[4][POLY]	chn4 poly	0x00000000
0x010C	CHN[4][INIT_DATA]	chn4 init_data	0x00000000

Offset	Register Name	Description	Reset Value
0x0110	CHN[4][XOROUT]	chn4 xorout	0x00000000
0x0114	CHN[4][MISC_SETTING]	chn4 misc_setting	0x00000000
0x0118	CHN[4][DATA]	chn4 data	0x00000000
0x011C	CHN[4][RESULT]	chn4 result	0x00000000
0x0140	CHN[5][PRE_SET]	5 pre set for crc setting	0x00000000
0x0144	CHN[5][CLR]	chn5 clear crc result and setting	0x00000000
0x0148	CHN[5][POLY]	chn5 poly	0x00000000
0x014C	CHN[5][INIT_DATA]	chn5 init_data	0x00000000
0x0150	CHN[5][XOROUT]	chn5 xorout	0x00000000
0x0154	CHN[5][MISC_SETTING]	chn5 misc_setting	0x00000000
0x0158	CHN[5][DATA]	chn5 data	0x00000000
0x015C	CHN[5][RESULT]	chn5 result	0x00000000
0x0180	CHN[6][PRE_SET]	6 pre set for crc setting	0x00000000
0x0184	CHN[6][CLR]	chn6 clear crc result and setting	0x00000000
0x0188	CHN[6][POLY]	chn6 poly	0x00000000
0x018C	CHN[6][INIT_DATA]	chn6 init_data	0x00000000
0x0190	CHN[6][XOROUT]	chn6 xorout	0x00000000
0x0194	CHN[6][MISC_SETTING]	chn6 misc_setting	0x00000000
0x0198	CHN[6][DATA]	chn6 data	0x00000000
0x019C	CHN[6][RESULT]	chn6 result	0x00000000
0x01C0	CHN[7][PRE_SET]	7 pre set for crc setting	0x00000000
0x01C4	CHN[7][CLR]	chn7 clear crc result and setting	0x00000000
0x01C8	CHN[7][POLY]	chn7 poly	0x00000000
0x01CC	CHN[7][INIT_DATA]	chn7 init_data	0x00000000
0x01D0	CHN[7][XOROUT]	chn7 xorout	0x00000000
0x01D4	CHN[7][MISC_SETTING]	chn7 misc_setting	0x00000000
0x01D8	CHN[7][DATA]	chn7 data	0x00000000
0x01DC	CHN[7][RESULT]	chn7 result	0x00000000

Table 162: CRC Register List

31.5 CRC register details

CRC is as follows:

31.5.1 CHN[PRE_SET] (0x0 + 0x40 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								PRE_SET							
N/A																								RW							

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0

CHN[PRE_SET] [31:0]

Bits	Name	Description
7-0	PRE_SET	0: no pre set 1: CRC32 2: CRC32-AUTOSAR 3: CRC16-CCITT 4: CRC16-XMODEM 5: CRC16-MODBUS 6: crc16_dnp 7: crc16_x25 8: crc16_usb 9: crc16_maxim 10: crc16_ibm 11: crc8_maxim 12: crc8_rohc 13: crc8_itu 14: crc8 15: crc5_usb

CHN[PRE_SET] Bits

31.5.2 CHN[CLR] (0x4 + 0x40 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																CLR															
N/A																RW															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0

CHN[CLR] [31:0]

Bits	Name	Description
0	CLR	write 1 to clr crc setting and result for its channel. always read 0.

CHN[CLR] Bits

31.5.3 CHN[POLY] (0x8 + 0x40 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POLY																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CHN[POLY] [31:0]

Bits	Name	Description
31-0	POLY	poly setting

CHN[POLY] Bits

31.5.4 CHN[INIT_DATA] (0xC + 0x40 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INIT_DATA																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CHN[INIT_DATA] [31:0]

Bits	Name	Description
31-0	INIT_DATA	initial data of CRC

CHN[INIT_DATA] Bits

31.5.5 CHN[XOROUT] (0x10 + 0x40 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XOROUT																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CHN[XOROUT] [31:0]

Bits	Name	Description
31-0	XOROUT	XOR for CRC result

CHN[XOROUT] Bits

31.5.6 CHN[MISC_SETTING] (0x14 + 0x40 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RSVD							BYTE_REV	RSVD							REV_OUT	RSVD							REV_IN	RSVD		POLY_WIDTH						
N/A							RW	N/A							RW	N/A							RW	N/A		RW						
x	x	x	x	x	x	x	0	x	x	x	x	x	x	x	0	x	x	x	x	x	x	x	0	x	x	0	0	0	0	0	0	

CHN[MISC_SETTING] [31:0]

Bits	Name	Description
24	BYTE_REV	0: no wrap input byte order 1: wrap input byte order
16	REV_OUT	0: no wrap output bit order 1: wrap output bit order
8	REV_IN	0: no wrap input bit order 1: wrap input bit order
5-0	POLY_WIDTH	crc data length

CHN[MISC_SETTING] Bits

31.5.7 CHN[DATA] (0x18 + 0x40 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CHN[DATA] [31:0]

Bits	Name	Description
31-0	DATA	data for crc

CHN[DATA] Bits

31.5.8 CHN[RESULT] (0x1C + 0x40 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESULT																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

CHN[RESULT] [31:0]

Bits	Name	Description
31-0	RESULT	crc result

CHN[RESULT] Bits

32 Overview of DMA and Mailbox

This chapter describes the DMA controller, DMAMUX, and communication mailbox MBX.

32.1 DMA Controller

This product includes 2 DMA controllers, XDMA and HDMA.

- XDMA is connected as a master device to the AXI system bus. It can achieve efficient data transfer between memories or between memory and external memory. It can also be used for data transfer between peripheral registers and memory, or between peripheral registers.
- HDMA is connected as a master device to the AHB peripheral bus. It can realize real-time data transfer between peripheral registers and memory, and between memories.

XDMA and HDMA have the same programming model. XDMA and HDMA each have 32 DMA request input signals, and the request sources of these DMA come from the DMA Request Router (DMAMUX).

32.2 DMA Request Router

DMA Request Router (DMAMUX). DMA requests from each peripheral module are assigned to the 64 output signal, which serves as the DMA transfer request source for XDMA and HDMA. The output 0 ~ 31 of the

DMAMUX is connected to the peripheral bus DMA controller HDMA, and the output 32 ~ 63 of the DMAMUX is connected to the system bus DMA controller XDMA.

The specific input allocation of DMAMUX is as follows:

DMA MUX Index	DMA MUX SOURCE Name	Description
0	SPI0_RX	SPI0 Receive
1	SPI0_TX	SPI0 Transmit
2	SPI1_RX	SPI1 Receive
3	SPI1_TX	SPI1 Transmit
4	SPI2_RX	SPI2 Receive
5	SPI2_TX	SPI2 Transmit
6	SPI3_RX	SPI3 Receive
7	SPI3_TX	SPI3 Transmit
8	UART0_RX	UART0 Receive
9	UART0_TX	UART0 Transmit
10	UART1_RX	UART1 Receive
11	UART1_TX	UART1 Transmit
12	UART2_RX	UART2 Receive
13	UART2_TX	UART2 Transmit
14	UART3_RX	UART3 Receive
15	UART3_TX	UART3 Transmit
16	UART4_RX	UART4 Receive
17	UART4_TX	UART4 Transmit
18	UART5_RX	UART5 Receive

DMA MUX Index	DMA MUX SOURCE Name	Description
19	UART5_TX	UART5 Transmit
20	UART6_RX	UART6 Receive
21	UART6_TX	UART6 Transmit
22	UART7_RX	UART7 Receive
23	UART7_TX	UART7 Transmit
24	I2C0	I2C0
25	I2C1	I2C1
26	I2C2	I2C2
27	I2C3	I2C3
28	MCAN0	CAN0
29	MCAN1	CAN1
30	MCAN2	CAN2
31	MCAN3	CAN3
32	GPTMR0_0	GPTMR0 channel 0
33	GPTMR0_1	GPTMR0 channel 1
34	GPTMR0_2	GPTMR0 channel 2
35	GPTMR0_3	GPTMR0 channel 3
36	GPTMR1_0	GPTMR1 channel 0
37	GPTMR1_1	GPTMR1 channel 1
38	GPTMR1_2	GPTMR1 channel 2
39	GPTMR1_3	GPTMR1 channel 3
40	GPTMR2_0	GPTMR2 channel 0
41	GPTMR2_1	GPTMR2 channel 1
42	GPTMR2_2	GPTMR2 channel 2
43	GPTMR2_3	GPTMR2 channel 3
44	GPTMR3_0	GPTMR3 channel 0
45	GPTMR3_1	GPTMR3 channel 1
46	GPTMR3_2	GPTMR3 channel 2
47	GPTMR3_3	GPTMR3 channel 3
48	OWR0_RX	Owire0 Receive
49	OWR0_TX	Owire0 Transmit
50	OWR1_RX	Owire1 Receive
51	OWR1_TX	Owire1 Transmit
52	MOT_0	TRGM DMA0(include pwm0~1,qei0~1,trgm0~1)
53	MOT_1	TRGM DMA1(include pwm0~1,qei0~1,trgm0~1)
54	MOT_2	TRGM DMA2(include pwm0~1,qei0~1,trgm0~1)

DMA MUX Index	DMA MUX SOURCE Name	Description
55	MOT_3	TRGM DMA3(include pwm0~1,qei0~1,trgm0~1)
56	MOT_4	TRGM DMA4(include pwm0~1,qei0~1,trgm0~1)
57	MOT_5	TRGM DMA5(include pwm0~1,qei0~1,trgm0~1)
58	MOT_6	TRGM DMA6(include pwm0~1,qei0~1,trgm0~1)
59	MOT_7	TRGM DMA7(include pwm0~1,qei0~1,trgm0~1)
60	ACMP0_0	ACMP0
61	ACMP0_1	ACMP1
62	XPI0_RX	XPI0 Receive
63	XPI0_TX	XPI0 Transmit
64	ESC_SYNC0	Ethercat sync0
65	ESC_SYNC1	Ethercat sync1

Table 163: DMA MUX List

32.3 Mail Box MBX

This product supports 1 communication mailbox MBX for inter-process communication. Processors can use MBX to send data to each other, which supports generating interrupts.

33 DMA Controller

This chapter introduces the functionality and features of the DMA controller.

33.1 Overview

The main features of the DMA controller are:

- Supports 32 configurable channels
- Channels support 2-level priority configuration
- Same priority channels use Round-Robin arbitration
- Supports chaining multiple DMA tasks
- Supports infinite loop
- Supports fixed burst (small loop)
- Supports byte swapping

33.2 DMA Architecture Block Diagram

The structure diagram of this module is shown in Figure 26:

In the DMA module, dma_apb_csr is the control register module, the dma_ahb_ahbmst module implements

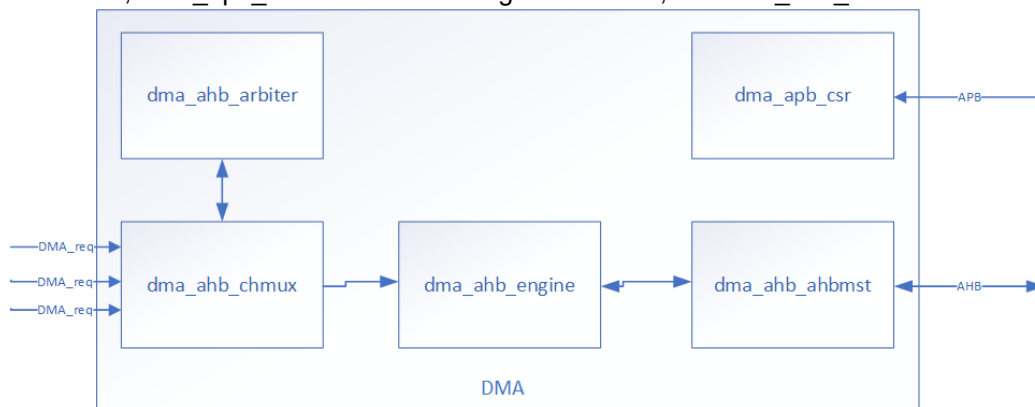


Figure 26: DMA Structure Block Diagram

the AHB master interface, the DMA_ahb_engine module implements the DMA operation engine state machine, and the dma_ahb_arbiter and dma_ahb_chmux modules implement arbitration and selection for multiple DMA requests.

33.3 Pin Description

Table 164: DMA Pin Description

Pin	Direction	Function Description
dma_req[31:0]	Input	Input DMA request signals
dma_ack[31:0]	Output	Output DMA acknowledge signals

33.4 Functional Description

The DMA controller supports 32 channels, and each channel can be independently configured for data transfer parameters. Users can use multiple DMA channels simultaneously. The DMA controller will alternately execute data transfer tasks for different channels, rather than waiting for one channel's transfer to complete before responding to the next channel's transfer request. This means the DMA controller may complete a single channel's data transfer in multiple segments. After all tasks for a channel are completed, the corresponding flag bit in the status register will be set.

A DMA channel's data transfer may be interrupted due to data transfer errors or can be aborted by software. In such cases, the DMA controller will stop transferring data and set the corresponding flag bit in the status register.

33.4.1 DMA Hardware Handshake

The DMA controller supports 32 pairs of request-acknowledge handshake signals. The request signal comes from the functional module and is routed to the DMA controller via the DMAMUX. The DMA controller uses a configured channel to execute this request. After completing a data transfer of length $\text{SrcBurstSize} * \text{SrcWidth}$, it sends an acknowledge signal to the DMAMUX, completing the hardware handshake.

33.4.2 DMA Chained Transfer

DMA supports chained transfer, allowing the continuous completion of multiple tasks with different configurations without processor intervention.

To use the DMA chained transfer function, users need to pre-build a series of linked DMA task descriptor lists in memory. The first element of the list is the control register of the DMA channel, and the remaining task descriptors are stored in memory. The `ChnLLPointer` control word in the DMA task descriptor is the pointer to the next task descriptor for DMA channel *n*. If the `ChnLLPointer` value is non-zero, the DMA controller will fetch the next task descriptor from the address pointed to by `ChnLLPointer` after completing the corresponding task of the current descriptor. If the value of `ChnLLPointer` is 0, it indicates that this task descriptor is the last in the list, and the chain terminates.

Table 165 lists the format of the task descriptor. The storage address of the descriptor must be 8-byte aligned, and the descriptor itself should not cross a 4K boundary.

33.4.3 Infinite Loop

DMA supports infinite loop mode. Before hardware starts transferring, it records the currently configured source address/destination address/transfer size. After completing the current transfer size, it resets the source address/destination address/transfer size and continues transferring.

Note: Chained transfer is not supported in infinite loop mode.

33.4.4 Fixed Burst

DMA supports fixed burst mode (small loop), and can be configured to support fixed burst for either the source address or the destination address.

After completing the current burst, the hardware will start from the configured source address/destination address and continue transferring a burst of data.

Table 165: DMA Chained Task Descriptor

Name	Address Offset	Description
Ctrl	0x00	Channel control, format same as CTRL register
TranSize	0x04	Total transfer size, format same as TRANSIZE register
SrcAddrL	0x08	Source Address L, format same as SRCADDRL register
SrcAddrH	0x0C	Source Address H, format same as SRCADDRH register
DstAddrL	0x10	Destination Address L, format same as DSTADDRL register
DstAddrH	0x14	Destination Address H, format same as DSTADDRH register
LLPointerL	0x18	Linked List Pointer L, format same as LLPOINTERL register
LLPointerH	0x1C	Linked List Pointer H, format same as LLPOINTERH register

Limited by the DMA FIFO size, the maximum number of bytes transferred per burst is: xdma 128 bytes, hdma 64 bytes.

Fixed burst mode only supports address increment.

33.4.5 Byte Swapping

DMA supports swapping read data by byte when transferring data on a multi-byte bus.

In this mode, the source data width must equal the destination data width.

HDMA supports 8-bit swapping for 16-bit accesses, and supports 8/16-bit swapping for 32-bit accesses.

XDMA supports 8-bit swapping for 16-bit accesses, supports 8/16-bit swapping for 32-bit accesses, and supports 8/16/32-bit swapping for 64-bit accesses.

DMA supports default 8/16/32-bit swap order, and can also support user-defined swap order. For details, see the register description.

33.4.6 Data Order

The DMA controller supports 4 address modes: increment mode, decrement mode, constant mode, and fixed burst mode.

- Increment mode means that after each DMA access to the source or destination address, the next DMA access will be to the address after this source or destination.
- Decrement mode means that after each DMA access to the source or destination address, the next DMA access will be to the address before this source or destination.
- Constant mode means that after each DMA access to the source or destination address, the next DMA access will be to a fixed, unchanged source or destination address.
- Fixed burst mode means that after each DMA accesses a burst of data for the source or destination address, the DMA re-accesses from the configured source or destination address.

When the source and destination address modes are configured the same, DMA will keep the byte order of

the data between the source and destination areas unchanged. However, if the source and destination addresses are configured oppositely, the byte order of the data between the source and destination will also be reversed. Among them, the constant mode byte order is the same as the increment mode.

33.5 DMA Registers

The list of DMA registers is as follows:

HDMA base address: 0xF00C8000

XDMA base address: 0xF3100000

Offset	Register Name	Description	Reset Value
0x0004	IDMISC	ID Misc	0x00000000
0x0010	DMACFG	DMAC Configuration Register	0x00000000
0x0014	DMACTRL	DMAC Control Register	0x00000000
0x0018	CHABORT	Channel Abort Register	0x00000000
0x0024	INTHALFSTS	Half Complete Interrupt Status	0x00000000
0x0028	INTTCSTS	Trans Complete Interrupt Status Register	0x00000000
0x002C	INTABORTSTS	Abort Interrupt Status Register	0x00000000
0x0030	INTERRSTS	Error Interrupt Status Register	0x00000000
0x0034	CHEN	Channel Enable Register	0x00000000
0x0040	CHCTRL[CH0][CTRL]	Channel ch0 Control Register	0x00000000
0x0044	CHCTRL[CH0][TRANSIZE]	Channel ch0 Transfer Size Register	0x00000000
0x0048	CHCTRL[CH0][SRCADDR]	Channel ch0 Source Address Low Part Register	0x00000000
0x004C	CHCTRL[CH0][CHANREQCTRL]	Channel ch0 DMA Request Control Register	0x00000000
0x0050	CHCTRL[CH0][DSTADDR]	Channel ch0 Destination Address Low Part Register	0x00000000
0x0054	CHCTRL[CH0][SWAPTABLE]	swap table register	0x00000000
0x0058	CHCTRL[CH0][LLPOINTER]	Channel ch0 Linked List Pointer Low Part Register	0x00000000
0x0060	CHCTRL[CH1][CTRL]	Channel ch1 Control Register	0x00000000
0x0064	CHCTRL[CH1][TRANSIZE]	Channel ch1 Transfer Size Register	0x00000000
0x0068	CHCTRL[CH1][SRCADDR]	Channel ch1 Source Address Low Part Register	0x00000000
0x006C	CHCTRL[CH1][CHANREQCTRL]	Channel ch1 DMA Request Control Register	0x00000000
0x0070	CHCTRL[CH1][DSTADDR]	Channel ch1 Destination Address Low Part Register	0x00000000

Offset	Register Name	Description	Reset Value
0x0074	CHCTRL[CH1][SWAPTABLE]	swap table register	0x00000000
0x0078	CHCTRL[CH1][LLPOINTER]	Channel ch1 Linked List Pointer Low Part Register	0x00000000
0x0080	CHCTRL[CH2][CTRL]	Channel ch2 Control Register	0x00000000
0x0084	CHCTRL[CH2][TRANSIZE]	Channel ch2Transfer Size Register	0x00000000
0x0088	CHCTRL[CH2][SRCADDR]	Channel ch2 Source Address Low Part Register	0x00000000
0x008C	CHCTRL[CH2][CHANREQCTRL]	Channel ch2 DMA Request Control Register	0x00000000
0x0090	CHCTRL[CH2][DSTADDR]	Channel ch2 Destination Address Low Part Register	0x00000000
0x0094	CHCTRL[CH2][SWAPTABLE]	swap table register	0x00000000
0x0098	CHCTRL[CH2][LLPOINTER]	Channel ch2 Linked List Pointer Low Part Register	0x00000000
0x00A0	CHCTRL[CH3][CTRL]	Channel ch3 Control Register	0x00000000
0x00A4	CHCTRL[CH3][TRANSIZE]	Channel ch3Transfer Size Register	0x00000000
0x00A8	CHCTRL[CH3][SRCADDR]	Channel ch3 Source Address Low Part Register	0x00000000
0x00AC	CHCTRL[CH3][CHANREQCTRL]	Channel ch3 DMA Request Control Register	0x00000000
0x00B0	CHCTRL[CH3][DSTADDR]	Channel ch3 Destination Address Low Part Register	0x00000000
0x00B4	CHCTRL[CH3][SWAPTABLE]	swap table register	0x00000000
0x00B8	CHCTRL[CH3][LLPOINTER]	Channel ch3 Linked List Pointer Low Part Register	0x00000000
0x00C0	CHCTRL[CH4][CTRL]	Channel ch4 Control Register	0x00000000
0x00C4	CHCTRL[CH4][TRANSIZE]	Channel ch4Transfer Size Register	0x00000000
0x00C8	CHCTRL[CH4][SRCADDR]	Channel ch4 Source Address Low Part Register	0x00000000
0x00CC	CHCTRL[CH4][CHANREQCTRL]	Channel ch4 DMA Request Control Register	0x00000000
0x00D0	CHCTRL[CH4][DSTADDR]	Channel ch4 Destination Address Low Part Register	0x00000000
0x00D4	CHCTRL[CH4][SWAPTABLE]	swap table register	0x00000000
0x00D8	CHCTRL[CH4][LLPOINTER]	Channel ch4 Linked List Pointer Low Part Register	0x00000000
0x00E0	CHCTRL[CH5][CTRL]	Channel ch5 Control Register	0x00000000

Offset	Register Name	Description	Reset Value
0x00E4	CHCTRL[CH5][TRANSIZE]	Channel ch5Transfer Size Register	0x00000000
0x00E8	CHCTRL[CH5][SRCADDR]	Channel ch5 Source Address Low Part Register	0x00000000
0x00EC	CHCTRL[CH5][CHANREQCTRL]	Channel ch5 DMA Request Control Register	0x00000000
0x00F0	CHCTRL[CH5][DSTADDR]	Channel ch5 Destination Address Low Part Register	0x00000000
0x00F4	CHCTRL[CH5][SWAPTABLE]	swap table register	0x00000000
0x00F8	CHCTRL[CH5][LLPOINTER]	Channel ch5 Linked List Pointer Low Part Register	0x00000000
0x0100	CHCTRL[CH6][CTRL]	Channel ch6 Control Register	0x00000000
0x0104	CHCTRL[CH6][TRANSIZE]	Channel ch6Transfer Size Register	0x00000000
0x0108	CHCTRL[CH6][SRCADDR]	Channel ch6 Source Address Low Part Register	0x00000000
0x010C	CHCTRL[CH6][CHANREQCTRL]	Channel ch6 DMA Request Control Register	0x00000000
0x0110	CHCTRL[CH6][DSTADDR]	Channel ch6 Destination Address Low Part Register	0x00000000
0x0114	CHCTRL[CH6][SWAPTABLE]	swap table register	0x00000000
0x0118	CHCTRL[CH6][LLPOINTER]	Channel ch6 Linked List Pointer Low Part Register	0x00000000
0x0120	CHCTRL[CH7][CTRL]	Channel ch7 Control Register	0x00000000
0x0124	CHCTRL[CH7][TRANSIZE]	Channel ch7Transfer Size Register	0x00000000
0x0128	CHCTRL[CH7][SRCADDR]	Channel ch7 Source Address Low Part Register	0x00000000
0x012C	CHCTRL[CH7][CHANREQCTRL]	Channel ch7 DMA Request Control Register	0x00000000
0x0130	CHCTRL[CH7][DSTADDR]	Channel ch7 Destination Address Low Part Register	0x00000000
0x0134	CHCTRL[CH7][SWAPTABLE]	swap table register	0x00000000
0x0138	CHCTRL[CH7][LLPOINTER]	Channel ch7 Linked List Pointer Low Part Register	0x00000000
0x0140	CHCTRL[CH8][CTRL]	Channel ch8 Control Register	0x00000000
0x0144	CHCTRL[CH8][TRANSIZE]	Channel ch8Transfer Size Register	0x00000000
0x0148	CHCTRL[CH8][SRCADDR]	Channel ch8 Source Address Low Part Register	0x00000000

Offset	Register Name	Description	Reset Value
0x014C	CHCTRL[CH8][CHANREQCTRL]	Channel ch8 DMA Request Control Register	0x00000000
0x0150	CHCTRL[CH8][DSTADDR]	Channel ch8 Destination Address Low Part Register	0x00000000
0x0154	CHCTRL[CH8][SWAPTABLE]	swap table register	0x00000000
0x0158	CHCTRL[CH8][LLPOINTER]	Channel ch8 Linked List Pointer Low Part Register	0x00000000
0x0160	CHCTRL[CH9][CTRL]	Channel ch9 Control Register	0x00000000
0x0164	CHCTRL[CH9][TRANSIZE]	Channel ch9 Transfer Size Register	0x00000000
0x0168	CHCTRL[CH9][SRCADDR]	Channel ch9 Source Address Low Part Register	0x00000000
0x016C	CHCTRL[CH9][CHANREQCTRL]	Channel ch9 DMA Request Control Register	0x00000000
0x0170	CHCTRL[CH9][DSTADDR]	Channel ch9 Destination Address Low Part Register	0x00000000
0x0174	CHCTRL[CH9][SWAPTABLE]	swap table register	0x00000000
0x0178	CHCTRL[CH9][LLPOINTER]	Channel ch9 Linked List Pointer Low Part Register	0x00000000
0x0180	CHCTRL[CH10][CTRL]	Channel ch10 Control Register	0x00000000
0x0184	CHCTRL[CH10][TRANSIZE]	Channel ch10 Transfer Size Register	0x00000000
0x0188	CHCTRL[CH10][SRCADDR]	Channel ch10 Source Address Low Part Register	0x00000000
0x018C	CHCTRL[CH10][CHANREQCTRL]	Channel ch10 DMA Request Control Register	0x00000000
0x0190	CHCTRL[CH10][DSTADDR]	Channel ch10 Destination Address Low Part Register	0x00000000
0x0194	CHCTRL[CH10][SWAPTABLE]	swap table register	0x00000000
0x0198	CHCTRL[CH10][LLPOINTER]	Channel ch10 Linked List Pointer Low Part Register	0x00000000
0x01A0	CHCTRL[CH11][CTRL]	Channel ch11 Control Register	0x00000000
0x01A4	CHCTRL[CH11][TRANSIZE]	Channel ch11 Transfer Size Register	0x00000000
0x01A8	CHCTRL[CH11][SRCADDR]	Channel ch11 Source Address Low Part Register	0x00000000
0x01AC	CHCTRL[CH11][CHANREQCTRL]	Channel ch11 DMA Request Control Register	0x00000000
0x01B0	CHCTRL[CH11][DSTADDR]	Channel ch11 Destination Address Low Part Register	0x00000000
0x01B4	CHCTRL[CH11][SWAPTABLE]	swap table register	0x00000000

Offset	Register Name	Description	Reset Value
0x01B8	CHCTRL[CH11][LLPOINTER]	Channel ch11 Linked List Pointer Low Part Register	0x00000000
0x01C0	CHCTRL[CH12][CTRL]	Channel ch12 Control Register	0x00000000
0x01C4	CHCTRL[CH12][TRANSIZE]	Channel ch12Transfer Size Register	0x00000000
0x01C8	CHCTRL[CH12][SRCADDR]	Channel ch12 Source Address Low Part Register	0x00000000
0x01CC	CHCTRL[CH12][CHANREQCTRL]	Channel ch12 DMA Request Control Register	0x00000000
0x01D0	CHCTRL[CH12][DSTADDR]	Channel ch12 Destination Address Low Part Register	0x00000000
0x01D4	CHCTRL[CH12][SWAPTABLE]	swap table register	0x00000000
0x01D8	CHCTRL[CH12][LLPOINTER]	Channel ch12 Linked List Pointer Low Part Register	0x00000000
0x01E0	CHCTRL[CH13][CTRL]	Channel ch13 Control Register	0x00000000
0x01E4	CHCTRL[CH13][TRANSIZE]	Channel ch13Transfer Size Register	0x00000000
0x01E8	CHCTRL[CH13][SRCADDR]	Channel ch13 Source Address Low Part Register	0x00000000
0x01EC	CHCTRL[CH13][CHANREQCTRL]	Channel ch13 DMA Request Control Register	0x00000000
0x01F0	CHCTRL[CH13][DSTADDR]	Channel ch13 Destination Address Low Part Register	0x00000000
0x01F4	CHCTRL[CH13][SWAPTABLE]	swap table register	0x00000000
0x01F8	CHCTRL[CH13][LLPOINTER]	Channel ch13 Linked List Pointer Low Part Register	0x00000000
0x0200	CHCTRL[CH14][CTRL]	Channel ch14 Control Register	0x00000000
0x0204	CHCTRL[CH14][TRANSIZE]	Channel ch14Transfer Size Register	0x00000000
0x0208	CHCTRL[CH14][SRCADDR]	Channel ch14 Source Address Low Part Register	0x00000000
0x020C	CHCTRL[CH14][CHANREQCTRL]	Channel ch14 DMA Request Control Register	0x00000000
0x0210	CHCTRL[CH14][DSTADDR]	Channel ch14 Destination Address Low Part Register	0x00000000
0x0214	CHCTRL[CH14][SWAPTABLE]	swap table register	0x00000000
0x0218	CHCTRL[CH14][LLPOINTER]	Channel ch14 Linked List Pointer Low Part Register	0x00000000
0x0220	CHCTRL[CH15][CTRL]	Channel ch15 Control Register	0x00000000
0x0224	CHCTRL[CH15][TRANSIZE]	Channel ch15Transfer Size Register	0x00000000

Offset	Register Name	Description	Reset Value
0x0228	CHCTRL[CH15][SRCADDR]	Channel ch15 Source Address Low Part Register	0x00000000
0x022C	CHCTRL[CH15][CHANREQCTRL]	Channel ch15 DMA Request Control Register	0x00000000
0x0230	CHCTRL[CH15][DSTADDR]	Channel ch15 Destination Address Low Part Register	0x00000000
0x0234	CHCTRL[CH15][SWAPTABLE]	swap table register	0x00000000
0x0238	CHCTRL[CH15][LLPOINTER]	Channel ch15 Linked List Pointer Low Part Register	0x00000000
0x0240	CHCTRL[CH16][CTRL]	Channel ch16 Control Register	0x00000000
0x0244	CHCTRL[CH16][TRANSIZE]	Channel ch16 Transfer Size Register	0x00000000
0x0248	CHCTRL[CH16][SRCADDR]	Channel ch16 Source Address Low Part Register	0x00000000
0x024C	CHCTRL[CH16][CHANREQCTRL]	Channel ch16 DMA Request Control Register	0x00000000
0x0250	CHCTRL[CH16][DSTADDR]	Channel ch16 Destination Address Low Part Register	0x00000000
0x0254	CHCTRL[CH16][SWAPTABLE]	swap table register	0x00000000
0x0258	CHCTRL[CH16][LLPOINTER]	Channel ch16 Linked List Pointer Low Part Register	0x00000000
0x0260	CHCTRL[CH17][CTRL]	Channel ch17 Control Register	0x00000000
0x0264	CHCTRL[CH17][TRANSIZE]	Channel ch17 Transfer Size Register	0x00000000
0x0268	CHCTRL[CH17][SRCADDR]	Channel ch17 Source Address Low Part Register	0x00000000
0x026C	CHCTRL[CH17][CHANREQCTRL]	Channel ch17 DMA Request Control Register	0x00000000
0x0270	CHCTRL[CH17][DSTADDR]	Channel ch17 Destination Address Low Part Register	0x00000000
0x0274	CHCTRL[CH17][SWAPTABLE]	swap table register	0x00000000
0x0278	CHCTRL[CH17][LLPOINTER]	Channel ch17 Linked List Pointer Low Part Register	0x00000000
0x0280	CHCTRL[CH18][CTRL]	Channel ch18 Control Register	0x00000000
0x0284	CHCTRL[CH18][TRANSIZE]	Channel ch18 Transfer Size Register	0x00000000
0x0288	CHCTRL[CH18][SRCADDR]	Channel ch18 Source Address Low Part Register	0x00000000
0x028C	CHCTRL[CH18][CHANREQCTRL]	Channel ch18 DMA Request Control Register	0x00000000

Offset	Register Name	Description	Reset Value
0x0290	CHCTRL[CH18][DSTADDR]	Channel ch18 Destination Address Low Part Register	0x00000000
0x0294	CHCTRL[CH18][SWAPTABLE]	swap table register	0x00000000
0x0298	CHCTRL[CH18][LLPOINTER]	Channel ch18 Linked List Pointer Low Part Register	0x00000000
0x02A0	CHCTRL[CH19][CTRL]	Channel ch19 Control Register	0x00000000
0x02A4	CHCTRL[CH19][TRANSIZE]	Channel ch19 Transfer Size Register	0x00000000
0x02A8	CHCTRL[CH19][SRCADDR]	Channel ch19 Source Address Low Part Register	0x00000000
0x02AC	CHCTRL[CH19][CHANREQCTRL]	Channel ch19 DMA Request Control Register	0x00000000
0x02B0	CHCTRL[CH19][DSTADDR]	Channel ch19 Destination Address Low Part Register	0x00000000
0x02B4	CHCTRL[CH19][SWAPTABLE]	swap table register	0x00000000
0x02B8	CHCTRL[CH19][LLPOINTER]	Channel ch19 Linked List Pointer Low Part Register	0x00000000
0x02C0	CHCTRL[CH20][CTRL]	Channel ch20 Control Register	0x00000000
0x02C4	CHCTRL[CH20][TRANSIZE]	Channel ch20 Transfer Size Register	0x00000000
0x02C8	CHCTRL[CH20][SRCADDR]	Channel ch20 Source Address Low Part Register	0x00000000
0x02CC	CHCTRL[CH20][CHANREQCTRL]	Channel ch20 DMA Request Control Register	0x00000000
0x02D0	CHCTRL[CH20][DSTADDR]	Channel ch20 Destination Address Low Part Register	0x00000000
0x02D4	CHCTRL[CH20][SWAPTABLE]	swap table register	0x00000000
0x02D8	CHCTRL[CH20][LLPOINTER]	Channel ch20 Linked List Pointer Low Part Register	0x00000000
0x02E0	CHCTRL[CH21][CTRL]	Channel ch21 Control Register	0x00000000
0x02E4	CHCTRL[CH21][TRANSIZE]	Channel ch21 Transfer Size Register	0x00000000
0x02E8	CHCTRL[CH21][SRCADDR]	Channel ch21 Source Address Low Part Register	0x00000000
0x02EC	CHCTRL[CH21][CHANREQCTRL]	Channel ch21 DMA Request Control Register	0x00000000
0x02F0	CHCTRL[CH21][DSTADDR]	Channel ch21 Destination Address Low Part Register	0x00000000
0x02F4	CHCTRL[CH21][SWAPTABLE]	swap table register	0x00000000
0x02F8	CHCTRL[CH21][LLPOINTER]	Channel ch21 Linked List Pointer Low Part Register	0x00000000

Offset	Register Name	Description	Reset Value
0x0300	CHCTRL[CH22][CTRL]	Channel ch22 Control Register	0x00000000
0x0304	CHCTRL[CH22][TRANSIZE]	Channel ch22Transfer Size Register	0x00000000
0x0308	CHCTRL[CH22][SRCADDR]	Channel ch22 Source Address Low Part Register	0x00000000
0x030C	CHCTRL[CH22][CHANREQCTRL]	Channel ch22 DMA Request Control Register	0x00000000
0x0310	CHCTRL[CH22][DSTADDR]	Channel ch22 Destination Address Low Part Register	0x00000000
0x0314	CHCTRL[CH22][SWAPTABLE]	swap table register	0x00000000
0x0318	CHCTRL[CH22][LLPOINTER]	Channel ch22 Linked List Pointer Low Part Register	0x00000000
0x0320	CHCTRL[CH23][CTRL]	Channel ch23 Control Register	0x00000000
0x0324	CHCTRL[CH23][TRANSIZE]	Channel ch23Transfer Size Register	0x00000000
0x0328	CHCTRL[CH23][SRCADDR]	Channel ch23 Source Address Low Part Register	0x00000000
0x032C	CHCTRL[CH23][CHANREQCTRL]	Channel ch23 DMA Request Control Register	0x00000000
0x0330	CHCTRL[CH23][DSTADDR]	Channel ch23 Destination Address Low Part Register	0x00000000
0x0334	CHCTRL[CH23][SWAPTABLE]	swap table register	0x00000000
0x0338	CHCTRL[CH23][LLPOINTER]	Channel ch23 Linked List Pointer Low Part Register	0x00000000
0x0340	CHCTRL[CH24][CTRL]	Channel ch24 Control Register	0x00000000
0x0344	CHCTRL[CH24][TRANSIZE]	Channel ch24Transfer Size Register	0x00000000
0x0348	CHCTRL[CH24][SRCADDR]	Channel ch24 Source Address Low Part Register	0x00000000
0x034C	CHCTRL[CH24][CHANREQCTRL]	Channel ch24 DMA Request Control Register	0x00000000
0x0350	CHCTRL[CH24][DSTADDR]	Channel ch24 Destination Address Low Part Register	0x00000000
0x0354	CHCTRL[CH24][SWAPTABLE]	swap table register	0x00000000
0x0358	CHCTRL[CH24][LLPOINTER]	Channel ch24 Linked List Pointer Low Part Register	0x00000000
0x0360	CHCTRL[CH25][CTRL]	Channel ch25 Control Register	0x00000000
0x0364	CHCTRL[CH25][TRANSIZE]	Channel ch25Transfer Size Register	0x00000000
0x0368	CHCTRL[CH25][SRCADDR]	Channel ch25 Source Address Low Part Register	0x00000000

Offset	Register Name	Description	Reset Value
0x036C	CHCTRL[CH25][CHANREQCTRL]	Channel ch25 DMA Request Control Register	0x00000000
0x0370	CHCTRL[CH25][DSTADDR]	Channel ch25 Destination Address Low Part Register	0x00000000
0x0374	CHCTRL[CH25][SWAPTABLE]	swap table register	0x00000000
0x0378	CHCTRL[CH25][LLPOINTER]	Channel ch25 Linked List Pointer Low Part Register	0x00000000
0x0380	CHCTRL[CH26][CTRL]	Channel ch26 Control Register	0x00000000
0x0384	CHCTRL[CH26][TRANSIZE]	Channel ch26 Transfer Size Register	0x00000000
0x0388	CHCTRL[CH26][SRCADDR]	Channel ch26 Source Address Low Part Register	0x00000000
0x038C	CHCTRL[CH26][CHANREQCTRL]	Channel ch26 DMA Request Control Register	0x00000000
0x0390	CHCTRL[CH26][DSTADDR]	Channel ch26 Destination Address Low Part Register	0x00000000
0x0394	CHCTRL[CH26][SWAPTABLE]	swap table register	0x00000000
0x0398	CHCTRL[CH26][LLPOINTER]	Channel ch26 Linked List Pointer Low Part Register	0x00000000
0x03A0	CHCTRL[CH27][CTRL]	Channel ch27 Control Register	0x00000000
0x03A4	CHCTRL[CH27][TRANSIZE]	Channel ch27 Transfer Size Register	0x00000000
0x03A8	CHCTRL[CH27][SRCADDR]	Channel ch27 Source Address Low Part Register	0x00000000
0x03AC	CHCTRL[CH27][CHANREQCTRL]	Channel ch27 DMA Request Control Register	0x00000000
0x03B0	CHCTRL[CH27][DSTADDR]	Channel ch27 Destination Address Low Part Register	0x00000000
0x03B4	CHCTRL[CH27][SWAPTABLE]	swap table register	0x00000000
0x03B8	CHCTRL[CH27][LLPOINTER]	Channel ch27 Linked List Pointer Low Part Register	0x00000000
0x03C0	CHCTRL[CH28][CTRL]	Channel ch28 Control Register	0x00000000
0x03C4	CHCTRL[CH28][TRANSIZE]	Channel ch28 Transfer Size Register	0x00000000
0x03C8	CHCTRL[CH28][SRCADDR]	Channel ch28 Source Address Low Part Register	0x00000000
0x03CC	CHCTRL[CH28][CHANREQCTRL]	Channel ch28 DMA Request Control Register	0x00000000
0x03D0	CHCTRL[CH28][DSTADDR]	Channel ch28 Destination Address Low Part Register	0x00000000
0x03D4	CHCTRL[CH28][SWAPTABLE]	swap table register	0x00000000

Offset	Register Name	Description	Reset Value
0x03D8	CHCTRL[CH28][LLPOINTER]	Channel ch28 Linked List Pointer Low Part Register	0x00000000
0x03E0	CHCTRL[CH29][CTRL]	Channel ch29 Control Register	0x00000000
0x03E4	CHCTRL[CH29][TRANSIZE]	Channel ch29 Transfer Size Register	0x00000000
0x03E8	CHCTRL[CH29][SRCADDR]	Channel ch29 Source Address Low Part Register	0x00000000
0x03EC	CHCTRL[CH29][CHANREQCTRL]	Channel ch29 DMA Request Control Register	0x00000000
0x03F0	CHCTRL[CH29][DSTADDR]	Channel ch29 Destination Address Low Part Register	0x00000000
0x03F4	CHCTRL[CH29][SWAPTABLE]	swap table register	0x00000000
0x03F8	CHCTRL[CH29][LLPOINTER]	Channel ch29 Linked List Pointer Low Part Register	0x00000000
0x0400	CHCTRL[CH30][CTRL]	Channel ch30 Control Register	0x00000000
0x0404	CHCTRL[CH30][TRANSIZE]	Channel ch30 Transfer Size Register	0x00000000
0x0408	CHCTRL[CH30][SRCADDR]	Channel ch30 Source Address Low Part Register	0x00000000
0x040C	CHCTRL[CH30][CHANREQCTRL]	Channel ch30 DMA Request Control Register	0x00000000
0x0410	CHCTRL[CH30][DSTADDR]	Channel ch30 Destination Address Low Part Register	0x00000000
0x0414	CHCTRL[CH30][SWAPTABLE]	swap table register	0x00000000
0x0418	CHCTRL[CH30][LLPOINTER]	Channel ch30 Linked List Pointer Low Part Register	0x00000000
0x0420	CHCTRL[CH31][CTRL]	Channel ch31 Control Register	0x00000000
0x0424	CHCTRL[CH31][TRANSIZE]	Channel ch31 Transfer Size Register	0x00000000
0x0428	CHCTRL[CH31][SRCADDR]	Channel ch31 Source Address Low Part Register	0x00000000
0x042C	CHCTRL[CH31][CHANREQCTRL]	Channel ch31 DMA Request Control Register	0x00000000
0x0430	CHCTRL[CH31][DSTADDR]	Channel ch31 Destination Address Low Part Register	0x00000000
0x0434	CHCTRL[CH31][SWAPTABLE]	swap table register	0x00000000
0x0438	CHCTRL[CH31][LLPOINTER]	Channel ch31 Linked List Pointer Low Part Register	0x00000000

Table 166: DMAV2 Register List

33.6 DMA Register Details

The detailed description of DMA registers is as follows:

33.6.1 IDMISC (0x4)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																DMASTATE		CURCHAN				RSVD									
N/A																RO		RO				N/A									
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	

IDMISC [31:0]

Bits	Name	Description
15-13	DMASTATE	DMA state machine localparam ST_IDLE = 3'b000; localparam ST_READ = 3'b001; localparam ST_READ_ACK = 3'b010; localparam ST_WRITE = 3'b011; localparam ST_WRITE_ACK = 3'b100; localparam ST_LL = 3'b101; localparam ST_END = 3'b110; localparam ST_END_WAIT = 3'b111;
12-8	CURCHAN	current channel in used

IDMISC Bits

33.6.2 DMACFG (0x10)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHAINXFR	REQSYNC	RSVD				DATAWIDTH		ADDRWIDTH								CORENUM	BUSNUM	REQNUM				FIFODEPTH				CHANNELNUM					
RO	RO	N/A				RO		RO								RO	RO	RO				RO				RO					
0	0	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMACFG [31:0]

Bits	Name	Description
31	CHAINXFR	Chain transfer 0x0: Chain transfer is not configured 0x1: Chain transfer is configured

Bits	Name	Description
30	REQSYNC	DMA request synchronization. The DMA request synchronization should be configured to avoid signal integrity problems when the request signal is not clocked by the system bus clock, which the DMA control logic operates in. If the request synchronization is not configured, the request signal is sampled directly without synchronization. 0x0: Request synchronization is not configured 0x1: Request synchronization is configured
25-24	DATAWIDTH	AXI bus data width 0x0: 32 bits 0x1: 64 bits 0x2: 128 bits 0x3: 256 bits
23-17	ADDRWIDTH	AXI bus address width 0x18: 24 bits 0x19: 25 bits ... 0x40: 64 bits Others: Invalid
16	CORENUM	DMA core number 0x0: 1 core 0x1: 2 cores
15	BUSNUM	AXI bus interface number 0x0: 1 AXI bus 0x1: 2 AXI busses
14-10	REQNUM	Request/acknowledge pair number 0x0: 0 pair 0x1: 1 pair 0x2: 2 pairs ... 0x10: 16 pairs
9-4	FIFODEPTH	FIFO depth 0x4: 4 entries 0x8: 8 entries 0x10: 16 entries 0x20: 32 entries Others: Invalid

Bits	Name	Description
3-0	CHANNELNUM	Channel number 0x1: 1 channel 0x2: 2 channels ... 0x8: 8 channels Others: Invalid

DMACFG Bits

33.6.3 DMACTRL (0x14)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																RESET															
N/A																WO															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0

DMACTRL [31:0]

Bits	Name	Description
0	RESET	Software reset control. Write 1 to this bit to reset the DMA core and disable all channels. Note: The software reset may cause the in-completion of AXI transaction.

DMACTRL Bits

33.6.4 CHABORT (0x18)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHABORT																WO															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CHABORT [31:0]

Bits	Name	Description
31-0	CHABORT	Write 1 to bit n to abort channel n. The bits should only be set when the corresponding channels are enabled. Otherwise, the writes will be ignored for channels that are not enabled. (N: Number of channels)

Bits	Name	Description

CHABORT Bits

33.6.5 INTHALFSTS (0x24)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STS																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

INTHALFSTS [31:0]

Bits	Name	Description
31-0	STS	half transfer done irq status

INTHALFSTS Bits

33.6.6 INTTCSTS (0x28)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STS																W1C															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

INTTCSTS [31:0]

Bits	Name	Description
31-0	STS	The terminal count status, one bit per channel. The terminal count status is set when a channel transfer finishes without the abort or error event. 0x0: Channel n has no terminal count status 0x1: Channel n has terminal count status

INTTCSTS Bits

33.6.7 INTABORTSTS (0x2C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STS																W1C															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

INTABORTSTS [31:0]

Bits	Name	Description
31-0	STS	The abort status of channel, one bit per channel. The abort status is set when a channel transfer is aborted. 0x0: Channel n has no abort status 0x1: Channel n has abort status

INTABORTSTS Bits

33.6.8 INTERRSTS (0x30)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STS																WIC															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

INTERRSTS [31:0]

Bits	Name	Description
31-0	STS	The error status, one bit per channel. The error status is set when a channel transfer encounters the following error events: - Bus error - Unaligned address - Unaligned transfer width - Reserved configuration 0x0: Channel n has no error status 0x1: Channel n has error status

INTERRSTS Bits

33.6.9 CHEN (0x34)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHEN																RO															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CHEN [31:0]

Bits	Name	Description
31-0	CHEN	Alias of the Enable field of all ChnCtrl registers

CHEN Bits

33.6.10 CHCTRL[CTRL] (0x40 + 0x20 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INFINITELOOP	HANDSHAKEOPT	PRIORITY	BURSTOPT	SRCBURSTSIZE				SRCWIDTH			DSTWIDTH			SRCMODE	DSTMODE	SRCADDRCTRL		DSTADDRCTRL		RSVD	SRC_FIXBURST	RSVD	DST_FIXBURST	SWAP_CTL		RSVD	INTHALFCNTMASK	INTABTMASK	INTERRMASK	INTTCMASK	ENABLE
RW	RW	RW	RW	RW				RW			RW			RW	RW	RW		RW		N/A	RW	N/A	RW	RW		N/A	RW	RW	RW	RW	RW
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	0	x	0	0	0	x	0	0	0	0	0

CHCTRL[CTRL] [31:0]

Bits	Name	Description
31	INFINITELOOP	set to loop current config infinitely
30	HANDSHAKEOPT	0: one request to transfer one burst 1: one request to transfer all the data defined in ch_tts
29	PRIORITY	Channel priority level 0x0: Lower priority 0x1: Higher priority
28	BURSTOPT	set to change burst_size definition
27-24	SRCBURSTSIZE	Source burst size. This field indicates the number of transfers before DMA channel re-arbitration. The burst transfer byte number is (SrcBurstSize * SrcWidth). 0x0: 1 transfer 0x1: 2 transfers 0x2: 4 transfers 0x3: 8 transfers 0x4: 16 transfers 0x5: 32 transfers 0x6: 64 transfers 0x7: 128 transfers 0x8: 256 transfers 0x9: 512 transfers 0xa: 1024 transfers 0xb - 0xf: Reserved, setting this field with a reserved value triggers the error exception

Bits	Name	Description
23-21	SRCWIDTH	Source transfer width 0x0: Byte transfer 0x1: Half-word transfer 0x2: Word transfer 0x3: Double word transfer 0x4: Quad word transfer 0x5: Eight word transfer 0x6 - 0x7: Reserved, setting this field with a reserved value triggers the error exception
20-18	DSTWIDTH	Destination transfer width. Both the total transfer byte number and the burst transfer byte number should be aligned to the destination transfer width; otherwise the error event will be triggered. For example, destination transfer width should be set as byte transfer if total transfer byte is not aligned to half-word. See field SrcBurstSize above for the definition of burst transfer byte number and section 3.2.8 for the definition of the total transfer byte number. 0x0: Byte transfer 0x1: Half-word transfer 0x2: Word transfer 0x3: Double word transfer 0x4: Quad word transfer 0x5: Eight word transfer 0x6 - 0x7: Reserved, setting this field with a reserved value triggers the error exception
17	SRCMODE	Source DMA handshake mode 0x0: Normal mode 0x1: Handshake mode Normal mode is enabled and started by software set Enable bit; Handshake mode is enabled by software set Enable bit, started by hardware dma request from DMAMUX block

Bits	Name	Description
16	DSTMODE	Destination DMA handshake mode 0x0: Normal mode 0x1: Handshake mode the difference between Source/Destination handshake mode is: the dma block will response hardware request after read in Source handshake mode; the dma block will response hardware request after write in Destination handshake mode; NOTE: can't set SrcMode and DstMode at same time, otherwise result unknown.
15-14	SRCADDRCTRL	Source address control 0x0: Increment address 0x1: Decrement address 0x2: Fixed address 0x3: Reserved, setting the field with this value triggers the error exception
13-12	DSTADDRCTRL	Destination address control 0x0: Increment address 0x1: Decrement address 0x2: Fixed address 0x3: Reserved, setting the field with this value triggers the error exception
10	SRC_FIXBURST	set to loop current burst, omit setting in src_addr_ctrl
8	DST_FIXBURST	set to loop current burst, omit setting in dst_addr_ctrl
7-6	SWAP_CTL	00 : use swap table(if swap table all 0, then act nothing) 01 : byte swap 10 : 16bit swap 11 : 32bit swap
4	INTHALFCNTMASK	Channel half interrupt mask 0x0: Allow the half interrupt to be triggered 0x1: Disable the half interrupt
3	INTABTMASK	Channel abort interrupt mask 0x0: Allow the abort interrupt to be triggered 0x1: Disable the abort interrupt
2	INTERRMASK	Channel error interrupt mask 0x0: Allow the error interrupt to be triggered 0x1: Disable the error interrupt
1	INTTCMASK	Channel terminal count interrupt mask 0x0: Allow the terminal count interrupt to be triggered 0x1: Disable the terminal count interrupt

Bits	Name	Description
0	ENABLE	Channel enable bit 0x0: Disable 0x1: Enable

CHCTRL[CTRL] Bits

33.6.11 CHCTRL[TRANSIZE] (0x44 + 0x20 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD				TRANSIZE																											
N/A				RW																											
x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CHCTRL[TRANSIZE] [31:0]

Bits	Name	Description
27-0	TRANSIZE	Total transfer size from source. The total number of transferred bytes is (TranSize * SrcWidth). This register is cleared when the DMA transfer is done. If a channel is enabled with zero total transfer size, the error event will be triggered and the transfer will be terminated.

CHCTRL[TRANSIZE] Bits

33.6.12 CHCTRL[SRCADDR] (0x48 + 0x20 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRCADDRRL																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CHCTRL[SRCADDR] [31:0]

Bits	Name	Description
31-0	SRCADDRRL	Low part of the source starting address. When the transfer completes, the value of {SrcAddrH,SrcAddrL} is updated to the ending address. This address must be aligned to the source transfer size; otherwise, an error event will be triggered.

Bits	Name	Description

CHCTRL[SRCADDR] Bits

33.6.13 CHCTRL[CHANREQCTRL] (0x4C + 0x20 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD			SRCREQSEL					RSVD			DSTREQSEL				RSVD																
N/A			RW					N/A			RW				N/A																
x	x	x	0	0	0	0	0	x	x	x	0	0	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

CHCTRL[CHANREQCTRL] [31:0]

Bits	Name	Description
28-24	SRCREQSEL	Source DMA request select. Select the request/ack handshake pair that the source device is connected to.
20-16	DSTREQSEL	Destination DMA request select. Select the request/ack handshake pair that the destination device is connected to.

CHCTRL[CHANREQCTRL] Bits

33.6.14 CHCTRL[DSTADDR] (0x50 + 0x20 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSTADDRL																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CHCTRL[DSTADDR] [31:0]

Bits	Name	Description
31-0	DSTADDRL	Low part of the destination starting address. When the transfer completes, the value of {DstAddrH,DstAddrL} is updated to the ending address. This address must be aligned to the destination transfer size; otherwise the error event will be triggered.

CHCTRL[DSTADDR] Bits

33.6.15 CHCTRL[SWAPTABLE] (0x54 + 0x20 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TABLE																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CHCTRL[SWAPTABLE] [31:0]

Bits	Name	Description
31-0	TABLE	"4bit for one byte, define how to swap, default is no swap. For HDMA, support up to 4-byte, the register default value is 0x3210"

CHCTRL[SWAPTABLE] Bits

33.6.16 CHCTRL[LLPOINTER] (0x58 + 0x20 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LLPOINTERL																													RSVD		
RW																													N/A		
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x	x

CHCTRL[LLPOINTER] [31:0]

Bits	Name	Description
31-3	LLPOINTERL	Low part of the pointer to the next descriptor. The pointer must be double word aligned.

CHCTRL[LLPOINTER] Bits

33.7 DMA Configuration and Usage Instructions

Taking starting a DMA handshake mode transfer as an example, the required steps are:

1. Configure DMAMUX to route srcreqsel or dstreqsel to the corresponding channel;
2. Configure the transize register for the corresponding channel, where the total data amount is transize x srcwidth;
3. Configure the srcaddr register for the corresponding channel, pointing to the start address of the source data;
4. Configure the chanreqctl register for the corresponding channel, including source data DMA request selection srcreqsel or destination data DMA request selection dstreqsel;
5. Configure the dstaddr register for the corresponding channel, pointing to the start address of the target data;

6. For non-chained transfer, configure llpointer to 0x0; otherwise, configure it as the start address of the next chained data structure;
7. Configure the ctrl register for the corresponding channel, including the priority field, source burst length srcburstsize field, source data width srcwidth, destination data width dstwidth fields, and the channel enable field enable;
8. Read the register inttcsts to determine if the transfer for the corresponding channel is complete;

34 DMA Request Router DMAMUX

This chapter describes the functions and features of the DMA request router DMAMUX.

For all the peripherals that support generating DMA requests, as well as the allocation details of these peripherals on the DMA request router DMAMUX, please refer to the "DMA Request Router" section.

34.1 Feature Summary

This chapter describes the main features of the DMA request router DMAMUX:

- supports up to 128 peripheral DMA requests
- supports 16 output channels

34.2 Function Description

DMA request router is to realize flexible and dynamic mapping of peripheral DMA requests to each channel of the DMA controller, so that any peripheral can send DMA requests to any channel of any DMA controller.

The steps to configure DMAMUX through the configuration register CHCFGx of each channel of DMAMUX are as follows:

- Select a DMA request from a peripheral via the **SOURCE** bit of **MUXCFGx**
- **ENABLE** bit of **MUXCFGx** is 1, opening channel x of DMAMUX

34.3 DMAMUX Register

DMAMUX is as follows:

DMAMUX base address: 0xF00C4000

Offset	Register Name	Description	Reset Value
0x0000	MUXCFG[HDMA_MUX0]	HDMA MUX0 Configuration	0x00000000
0x0004	MUXCFG[HDMA_MUX1]	HDMA MUX1 Configuration	0x00000000
0x0008	MUXCFG[HDMA_MUX2]	HDMA MUX2 Configuration	0x00000000
0x000C	MUXCFG[HDMA_MUX3]	HDMA MUX3 Configuration	0x00000000
0x0010	MUXCFG[HDMA_MUX4]	HDMA MUX4 Configuration	0x00000000
0x0014	MUXCFG[HDMA_MUX5]	HDMA MUX5 Configuration	0x00000000
0x0018	MUXCFG[HDMA_MUX6]	HDMA MUX6 Configuration	0x00000000
0x001C	MUXCFG[HDMA_MUX7]	HDMA MUX7 Configuration	0x00000000
0x0020	MUXCFG[HDMA_MUX8]	HDMA MUX8 Configuration	0x00000000
0x0024	MUXCFG[HDMA_MUX9]	HDMA MUX9 Configuration	0x00000000
0x0028	MUXCFG[HDMA_MUX10]	HDMA MUX10 Configuration	0x00000000
0x002C	MUXCFG[HDMA_MUX11]	HDMA MUX11 Configuration	0x00000000
0x0030	MUXCFG[HDMA_MUX12]	HDMA MUX12 Configuration	0x00000000
0x0034	MUXCFG[HDMA_MUX13]	HDMA MUX13 Configuration	0x00000000
0x0038	MUXCFG[HDMA_MUX14]	HDMA MUX14 Configuration	0x00000000
0x003C	MUXCFG[HDMA_MUX15]	HDMA MUX15 Configuration	0x00000000

Offset	Register Name	Description	Reset Value
0x0040	MUXCFG[HDMA_MUX16]	HDMA MUX16 Configuration	0x00000000
0x0044	MUXCFG[HDMA_MUX17]	HDMA MUX17 Configuration	0x00000000
0x0048	MUXCFG[HDMA_MUX18]	HDMA MUX18 Configuration	0x00000000
0x004C	MUXCFG[HDMA_MUX19]	HDMA MUX19 Configuration	0x00000000
0x0050	MUXCFG[HDMA_MUX20]	HDMA MUX20 Configuration	0x00000000
0x0054	MUXCFG[HDMA_MUX21]	HDMA MUX21 Configuration	0x00000000
0x0058	MUXCFG[HDMA_MUX22]	HDMA MUX22 Configuration	0x00000000
0x005C	MUXCFG[HDMA_MUX23]	HDMA MUX23 Configuration	0x00000000
0x0060	MUXCFG[HDMA_MUX24]	HDMA MUX24 Configuration	0x00000000
0x0064	MUXCFG[HDMA_MUX25]	HDMA MUX25 Configuration	0x00000000
0x0068	MUXCFG[HDMA_MUX26]	HDMA MUX26 Configuration	0x00000000
0x006C	MUXCFG[HDMA_MUX27]	HDMA MUX27 Configuration	0x00000000
0x0070	MUXCFG[HDMA_MUX28]	HDMA MUX28 Configuration	0x00000000
0x0074	MUXCFG[HDMA_MUX29]	HDMA MUX29 Configuration	0x00000000
0x0078	MUXCFG[HDMA_MUX30]	HDMA MUX30 Configuration	0x00000000
0x007C	MUXCFG[HDMA_MUX31]	HDMA MUX31 Configuration	0x00000000
0x0080	MUXCFG[XDMA_MUX0]	XDMA MUX0 Configuration	0x00000000
0x0084	MUXCFG[XDMA_MUX1]	XDMA MUX1 Configuration	0x00000000
0x0088	MUXCFG[XDMA_MUX2]	XDMA MUX2 Configuration	0x00000000
0x008C	MUXCFG[XDMA_MUX3]	XDMA MUX3 Configuration	0x00000000
0x0090	MUXCFG[XDMA_MUX4]	XDMA MUX4 Configuration	0x00000000
0x0094	MUXCFG[XDMA_MUX5]	XDMA MUX5 Configuration	0x00000000
0x0098	MUXCFG[XDMA_MUX6]	XDMA MUX6 Configuration	0x00000000
0x009C	MUXCFG[XDMA_MUX7]	XDMA MUX7 Configuration	0x00000000
0x00A0	MUXCFG[XDMA_MUX8]	XDMA MUX8 Configuration	0x00000000
0x00A4	MUXCFG[XDMA_MUX9]	XDMA MUX9 Configuration	0x00000000
0x00A8	MUXCFG[XDMA_MUX10]	XDMA MUX10 Configuration	0x00000000
0x00AC	MUXCFG[XDMA_MUX11]	XDMA MUX11 Configuration	0x00000000
0x00B0	MUXCFG[XDMA_MUX12]	XDMA MUX12 Configuration	0x00000000
0x00B4	MUXCFG[XDMA_MUX13]	XDMA MUX13 Configuration	0x00000000
0x00B8	MUXCFG[XDMA_MUX14]	XDMA MUX14 Configuration	0x00000000
0x00BC	MUXCFG[XDMA_MUX15]	XDMA MUX15 Configuration	0x00000000
0x00C0	MUXCFG[XDMA_MUX16]	XDMA MUX16 Configuration	0x00000000
0x00C4	MUXCFG[XDMA_MUX17]	XDMA MUX17 Configuration	0x00000000
0x00C8	MUXCFG[XDMA_MUX18]	XDMA MUX18 Configuration	0x00000000
0x00CC	MUXCFG[XDMA_MUX19]	XDMA MUX19 Configuration	0x00000000
0x00D0	MUXCFG[XDMA_MUX20]	XDMA MUX20 Configuration	0x00000000
0x00D4	MUXCFG[XDMA_MUX21]	XDMA MUX21 Configuration	0x00000000
0x00D8	MUXCFG[XDMA_MUX22]	XDMA MUX22 Configuration	0x00000000
0x00DC	MUXCFG[XDMA_MUX23]	XDMA MUX23 Configuration	0x00000000

Offset	Register Name	Description	Reset Value
0x00E0	MUXCFG[XDMA_MUX24]	XDMA MUX24 Configuration	0x00000000
0x00E4	MUXCFG[XDMA_MUX25]	XDMA MUX25 Configuration	0x00000000
0x00E8	MUXCFG[XDMA_MUX26]	XDMA MUX26 Configuration	0x00000000
0x00EC	MUXCFG[XDMA_MUX27]	XDMA MUX27 Configuration	0x00000000
0x00F0	MUXCFG[XDMA_MUX28]	XDMA MUX28 Configuration	0x00000000
0x00F4	MUXCFG[XDMA_MUX29]	XDMA MUX29 Configuration	0x00000000
0x00F8	MUXCFG[XDMA_MUX30]	XDMA MUX30 Configuration	0x00000000
0x00FC	MUXCFG[XDMA_MUX31]	XDMA MUX31 Configuration	0x00000000

Table 167: DMAMUX Register List

34.4 DMAMUX register details

DMAMUX is as follows:

34.4.1 MUXCFG (0x0 + 0x4 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE	RSVD																									SOURCE					
WO	N/A																									WO					
0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0

MUXCFG [31:0]

Bits	Name	Description
31	ENABLE	DMA Mux Channel Enable Enables the channel for DMA Mux. The DMA has separate channel enables/disables, which should be used to disable or reconfigure a DMA channel. 0b - DMA Mux channel is disabled 1b - DMA Mux channel is enabled
6-0	SOURCE	DMA Channel Source Specifies which DMA source, if any, is routed to a particular DMA channel. See the "DMA MUX Mapping"

MUXCFG Bits

35 Communication Mailbox MBX

This chapter describes the functions and features of the communication mailbox MBX.

35.1 Feature Summary

communication mailbox MBX supports inter-processor core communication, as well as inter-process communication between processors. The main characteristics are as follows:

- supports 2 register access interfaces
- Each interface supports TX FIFO and RX FIFO
- support flag bits reflect TX FIFO and RX FIFO status
- supports generation of interrupts

35.2 Function Description

This section describes the functions of the communication mailbox MBX.

35.2.1 MBX register access interface

communication mailbox MBX has two sets of register access interfaces, interface A and interface B. Both the A and B interfaces have a set of TX FIFO registers, RX FIFO registers, control registers, and status registers. The data sent by the

user from the transmitter TX of the A interface can be received at the receiver RX of the B interface. Similarly, the receiving end RX of the A interface can receive data sent by the transmitting end TX of the B interface.

35.2.2 FIFO Management

MBX is 4 words (4×32 bits), and the write interface of the TX FIFO is the TXREG register. MBX supports prompts for various states of the TX FIFO, namely whether the TX FIFO is empty, whether the TX FIFO is full, and the number of words empty in the TX FIFO. The receiver RX FIFO depth of

MBX is 4 words (4×32 bits). The read interface of the RX FIFO is the RXREG register. MBX supports prompts for various states of the RX FIFO, namely whether the RX FIFO is empty, whether the RX FIFO is full, and the number of valid words in the RX FIFO.

35.2.3 interrupt

MBX supports generating the following interrupts:

- TX FIFO is empty.
- An interrupt can be triggered when the TX FIFO is not full.
- Interrupts can be triggered when the RX FIFO is not empty
- Interrupts can be triggered when the RX FIFO is full
- An interrupt can be triggered when the content of the message word sent is empty, that is, it has been read
- An interrupt can be triggered when the content of the received message word is valid

MBX can generate two interrupts, corresponding to port A and port B. It can be assigned to different cores in a multi-core system or to different processes in a single-core system.

35.3 MBX register

MBX is as follows:

MBX0A base address: 0xF00A0000

MBX0B base address: 0xF00A4000

Offset	Register Name	Description	Reset Value
0x0000	CR	Command Registers	0x00000000
0x0004	SR	Status Registers	0x000000E2
0x0008	TXREG	Transmit word message to other core.	0x00000000
0x000C	RXREG	Receive word message from other core.	0x00000000
0x0010	TXWRD[TXFIFO0]	TXFIFO for sending message to other core	0x00000000
0x0020	RXWRD[RXFIFO0]	RXFIFO for receiving message from other core	0x00000000

Table 168: MBX Register List

35.4 MBX register details

MBX are as follows:

35.4.1 CR (0x0)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXRESET	RSVD															BARCTL	RSVD					BEIE	TFMAIE	TFMEIE	RFMAIE	RFMFIE	RSVD		TWMEIE	RWMVIE	
RW	N/A															RW	N/A					RW	RW	RW	RW	RW	N/A		RW	RW	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CR [31:0]

Bits	Name	Description
31	TXRESET	Reset TX Fifo and word.

Bits	Name	Description
15-14	BARCTL	Bus Access Response Control, when bit 15:14= 00: no bus error will be generated, no wait for fifo write when fifo full and no wait for word/fifo read when word message invalid or fifo empty; or when write to word/fifo message will be ignored. 01: bus error will be generated when: 1, access invalid address; 2, write to ready only addr; 3, write to full fifo or valid message; 4, read from a emptied fifo/word message. 10: no error will be generated, but bus will wait when 1, write to full fifo/reg message; 2, read from a emptied fifo/reg message; write to word message will overwrite the existing reg value even word message are still valid; read from invalid word message will read out last read out message data.happen. 11: reserved.
8	BEIE	Bus Error Interrupt Enable, will enable the interrupt for any bus error as described in the SR bit 13 to bit 8. 1, enable the bus access error interrupt. 0, disable the bus access error interrupt.
7	TFMAIE	TX FIFO message available interrupt enable. 1, enable the TX FIFO message available interrupt. 0, disable the TX FIFO message available interrupt.
6	TFMEIE	TX FIFO message empty interrupt enable. 1, enable the TX FIFO message empty interrupt. 0, disable the TX FIFO message empty interrupt.
5	RFMAIE	RX FIFO message available interrupt enable. 1, enable the RX FIFO message available interrupt. 0, disable the RX FIFO message available interrupt.
4	RFMFIE	RX fifo message full interrupt enable. 1, enable the RX fifo message full interrupt. 0, disable the RX fifo message full interrupt.
1	TWMEIE	TX word message empty interrupt enable. 1, enable the TX word message empty interrupt. 0, disable the TX word message empty interrupt.
0	RWMVIE	RX word message valid interrupt enable. 1, enable the RX word message valid interrupt. 0, disable the RX word message valid interrupt.

CR Bits

35.4.2 SR (0x4)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								RFVC				TFEC				RSVD		ERRRE	EWTRF	ERRFE	EWTFE	EAIVA	EW2RO	TFMA	TFME	RFMA	RFMF	RSVD		TWME	RWMV
N/A								RO				RO				N/A		W1C	W1C	W1C	W1C	W1C	W1C	RW	RW	RO	RO	N/A		RO	RO
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x	0	0	0	0	0	0	1	1	1	0	0	0	1	0

SR [31:0]

Bits	Name	Description
23-20	RFVC	RX FIFO valid message count
19-16	TFEC	TX FIFO empty message word count
13	ERRRE	bus Error for read when rx word message are still invalid, this bit is W1C bit. 1, read from word message when the word message are still invalid will cause this error bit set. 0, nothis kind of bus error; write this bit to 1 will clear this bit when this kind of error happen.
12	EWTRF	bus Error for write when tx word message are still valid, this bit is W1C bit. 1, write to word message when the word message are still valid will cause this error bit set. 0, nothis kind of bus error; write this bit to 1 will clear this bit when this kind of error happen.
11	ERRFE	bus Error for read when rx fifo empty, this bit is W1C bit. 1, read from a empty rx fifo will cause this error bit set. 0, nothis kind of bus error; write this bit to 1 will clear this bit when this kind of error happen.
10	EWTFE	bus Error for write when tx fifo full, this bit is W1C bit. 1, write to a full tx fifo will cause this error bit set. 0, nothis kind of bus error; write this bit to 1 will clear this bit when this kind of error happen.
9	EAIVA	bus Error for Accessing Invalid Address; this bit is W1C bit. 1, read and write to invalid address in the bus of this block, will set this bit. 0, nothis kind of bus error; write this bit to 1 will clear this bit when this kind of error happen.
8	EW2RO	bus Error for Write to Read Only address; this bit is W1C bit. 1, write to read only address happened in the bus of this block. 0, nothis kind of bus error; write this bit to 1 will clear this bit when this kind of error happen.

Bits	Name	Description
7	TFMA	TX FIFO Message slot available, the 4x32 TX FIFO message buffer to the other core full, will not trigger any interrupt. 1, TXFIFO message buffer has slot available 0, no slot available (fifo full)
6	TFME	TX FIFO Message Empty, no any data in the message FIFO buffer from other core, will not trigger any interrupt.message from other core. 1, no any message data in TXFIFO from other core. 0, there are some data in the 4x32 TX FIFO from other core yet.
5	RFMA	RX FIFO Message Available, available data in the 4x32 TX FIFO message buffer to the other core, will trigger interrupt if the related interrupt enable bit set in the control (CR) register. 1, no any data in the 4x32 TXFIFO message buffer. 0, there are some data in the the 4x32 TXFIFO message buffer already.
4	RFMF	RX FIFO Message Full, message from other core; will trigger interrupt if the related interrupt enable bit set in the control (CR) register. 1, the other core had written 4x32 message in the RXFIFO. 0, no 4x32 RX FIFO message from other core yet.
1	TWME	TX word message empty, will trigger interrupt if the related interrupt enable bit set in the control (CR) register. 1, means this core had write word message to TXREG. 0, means no valid word message in the TXREG yet.
0	RWMV	RX word message valid, will trigger interrupt if the related interrupt enable bit set in the control (CR) register. 1, the other core had written word message in the RXREG. 0, no valid word message yet in the RXREG.

SR Bits

35.4.3 TXREG (0x8)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXREG																															
WO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TXREG [31:0]

Bits	Name	Description
31-0	TXREG	Transmit word message to other core.

TXREG Bits

35.4.4 RXREG (0xC)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXREG																RO															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RXREG [31:0]

Bits	Name	Description
31-0	RXREG	Receive word message from other core.

RXREG Bits

35.4.5 TXWRD (0x10 + 0x4 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXFIFO																WO															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TXWRD [31:0]

Bits	Name	Description
31-0	TXFIFO	TXFIFO for sending message to other core, FIFO size, 4x32 can write one of the word address to push data to the FIFO; can also use 4x32 burst write from 0x010 to push 4 words to the FIFO.

TXWRD Bits

35.4.6 RXWRD (0x20 + 0x4 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXFIFO																RO															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RXWRD [31:0]

Bits	Name	Description
31-0	RXFIFO	RXFIFO for receiving message from other core, FIFO size, 4x32 can read one of the word address to pop data to the FIFO; can also use 4x32 burst read from 0x020 to read 4 words from the FIFO.

RXWRD Bits

36 Electric Control System Overview

This chapter introduces the electric control system (EMDS) of this product. The electric control system of this product includes 2 PWM timers, 2 quadrature encoder inputs QEI, 2 quadrature encoder outputs QEO, 1 $\Sigma\Delta$ signal receiving unit, and also includes the programmable logic unit PLB and interconnection manager TRGM.

electric control system can be used in conjunction with on-chip analog peripherals such as analog-to-digital converter ADC, comparator ACMP, etc.

includes a synchronization timer to provide timestamps to the motorized control system.

All motion control units and synchronous timers work on the external bus clock (CLK_TOP_AHB0) to achieve minimum delay.

Before using any module of the electric control system, you should first open the EMDS resource node to enable the electric control system clock.

36.1 PWM timer PWMV2

This product supports 2 PWM timers. Each PWM supports generating 8 independent or 4 pairs of complementary PWM outputs, and supports dead zone control and fault protection.

On this product, the connection relationship between the PWM module and other parts of the system is as shown in Figure 27 :

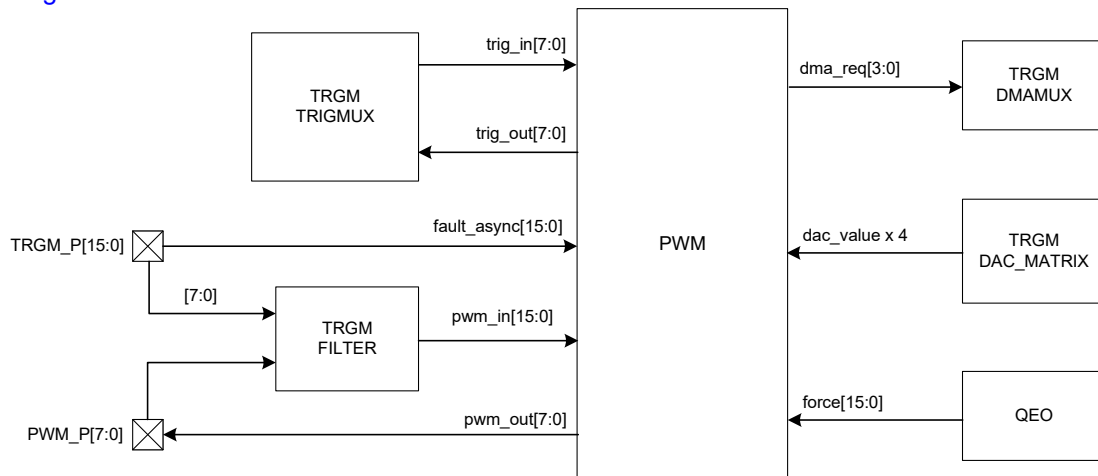


Figure 27: PWM module external connection diagram

- PWM output pwm_out, connected to PWM IO PWM_P
- input capture pwm_in comes from the interconnection manager IO TRGM_P and PWM IO PWM_P. The IO signal first passes through the digital filter of the interconnection manager and can be filtered.
- The trigger input trig_in and the trigger output trig_out are connected to the TRIGMUX of the interconnect manager, which can be connected to various on-chip modules and IO
- asynchronous fault protection input is directly connected to the interconnection manager IO TRGM_P to implement fault protection in a clockless state
- DMA request is connected to the DMAMUX module of the interconnect manager
- DAC input value comes from interconnect manager's DAC transponder
- forced output signal: the force signal of PWM0 comes from the QEO0 module, and the force signal of PWM1 comes from the QEO1 module

and the interconnection manager digital filter. For the specific connection signal bits between

and the interconnection manager TRIGMUX, please refer to [Table 169](#) and [Table 170](#) . For the specific connection signal bits between

and the interconnection manager DMAMUX, please refer to [Table 171](#) . Please refer to [Table 175](#) for the specific connection signal bits between

and the interconnection manager DAC transponder.

36.2 Quadrature encoder interface QEIV2

This product supports 2 orthogonal encoder interfaces QEIV2. QEI supports various types of external quadrature encoders. QEI also supports digital Hall signal input to sense the position information of the motor and provide the motor's steering and speed information.

On this product, the connection relationship between the QEI module and other parts of the system is as shown in [Figure 28](#) :

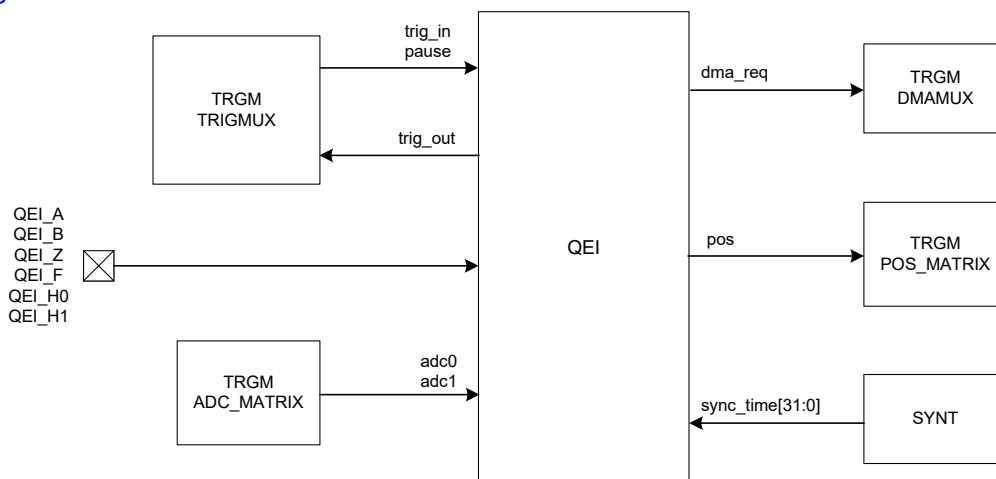


Figure 28: QEI module external connection diagram

- A/U phase input, from IO
- B/V phase input, from IO
- Z/W phase input, from IO
- H0/H1 signal input, also called HOME, from IO
- F signal input, also called FAULT, from IO
- trigger output trig_out and trigger input trig_in, connected to TRIGMUX of the interconnection manager
- ADC SinCos 2-Phase Waveform Input, ADC Repeater from Interconnect Manager
- DMA request is connected to the DMAMUX module of the interconnect manager
- location output, connected to the location forwarder of the interconnect manager
- receives the time count information of the SYNT module to add a timestamp when outputting location information.

and the interconnection manager TRIGMUX, please refer to [Table 169](#) and [Table 170](#) . Please refer to [Table 174](#) for the specific connection signal bits between

and the interconnection manager ADC transponder. For the specific connection signal bits between

and the interconnection manager DMAMUX, please refer to [Table 171](#) . Please refer to [Table 173](#) for the specific connection signal bits between

and the interconnection manager location transponder.

36.3 Quadrature encoder output QEOV2

This product supports 2 orthogonal encoder outputs QEOV2. QEO supports sending orthogonal encoded signals to the outside.

On this product, the connection relationship between the QEO module and other parts of the system is as shown in [Figure 29](#) :

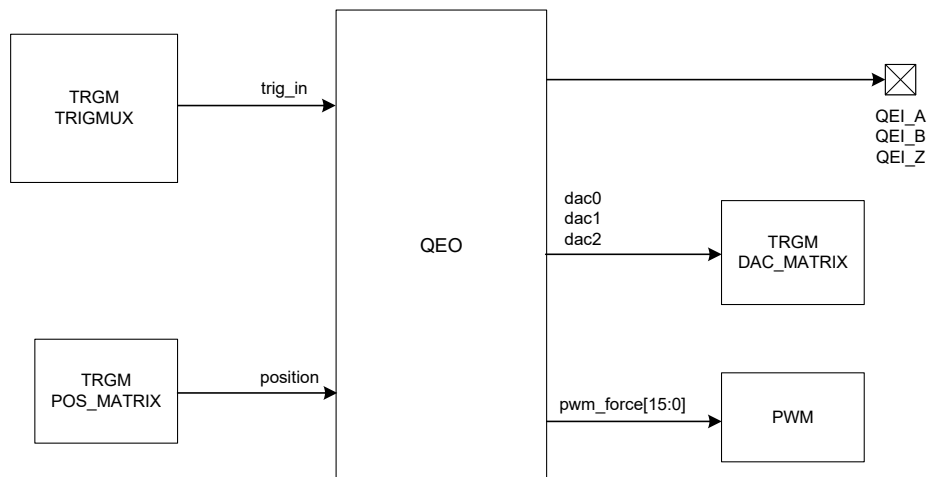


Figure 29: QEO module external connection diagram

- A/U phase output, output through IO
- B/V phase output, output through IO
- Z/W phase output, output through IO
- Location input, location forwarder from Interconnect Manager
- 3-phase DAC output, DAC repeater connected to interconnect manager
- pwm_force signal is directly connected to the PWM module, QEO0 is connected to PWM0, and QEO1 is connected to PWM1
- trigger input trig_in connected to TRIGMUX of the interconnect manager

and the interconnection manager location transponder. Please refer to [Table 175](#) for the specific connection signal bits between

and the interconnection manager DAC transponder. For the specific connection signal bits between

and the interconnection manager TRIGMUX, please refer to [Table 170](#) .

36.4 Programmable Logic Unit PLB

This product supports 1 programmable logic unit PLB. Both the 64-bit trig_in and the 64-bit trig_out of the PLB are connected to the TRIGMUX of the interconnect manager.

can connect IO and other modules through TRIGMUX, and realize multi-level connection of PLB units. For the specific connection signal bits between

and the interconnection manager TRIGMUX, please refer to [Table 169](#) and [Table 170](#) .

36.5 $\Sigma\Delta$ signal receiving unit SDM

This product integrates an $\Sigma\Delta$ signal receiving unit SDM with a built-in 4-channel SINC digital filter. A total of 4 external $\Sigma\Delta$ modulators can be connected.

On this product, the connection relationship between the SDM module and other parts of the system is as shown in [Figure 30](#) :

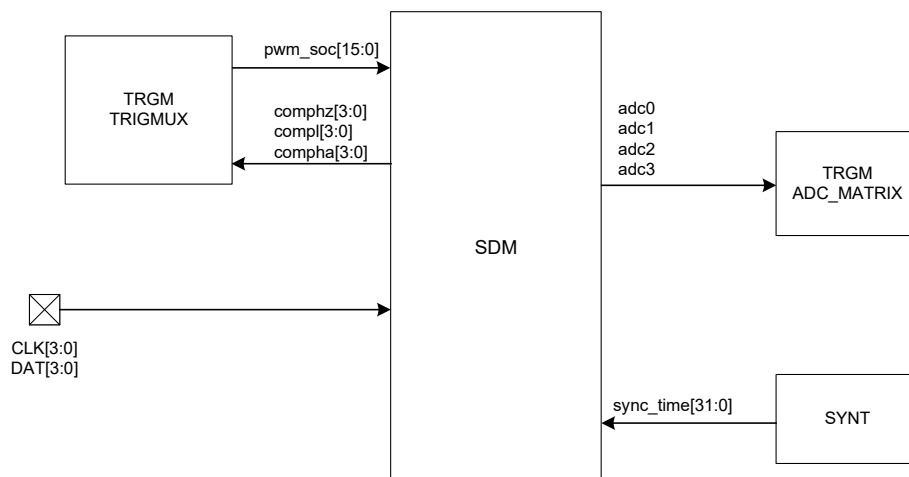


Figure 30: SDM module external connection diagram

- 4 groups of CLK and DAT, input through IO
- 4 sets of ADC outputs, connected to the ADC repeater of the interconnect manager
- 16-bit trigger input, TRIGMUX from interconnect manager
- 4 groups of 12 trigger outputs in total, connected to TRIGMUX of the interconnection manager
- receives the time count information of the SYNT module to add a timestamp when outputting location information.

and the interconnection manager ADC transponder. For the specific connection signal bits between

and the interconnection manager TRIGMUX, please refer to [Table 169](#) and [Table 170](#) .

36.6 Synchronization timer SYNT

This product supports 1 synchronous timer SYNT, SYNT supports 32-bit counter, and 8 output comparison channels. The 8 output comparison channels of

SYNT correspond to the 8-bit trigger output sync_ch0 ~ 7, which is connected to the TRIGMUX of the interconnection manager. The trigger input sync signal of

SYNT also comes from the TRIGMUX of the interconnect manager.

SYNT provides 32-bit time counting information for multiple modules of the electric control system to synchronize the generation of time stamps. For the specific connection signal bits between

and the interconnection manager TRIGMUX, please refer to [Table 169](#) and [Table 170](#) .

36.7 Interconnect Manager TRGM

The interconnection manager TRGM of this product is used to interconnect the signals of various modules inside and outside the electric control system to achieve mutual synchronization and mutual cooperation between peripherals. The

Interconnect Manager consists of the following submodules:

- trigger manager TRIGMUX
- DMA manager DMAMUX
- Digital filter FILTER
- Location Transponder POS_MATRIX
- ADC Repeater ADC_MATRIX
- DAC transponder DAC_MATRIX

36.7.1 trigger manager TRIGMUX

TRIGMUX input signal distribution see [Table 169](#) :

Index	Source	Description
0	VSS	low level voltage
1	VDD	high level voltage
2	PTPC_CMP0	PTPC compare output0
3	PTPC_CMP1	PTPC compare output1
4	USB0_SOF	USB0 start of frame marker
5	DEBUG_FLAG	debug mode flag
6	ESC_SYNC1	ethernet sync1 event
7	ESC_SYNC0	ethernet sync0 event
8	SYNT_CH00	SYNT channel0 pulse output
9	SYNT_CH01	SYNT channel1 pulse output
10	SYNT_CH02	SYNT channel2 pulse output
11	SYNT_CH03	SYNT channel3 pulse output
12	SYNT_CH04	SYNT channel4 pulse output
13	SYNT_CH05	SYNT channel5 pulse output
14	SYNT_CH06	SYNT channel6 pulse output
15	SYNT_CH07	SYNT channel7 pulse output
16	GPTMR0_OUT2	GPTMR0 channel2 compare output
17	GPTMR0_OUT3	GPTMR0 channel3 compare output
18	GPTMR1_OUT2	GPTMR1 channel2 compare output
19	GPTMR1_OUT3	GPTMR1 channel3 compare output
20	GPTMR2_OUT2	GPTMR2 channel2 compare output
21	GPTMR2_OUT3	GPTMR2 channel3 compare output
22	GPTMR3_OUT2	GPTMR3 channel2 compare output
23	GPTMR3_OUT3	GPTMR3 channel3 compare output
24	PWM0_TRGO_0	PWM0 trigger out0

Index	Source	Description
25	PWM0_TRGO_1	PWM0 trigger out1
26	PWM0_TRGO_2	PWM0 trigger out2
27	PWM0_TRGO_3	PWM0 trigger out3
28	PWM0_TRGO_4	PWM0 trigger out4
29	PWM0_TRGO_5	PWM0 trigger out5
30	PWM0_TRGO_6	PWM0 trigger out6
31	PWM0_TRGO_7	PWM0 trigger out7
32	PWM1_TRGO_0	PWM1 trigger out0
33	PWM1_TRGO_1	PWM1 trigger out1
34	PWM1_TRGO_2	PWM1 trigger out2
35	PWM1_TRGO_3	PWM1 trigger out3
36	PWM1_TRGO_4	PWM1 trigger out4
37	PWM1_TRGO_5	PWM1 trigger out5
38	PWM1_TRGO_6	PWM1 trigger out6
39	PWM1_TRGO_7	PWM1 trigger out7
40	TRGM0_P00	TRGM input data0(from IO)
41	TRGM0_P01	TRGM input data1(from IO)
42	TRGM0_P02	TRGM input data2(from IO)
43	TRGM0_P03	TRGM input data3(from IO)
44	TRGM0_P04	TRGM input data4(from IO)
45	TRGM0_P05	TRGM input data5(from IO)
46	TRGM0_P06	TRGM input data6(from IO)
47	TRGM0_P07	TRGM input data7(from IO)
48	TRGM0_P08	TRGM input data8(from IO)
49	TRGM0_P09	TRGM input data9(from IO)
50	TRGM0_P10	TRGM input data10(from IO)
51	TRGM0_P11	TRGM input data11(from IO)
52	TRGM0_P12	TRGM input data12(from IO)
53	TRGM0_P13	TRGM input data13(from IO)
54	TRGM0_P14	TRGM input data14(from IO)
55	TRGM0_P15	TRGM input data15(from IO)
56	SDM0_COMPHZ0	sdm0 channel0 value through zero
57	SDM0_COMPHZ1	sdm0 channel1 value through zero
58	SDM0_COMPHZ2	sdm0 channel2 value through zero
59	SDM0_COMPHZ3	sdm0 channel3 value through zero
60	SDM0_COMPL0	sdm0 channel0 value over upper limit
61	SDM0_COMPL1	sdm0 channel1 value over upper limit
62	SDM0_COMPL2	sdm0 channel2 value over upper limit
63	SDM0_COMPL3	sdm0 channel3 value over upper limit
64	SDM0_COMPHA0	sdm0 channel value beyond lower limit

Index	Source	Description
65	SDM0_COMPHA1	sdm0 channel value beyond lower limit
66	SDM0_COMPHA2	sdm0 channel value beyond lower limit
67	SDM0_COMPHA3	sdm0 channel value beyond lower limit
68	PLB_OUT00	PLB trigger out0
69	PLB_OUT01	PLB trigger out1
70	PLB_OUT02	PLB trigger out2
71	PLB_OUT03	PLB trigger out3
72	PLB_OUT04	PLB trigger out4
73	PLB_OUT05	PLB trigger out5
74	PLB_OUT06	PLB trigger out6
75	PLB_OUT07	PLB trigger out7
76	PLB_OUT08	PLB trigger out8
77	PLB_OUT09	PLB trigger out9
78	PLB_OUT10	PLB trigger out10
79	PLB_OUT11	PLB trigger out11
80	PLB_OUT12	PLB trigger out12
81	PLB_OUT13	PLB trigger out13
82	PLB_OUT14	PLB trigger out14
83	PLB_OUT15	PLB trigger out15
84	PLB_OUT16	PLB trigger out16
85	PLB_OUT17	PLB trigger out17
86	PLB_OUT18	PLB trigger out18
87	PLB_OUT19	PLB trigger out19
88	PLB_OUT20	PLB trigger out20
89	PLB_OUT21	PLB trigger out21
90	PLB_OUT22	PLB trigger out22
91	PLB_OUT23	PLB trigger out23
92	PLB_OUT24	PLB trigger out24
93	PLB_OUT25	PLB trigger out25
94	PLB_OUT26	PLB trigger out26
95	PLB_OUT27	PLB trigger out27
96	PLB_OUT28	PLB trigger out28
97	PLB_OUT29	PLB trigger out29
98	PLB_OUT30	PLB trigger out30
99	PLB_OUT31	PLB trigger out31
100	PLB_OUT32	PLB trigger out32
101	PLB_OUT33	PLB trigger out33
102	PLB_OUT34	PLB trigger out34
103	PLB_OUT35	PLB trigger out35
104	PLB_OUT36	PLB trigger out36

Index	Source	Description
105	PLB_OUT37	PLB trigger out37
106	PLB_OUT38	PLB trigger out38
107	PLB_OUT39	PLB trigger out39
108	PLB_OUT40	PLB trigger out40
109	PLB_OUT41	PLB trigger out41
110	PLB_OUT42	PLB trigger out42
111	PLB_OUT43	PLB trigger out43
112	PLB_OUT44	PLB trigger out44
113	PLB_OUT45	PLB trigger out45
114	PLB_OUT46	PLB trigger out46
115	PLB_OUT47	PLB trigger out47
116	PLB_OUT48	PLB trigger out48
117	PLB_OUT49	PLB trigger out49
118	PLB_OUT50	PLB trigger out50
119	PLB_OUT51	PLB trigger out51
120	PLB_OUT52	PLB trigger out52
121	PLB_OUT53	PLB trigger out53
122	PLB_OUT54	PLB trigger out54
123	PLB_OUT55	PLB trigger out55
124	PLB_OUT56	PLB trigger out56
125	PLB_OUT57	PLB trigger out57
126	PLB_OUT58	PLB trigger out58
127	PLB_OUT59	PLB trigger out59
128	PLB_OUT60	PLB trigger out60
129	PLB_OUT61	PLB trigger out61
130	PLB_OUT62	PLB trigger out62
131	PLB_OUT63	PLB trigger out63
132	QEIO_TRIG_OUT	QEIO trigger out
133	QEI1_TRIG_OUT	QEI1 trigger out
134	ADC0_TRIG_OUT	ADC0 trigger out
135	ADC1_TRIG_OUT	ADC1 trigger out
136	ACMP0_OUT_CH0	ACMP0 CH0 compare output
137	ACMP0_OUT_CH1	ACMP0 CH1 compare output
138	ENET0_PTP_3	ENET0 PTP output bit3
139	QEIO_POS_VALID	QEIO position valid
140	QEI1_POS_VALID	QEI1 position valid
141	ADC0_ADC_VALID	ADC0 ADC data valid
142	ADC1_ADC_VALID	ADC1 ADC data valid
143	SDM0_ADC0_VALID	SDM0 ADC0 data valid
144	SDM0_ADC1_VALID	SDM0 ADC1 data valid

Index	Source	Description
145	SDM0_ADC2_VALID	SDM0 ADC2 data valid
146	SDM0_ADC3_VALID	SDM0 ADC3 data valid
147	QEO0_DAC0_VALID	QEO0 DAC0 data valid
148	QEO1_DAC0_VALID	QEO1 DAC1 data valid

Table 169: TRGM TRIGMUX INPUT List

The output signal distribution of TRIGMUX is shown in [Table 170](#) :

Index	Destination	Description
0	TRGM0_P00	TRGM output data0(to IO)
1	TRGM0_P01	TRGM output data1(to IO)
2	TRGM0_P02	TRGM output data2(to IO)
3	TRGM0_P03	TRGM output data3(to IO)
4	TRGM0_P04	TRGM output data4(to IO)
5	TRGM0_P05	TRGM output data5(to IO)
6	TRGM0_P06	TRGM output data6(to IO)
7	TRGM0_P07	TRGM output data7(to IO)
8	TRGM0_P08	TRGM output data8(to IO)
9	TRGM0_P09	TRGM output data9(to IO)
10	TRGM0_P10	TRGM output data10(to IO)
11	TRGM0_P11	TRGM output data11(to IO)
12	TRGM0_P12	TRGM output data12(to IO)
13	TRGM0_P13	TRGM output data13(to IO)
14	TRGM0_P14	TRGM output data14(to IO)
15	TRGM0_P15	TRGM output data15(to IO)
16	PWM0_TRIG_IN_0	PWM0 trigger in0
17	PWM0_TRIG_IN_1	PWM0 trigger in1
18	PWM0_TRIG_IN_2	PWM0 trigger in2
19	PWM0_TRIG_IN_3	PWM0 trigger in3
20	PWM0_TRIG_IN_4	PWM0 trigger in4
21	PWM0_TRIG_IN_5	PWM0 trigger in5
22	PWM0_TRIG_IN_6	PWM0 trigger in6
23	PWM0_TRIG_IN_7	PWM0 trigger in7
24	PWM1_TRIG_IN_0	PWM1 trigger in0
25	PWM1_TRIG_IN_1	PWM1 trigger in1
26	PWM1_TRIG_IN_2	PWM1 trigger in2
27	PWM1_TRIG_IN_3	PWM1 trigger in3
28	PWM1_TRIG_IN_4	PWM1 trigger in4
29	PWM1_TRIG_IN_5	PWM1 trigger in5
30	PWM1_TRIG_IN_6	PWM1 trigger in6

Index	Destination	Description
31	PWM1_TRIG_IN_7	PWM1 trigger in7
32	ADC0_STRGI	ADC0 sequence queue trigger
33	ADC1_STRGI	ADC1 sequence queue trigger
34	ADCX_PTRGI0A	ADC preemption trigger0A
35	ADCX_PTRGI0B	ADC preemption trigger0B
36	ADCX_PTRGI0C	ADC preemption trigger0C
37	ADCX_PTRGI1A	ADC preemption trigger1A
38	ADCX_PTRGI1B	ADC preemption trigger1B
39	ADCX_PTRGI1C	ADC preemption trigger1C
40	ADCX_PTRGI2A	ADC preemption trigger2A
41	ADCX_PTRGI2B	ADC preemption trigger2B
42	ADCX_PTRGI2C	ADC preemption trigger2C
43	ADCX_PTRGI3A	ADC preemption trigger3A
44	ADCX_PTRGI3B	ADC preemption trigger3B
45	ADCX_PTRGI3C	ADC preemption trigger3C
46	QEIO_TRIG_IN	QEIO trigger in
47	QEI1_TRIG_IN	QEI1 trigger in
48	QEIO_PAUSE	QEIO pause
49	QEI1_PAUSE	QEI1 pause
50	QEO0_TRIG_IN_0	QEO0 trigger in0
51	QEO0_TRIG_IN_1	QEO0 trigger in1
52	QEO1_TRIG_IN_0	QEO1 trigger in0
53	QEO1_TRIG_IN_1	QEO1 trigger in1
54	ACMP0_WIN_CH_0	ACMP0 CH0 window
55	ACMP0_WIN_CH_1	ACMP0 CH1 window
56	GPTMR0_CAPT_2	GPTMR0 channel2 capture in
57	GPTMR0_CAPT_3	GPTMR0 channel3 capture in
58	GPTMR1_CAPT_2	GPTMR1 channel2 capture in
59	GPTMR1_CAPT_3	GPTMR1 channel3 capture in
60	GPTMR2_CAPT_2	GPTMR2 channel2 capture in
61	GPTMR2_CAPT_3	GPTMR2 channel3 capture in
62	GPTMR3_CAPT_2	GPTMR3 channel2 capture in
63	GPTMR3_CAPT_3	GPTMR3 channel3 capture in
64	GPTMR0_SYNC	GPTMR0 hardware sync in
65	GPTMR1_SYNC	GPTMR1 hardware sync in
66	GPTMR2_SYNC	GPTMR2 hardware sync in
67	GPTMR3_SYNC	GPTMR3 hardware sync in
68	PTPC_CAPT_0	PTPC capture in0
69	PTPC_CAPT_1	PTPC capture in1
70	UART_TRIG0	UART0 ~ UART3 trigger event

Index	Destination	Description
71	UART_TRIG1	UART4 ~ UART7 trigger event
72	SDM0_PWM_SOC00	SDM0 sync event input0
73	SDM0_PWM_SOC01	SDM0 sync event input1
74	SDM0_PWM_SOC02	SDM0 sync event input2
75	SDM0_PWM_SOC03	SDM0 sync event input3
76	SDM0_PWM_SOC04	SDM0 sync event input4
77	SDM0_PWM_SOC05	SDM0 sync event input5
78	SDM0_PWM_SOC06	SDM0 sync event input6
79	SDM0_PWM_SOC07	SDM0 sync event input7
80	SDM0_PWM_SOC08	SDM0 sync event input8
81	SDM0_PWM_SOC09	SDM0 sync event input9
82	SDM0_PWM_SOC10	SDM0 sync event input10
83	SDM0_PWM_SOC11	SDM0 sync event input11
84	SDM0_PWM_SOC12	SDM0 sync event input12
85	SDM0_PWM_SOC13	SDM0 sync event input13
86	SDM0_PWM_SOC14	SDM0 sync event input14
87	SDM0_PWM_SOC15	SDM0 sync event input15
88	PLB_IN_00	PLB trigger in0
89	PLB_IN_01	PLB trigger in1
90	PLB_IN_02	PLB trigger in2
91	PLB_IN_03	PLB trigger in3
92	PLB_IN_04	PLB trigger in4
93	PLB_IN_05	PLB trigger in5
94	PLB_IN_06	PLB trigger in6
95	PLB_IN_07	PLB trigger in7
96	PLB_IN_08	PLB trigger in8
97	PLB_IN_09	PLB trigger in9
98	PLB_IN_10	PLB trigger in10
99	PLB_IN_11	PLB trigger in11
100	PLB_IN_12	PLB trigger in12
101	PLB_IN_13	PLB trigger in13
102	PLB_IN_14	PLB trigger in14
103	PLB_IN_15	PLB trigger in15
104	PLB_IN_16	PLB trigger in16
105	PLB_IN_17	PLB trigger in17
106	PLB_IN_18	PLB trigger in18
107	PLB_IN_19	PLB trigger in19
108	PLB_IN_20	PLB trigger in20
109	PLB_IN_21	PLB trigger in21
110	PLB_IN_22	PLB trigger in22

Index	Destination	Description
111	PLB_IN_23	PLB trigger in23
112	PLB_IN_24	PLB trigger in24
113	PLB_IN_25	PLB trigger in25
114	PLB_IN_26	PLB trigger in26
115	PLB_IN_27	PLB trigger in27
116	PLB_IN_28	PLB trigger in28
117	PLB_IN_29	PLB trigger in29
118	PLB_IN_30	PLB trigger in30
119	PLB_IN_31	PLB trigger in31
120	PLB_IN_32	PLB trigger in32
121	PLB_IN_33	PLB trigger in33
122	PLB_IN_34	PLB trigger in34
123	PLB_IN_35	PLB trigger in35
124	PLB_IN_36	PLB trigger in36
125	PLB_IN_37	PLB trigger in37
126	PLB_IN_38	PLB trigger in38
127	PLB_IN_39	PLB trigger in39
128	PLB_IN_40	PLB trigger in40
129	PLB_IN_41	PLB trigger in41
130	PLB_IN_42	PLB trigger in42
131	PLB_IN_43	PLB trigger in43
132	PLB_IN_44	PLB trigger in44
133	PLB_IN_45	PLB trigger in45
134	PLB_IN_46	PLB trigger in46
135	PLB_IN_47	PLB trigger in47
136	PLB_IN_48	PLB trigger in48
137	PLB_IN_49	PLB trigger in49
138	PLB_IN_50	PLB trigger in50
139	PLB_IN_51	PLB trigger in51
140	PLB_IN_52	PLB trigger in52
141	PLB_IN_53	PLB trigger in53
142	PLB_IN_54	PLB trigger in54
143	PLB_IN_55	PLB trigger in55
144	PLB_IN_56	PLB trigger in56
145	PLB_IN_57	PLB trigger in57
146	PLB_IN_58	PLB trigger in58
147	PLB_IN_59	PLB trigger in59
148	PLB_IN_60	PLB trigger in60
149	PLB_IN_61	PLB trigger in61
150	PLB_IN_62	PLB trigger in62

Index	Destination	Description
151	PLB_IN_63	PLB trigger in63
152	SYNT_SYNC	SYNT sync trigger
153	DCCS_TRIG	DCDC clock control system trigger
154	IRQ_TRGM_0	TRGM interrupt0
155	IRQ_TRGM_1	TRGM interrupt1
156	DMA_TRGM_0	TRGM dma request0
157	DMA_TRGM_1	TRGM dma request1
158	ESC_LATCH0	ESC latch0

Table 170: TRGM TRIGMUX OUTPUT List

The user can connect any input signal of TRIGMUX to any output.

Note: When connecting the internal signals of the electric control system to the outside of the electric control system, the clock frequency and the effective length of the signal need to be considered.

36.7.2 DMA manager DMAMUX

There are many DMA requests in the electric control system. DMAMUX is first used inside the electric control system for multiplexing processing, and finally merged into 8 DMA requests, and then sent to the DMAMUX of the SoC.

For the DMA request input allocation of DMAMUX of the interconnect manager, see [Table 171](#) :

Index	Source	Description
0	PWM0_0	PWM0 DMA0
1	PWM0_1	PWM0 DMA1
2	PWM0_2	PWM0 DMA2
3	PWM0_3	PWM0 DMA3
4	PWM1_0	PWM1 DMA0
5	PWM1_1	PWM1 DMA1
6	PWM1_2	PWM1 DMA2
7	PWM1_3	PWM1 DMA3
8	QEIO	QEIO
9	QEIO	QEIO
10	TRGM_0	TRGM DMA0
11	TRGM_1	TRGM DMA1

Table 171: TRGM DMAMUX List

36.7.3 Digital filter FILTER

The digital filter of the interconnection manager can filter the IO input signals of PWM_P and TRGM_P. The input and output connection assignments of the

digital filter are shown in [Table 172](#) :

Number	IO name	Connection 1	Connection 2	Description
0	PWM0_P_0	PWM0_IN_0		PWM0 input capture 0
1	PWM0_P_1	PWM0_IN_1		PWM0 input capture 1
2	PWM0_P_2	PWM0_IN_2		PWM0 input capture 2
3	PWM0_P_3	PWM0_IN_3		PWM0 input capture 3
4	PWM0_P_4	PWM0_IN_4		PWM0 input capture 4
5	PWM0_P_5	PWM0_IN_5		PWM0 input capture 5
6	PWM0_P_6	PWM0_IN_6		PWM0 input capture 6
7	PWM0_P_7	PWM0_IN_7		PWM0 input capture 7
8	PWM0_P_8	PWM0_IN_8		PWM0 input capture 8
9	PWM0_P_9	PWM0_IN_9		PWM0 input capture 9
10	PWM0_P_10	PWM0_IN_10		PWM0 input capture 10
11	PWM0_P_11	PWM0_IN_11		PWM0 input capture 11
12	PWM0_P_12	PWM0_IN_12		PWM0 input capture 12
13	PWM0_P_13	PWM0_IN_13		PWM0 input capture 13
14	PWM0_P_14	PWM0_IN_14		PWM0 input capture 14
15	PWM0_P_15	PWM0_IN_15		PWM0 input capture 15
16	TRGM_P_00	PWM0_IN_8	TRGM_P_00	TRGM IO Input 00 and PWM0 input capture 8
17	TRGM_P_01	PWM0_IN_9	TRGM_P_01	TRGM IO Input 01 and PWM0 input capture 9
18	TRGM_P_02	PWM0_IN_10	TRGM_P_02	TRGM IO Input 02 and PWM0 input capture 10
19	TRGM_P_03	PWM0_IN_11	TRGM_P_03	TRGM IO Input 03 and PWM0 input capture 11
20	TRGM_P_04	PWM0_IN_12	TRGM_P_04	TRGM IO Input 04 and PWM0 input capture 12
21	TRGM_P_05	PWM0_IN_13	TRGM_P_05	TRGM IO Input 05 and PWM0 input capture 13
22	TRGM_P_06	PWM0_IN_14	TRGM_P_06	TRGM IO Input 06 and PWM0 input capture 14
23	TRGM_P_07	PWM0_IN_15	TRGM_P_07	TRGM IO Input 07 and PWM0 input capture 15
24	TRGM_P_08	PWM0_IN_8	TRGM_P_08	TRGM IO Input 08 and PWM0 input capture 8
25	TRGM_P_09	PWM0_IN_9	TRGM_P_09	TRGM IO Input 09 and PWM0 input capture 9
26	TRGM_P_00	PWM0_IN_10	TRGM_P_10	TRGM IO Input 10 and PWM0 input capture 10
27	TRGM_P_00	PWM0_IN_11	TRGM_P_11	TRGM IO Input 11 and PWM0 input capture 11

Number	IO name	Connection 1	Connection 2	Description
28	TRGM_P_00	PWM0_IN_12	TRGM_P_12	TRGM IO Input 12 and PWM0 input capture 12
29	TRGM_P_00	PWM0_IN_13	TRGM_P_13	TRGM IO Input 13 and PWM0 input capture 13
30	TRGM_P_00	PWM0_IN_14	TRGM_P_14	TRGM IO Input 14 and PWM0 input capture 14
31	TRGM_P_00	PWM0_IN_15	TRGM_P_15	TRGM IO Input 15 and PWM0 input capture 15

Table 172: TRGM_FILTER list

36.7.4 Location Transponder POS_MATRIX

The electric control system provides a transmission path for position data between modules. POS data is directly transmitted between modules through the position transponder without the need for CPU or DMA to participate in the transfer.

The modules on this product that can output POS data are QEI, and the modules that can receive POS data are QEO. For the input and output channel assignments of the

position transponder, see [Table 173](#) :

输入编号	input source	output number	output destination
0	QEIO	0	QEO0
1	QEI1	1	QEO1

Table 173: TRGM_POS_MATRIX list

36.7.5 ADC Data Transponder ADC_MATRIX

The electric control system provides a transmission path for analog-to-digital conversion (ADC) data between modules. ADC data is directly transmitted between modules through the ADC transponder without the need for CPU or DMA to participate in the transfer.

The modules on this product that can output ADC data are ADC and SDM, and the modules that can receive POS data are QEI. For the input and output channel allocation of the

ADC transponder, see [Table 174](#) :

Input Index	input source	output number	output destination
0	ADC0	0	QEIO_ADC0
1	ADC1	1	QEIO_ADC1
2	SDM0_ADC0	2	QEI1_ADC0
3	SDM0_ADC1	3	QEI1_ADC1

Input Index	input source	output number	output destination
4	SDM0_ADC2		
5	SDM0_ADC3		

Table 174: TRGM_ADC_MATRIX list

36.7.6 DAC data transponder DAC MATRIX

The electric control system provides a transmission path for digital-to-analog conversion (DAC) data between modules. The DAC data is directly transmitted between modules through the DAC transponder without the need for CPU or DMA to be involved in the transfer.

The modules on this product that can output DAC data are QEO, and the modules that can receive DAC data are ACMP and PWM. For the input and output channel assignments of the

DAC transponder, see [Table 175](#) :

Input Index	input source	output number	output destination
0	QEO0_DAC0	0	ACMP0_CH0
1	QEO0_DAC1	1	ACMP0_CH1
2	QEO0_DAC2	2	PWM0_DAC0
3	QEO1_DAC0	3	PWM0_DAC1
4	QEO1_DAC1	4	PWM0_DAC2
5	QEO1_DAC2	5	PWM0_DAC3
		6	PWM1_DAC0
		7	PWM1_DAC1
		8	PWM1_DAC2
		9	PWM1_DAC3

Table 175: TRGM_DAC_MATRIX list

37 Pulse Width Modulation Module PWMv2

This chapter describes the main functions and features of the pulse code modulation module PWMv2.

37.1 Feature Summary

PWMv2 has the following characteristics:

- contains 4 PWM generation sub-modules, supporting a total of 8 independent PWM or 4 pairs of complementary PWM outputs
 - Each PWM generation sub-module contains a 32-bit counter, including 24-bit integer counting and 8-bit decimal counting. Each sub-module counter supports synchronization
 - PWM sub-module supports 2 independent PWM or 1 pair of complementary PWM outputs
- supports high-resolution PWM output with an output accuracy of up to 100ps
- Complementary PWM output supports dead zone insertion
- supports PWM forced output
- supports fault protection
- The counter cycle is determined by a configurable overload value, and the count overload value configuration and update are performed by the computing unit
- supports 24 comparison values
- comparison values 0~15 are fixedly distributed to 4 PWM generation sub-modules
- Comparative values 16~23 can be allocated freely
- comparison values support configuration by shadow registers or general purpose computing units
- supports 28 shadow registers that can be used to configure counter overload values and comparison values
- supports 16 general purpose computing units that can be used to configure comparison values and counters
- supports 4 DAC values, and the DAC values participate in the operations of the general purpose computing unit and the counter computing unit
- DAC values can be configured by software or entered by other external modules
- supports 8 common trigger inputs
- supports 8 common trigger outputs

37.2 Block Diagram

PWMv2 is shown as [Figure 31](#) .

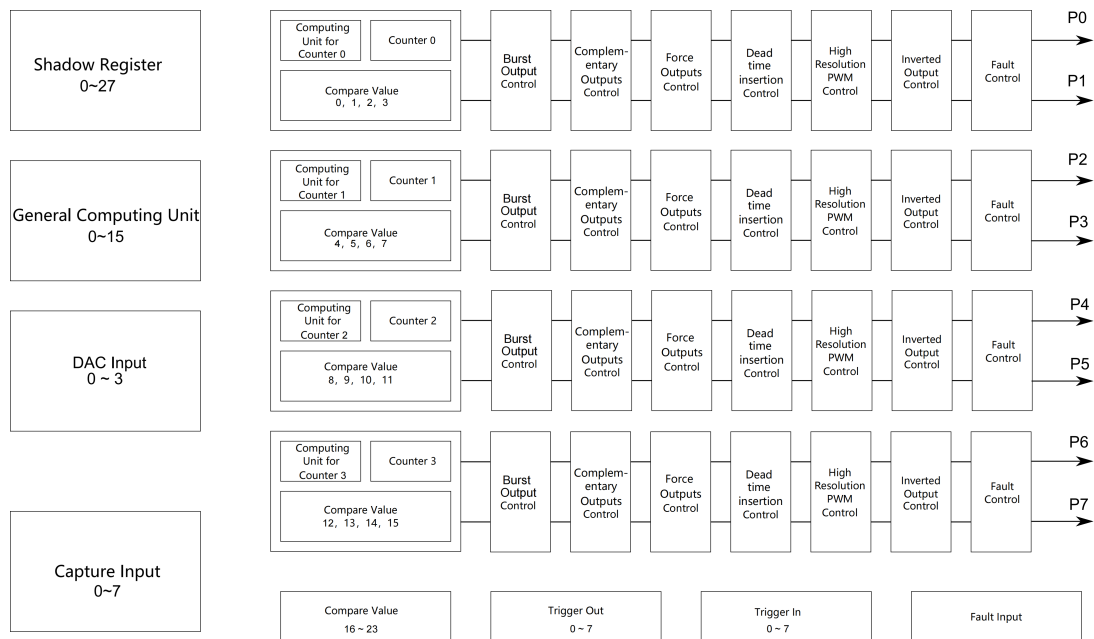


Figure 31: PWMv2 system diagram

37.3 pin description

PWMxP[7:0], inputs or outputs, can be captured as PWM outputs or PWM inputs.

37.4 Function Description

37.4.1 shadow register

PWMv2 module contains 28 32-bit shadow registers. Software directly reads and writes the values of the shadow registers. The shadow registers can be directly or indirectly used for counters, counter overload values, comparison values, counter calculation units or general purpose calculation units. The PWM output waveform is usually directly controlled by the comparison value and the value of the working register of the counter overload value, while the value of the shadow register requires a specific mechanism to be updated to the working register, so these registers are named shadow registers.

37.4.2 DAC Input

PWMv2 module contains four 32-bit DAC inputs. The DAC input values can be directly or indirectly used for counters, counter overload values, comparison values, counter calculation units or general purpose calculation units, which affect the PWM output waveform. The DAC input value for PWMv2 can be generated by other on-chip modules other than PWMv2, and its value can be read through the DAC_VALUE<n>_SV register.

When GLB_CTRL2 [DAC_SW_MODE], the DAC value can be set to be written by software or from other modules on the chip. The DAC_SW_MODE bit field has a width of 4, and each bit corresponds to one DAC input. When a certain bit n is set to 1, the DAC_VALUE<n>_SV register can be read and written by software.

37.4.3 Time Baseline

PWMv2 supports 4 PWM generation sub-modules, each sub-module containing a 32-bit counter that forms the time reference for the output signal.

PWMv2 contains 4 counters. The current value of the counter can be read out by CNT<n>_VAL. This register is 32 bits, of which the upper 24 bits are integer count bits and the lower 8 bits are decimal count bits. The integer count bits are counted according to the clock of PWMv2, and the decimal count bits are used for high-resolution PWM mode.

The four counters support monotonically counting up. The counter starts counting from 0. When the counter counts to the reload value, a reload event is generated. In the IRQ_STS_RELOAD register, the corresponding flag bit will be set to 1. After the reload, the counter starts counting from 0 again. The register CNT_RELOAD_WORK <n> contains the current overloaded value of counter n. The overload value determines the counting period of the counter.

37.4.4 Counter calculation unit

The current overloaded value of the counter CNT_RELOAD_WORK <n> is a read-only register, and its value is determined by the calculation unit of each counter. The output of the calculation unit of counter n is connected to the CNT_RELOAD_WORK <n> register.

Operating modes of the counter calculation unit are as follows:

Figure 32.

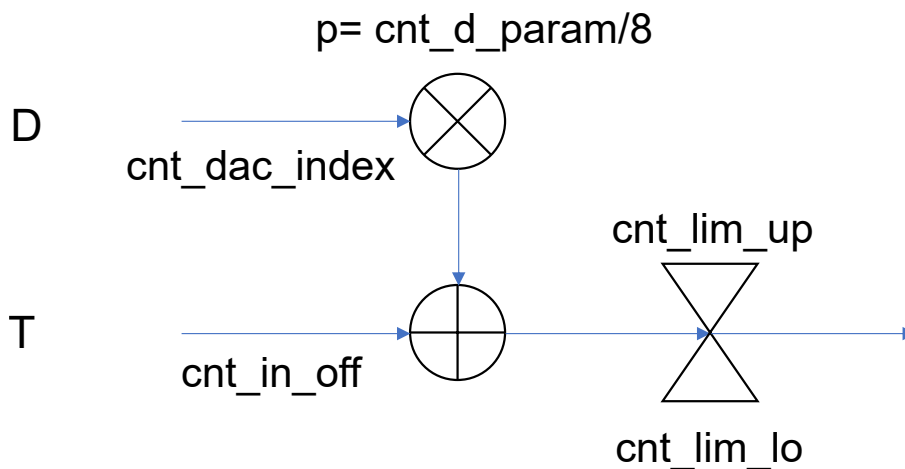


Figure 32: Schematic diagram of PWMv2 counter calculation unit

$$\text{Output} = T + D \cdot p$$

Where T can be configured by CNT_CFG1 <n>[CNT_IN_OFF], select one from the shadow register SHADOW_VAL[0:27] as the T input calculation unit. If CNT_IN_OFF = 3, then SHADOW_VAL[3] is used as the T input of the counter calculation unit.

D can be configured by CNT_CFG1 <n>[CNT_DAC_INDEX], and 1 DAC_VALUE<n>_SV is selected as the D input calculation unit. If CNT_DAC_INDEX = 1, then DAC_VALUE1_SV is used as the D input of the counter calculation unit.

p is a multiplication coefficient and is configured by CNT_CFG0 <n>[CNT_D_PARAM]. The CNT_D_PARAM

bit field is 5 bits, where the highest bit is the sign bit, $CNT\_D\_PARAM[4] = 0b0$, p is a positive number, $p = CNT\_D\_PARAM[3:0] / 8$, $CNT\_D\_PARAM[4] = 0b1$, p is a negative number, $p = -CNT\_D\_PARAM[3:0] / 8$. Therefore, the range of values for p is $-15/8, -14/8, \dots, -1/8, 0, 1/8, 2/8, \dots, 15/8$.

Obviously, if $p = 0$ is configured, the result of the counter calculation unit is independent of the DAC value and is determined only by T , i.e., the overload value is equal to the selected $SHADOW\_VAL$ register value.

The counter calculation unit supports checking and clipping the output result. The $CNT\_CFG1 \langle n \rangle [CNT\_LU\_EN]$ position is 1 to limit the upper limit of the calculation result. If the result is greater than the upper limit, the output is limited to a specific upper limit value. The $CNT\_CFG1 \langle n \rangle [CNT\_LL\_EN]$ position is 1 to limit the lower limit of the calculation result. If the result is less than the lower limit, the output is limited to a specific lower limit value.

The $CNT\_CFG1 \langle n \rangle [CNT\_LIM\_UP]$ bit can configure the upper limit value. CNT_LIM_UP is the index number of the shadow register. For example, $CNT\_LIM\_UP = 0b0000$, then the value of the shadow register $SHADOW\_VAL[0]$ is the upper limit of the result of the calculation unit.

The $CNT\_CFG1 \langle n \rangle [CNT\_LIM\_LO]$ bit can configure the lower limit value. CNT_LIM_LO is the index number of the shadow register. For example, $CNT\_LIM\_LO = 0b0001$, then the value of the shadow register $SHADOW\_VAL[1]$ is the lower limit of the result of the calculation unit.

If the user updates any relevant quantities of the counter calculation unit n through software or other means, such as T , D , and p , the output of the counter calculation unit will also be updated synchronously, but the result will not necessarily be updated to CNT_RELOAD_WORK immediately $\langle n \rangle$, which will change the counting period of the counter. Instead, it is determined by $\langle n \rangle$ the configuration of $CNT\_CFG0 [RLD\_UPDATE\_TIME]$.

- $RLD\_UPDATE\_TIME = 0b00$, updated when $WORK\_CTRL1 [SHADOW\_LOCK]$ bit 1
- $RLD\_UPDATE\_TIME = 0b01$, $CNT\_CFG0 \langle n \rangle [RLD\_CMP\_SEL0]$ or $[RLD\_CM\_SEL1]$ is updated when the selected comparison value matches the counter value
- $RLD\_UPDATE\_TIME = 0b10$, updated when counter is overloaded
- When $RLD\_UPDATE\_TIME = 0b11$, updated on the $\langle n \rangle$ rising edge of the selected universal trigger input signal $CNT\_CFG0 [RLD\_TRIG\_SEL]$

register CNT_GLBCFG can be used to control counters, $TIME\_ENABLE[3:0]$ bits can be used to turn on or off 4 counters, and 1 bit or multiple bits in $TIME\_RESET[3:0]$ can reset the counter to 0.

37.4.5 PWM Generation

PWMv2 supports 4 PWM generation sub-modules, each sub-module contains a 32-bit counter and 4 32-bit comparison values, and supports outputting 2 independent or 1 pair of complementary PWM outputs. The relationship between the counter value and the comparison value determines the PWM output waveform.

The waveform generation mechanism of PWM will be described below. For convenience, the outputs of a PWM generation sub-module are called outputs X and Y , and the four comparison values are called a , b , c , and d from small to large. The corresponding overload value of the counter $CNT\_RELOAD\_WORK \langle n \rangle$, that is, the counting period of the counter is called T .

When $PWM\_CFG1 \langle n \rangle [PAIR\_MODE]$ bit is cleared to 0 ($n = 0, 2, 4, 6$), the outputs X and Y are independent of each other. When $PWM\_CFG0 \langle n \rangle [TRIG\_SEL4]$ is cleared to 0, the output waveform of X depends on a and b , and the output waveform of Y depends on c and d . Take output X as an example:

- $b \leq T$, the output is like [Figure 33](#)

- $a < b$, when $a \leq \text{CNT} < b$, X outputs a high level, and outputs a low level at the rest of the time
- $b < a$, X always outputs low level
- $b > T$, and $a \leq T$, the output is as follows: Figure 34
 - $b \geq (T+a)$, X always outputs a high level
 - $b \leq (T+a)$, $(b-T) \leq \text{CNT} < a$, X outputs a low level, and outputs a high level at other times
- $b > T$, and $a > T$, the output is like Figure 35
 - $a > b$, X always outputs low level
 - $a < b$, $(a-T) \leq \text{CNT} < (b-T)$, X outputs a high level, and outputs a low level at other times
- $b \geq 2T$, no waveform is output

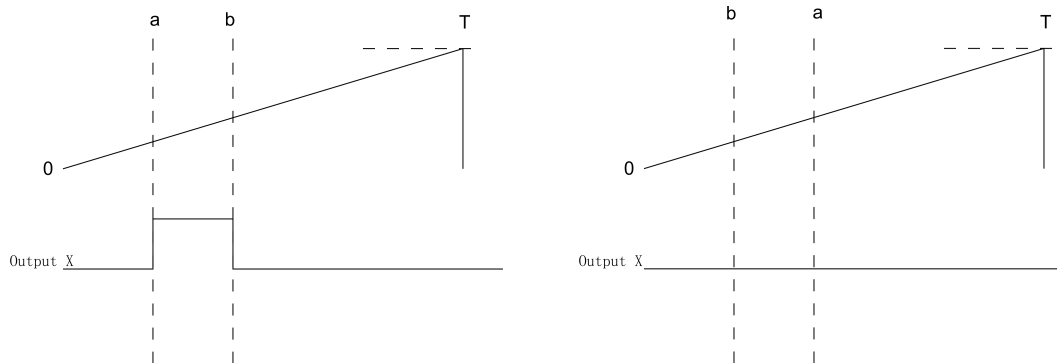


Figure 33: PWM output schematic, $b \leq T$

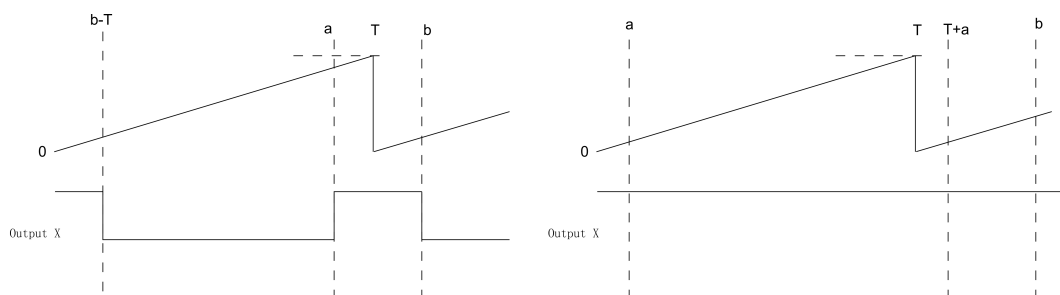


Figure 34: PWM output schematic, $a \leq T$, $b > T$

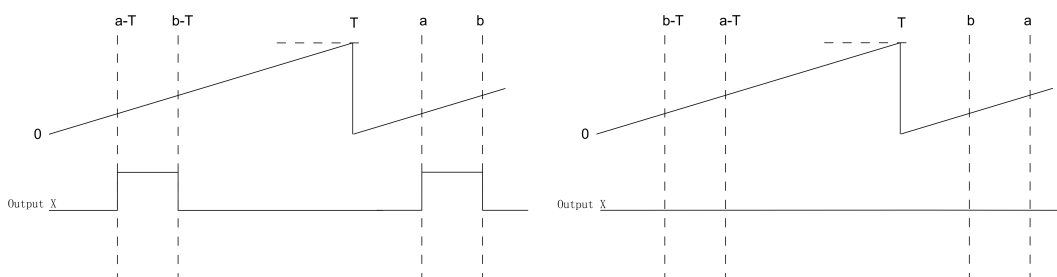


Figure 35: PWM output schematic, $a > T$, $b > T$

When $\text{PWM\_CFG0} \langle n \rangle [\text{TRIG\_SEL4}]$ is set to 1 ($n = 0, 2, 4, 6$), the output X depends on a, b, c, and d. It is recommended that the four comparison values set by the user satisfy $a \leq b$; $b \leq c$; $c \leq d$.

37.4.6 Burst

PWM output supports Burst, which means that the counter generates a specified number of PWM cycles, that is, the PWM output is stopped after a specified number of overloads have occurred. The user can configure

the <n>number of bursts for counter n through CNT_CFG3 [CNT_BURST]. When CNT_BURST is configured to 0, it means counting 1 cycle, when it is configured to 1, it means counting 2 cycles, and when it is configured to 0xFFFF, it means counting continuously.

Regardless of how CNT_BURST is configured, PWM starts outputting after software or hardware triggering

- CNT_GLBCFG register CNT_SW_START[3:0] bit field is set to 1 to realize software triggering. This bit field is 4 bits, and each bit corresponds to 1 counter of PWMv2
- CNT_CFG3 [CNT_HW_START_EN] bit 1 enables hardware triggering, and then by configuring the CNT_CFG3 [CNT_START_SEL] bit field, Select one of the trigger inputs as the hardware trigger signal

is triggered, the Burst is related to the period of the comparison value and the counter, that is, the overloaded value CNT_RELOAD_WORK<n>. For convenience, we call the overloaded value of the counter, that is, the counting period is T. This counter corresponds to the waveform to generate four related comparison values, which are called a, b, c, and d according to the sequence number from low to high.

- If $a < T$, $b < T$, the waveform starts to be output after the counter is reloaded for the first time
- If $a > T$ or $b > T$, the waveform starts to be output after the counter matches the maximum comparison value

When the CNT_BURST configuration is not 0xFFFF, that is, when the PWM output is a limited number of cycles, the stop of the PWM output is also related to the relationship between the comparison value and T

- If $a \leq T$, $b \leq T$, the output waveform ends after the counter is last overloaded
- If $a > T$ or $b > T$, the output waveform ends after the counter is reloaded for the last time and the counter matches the maximum comparison value

Figure 36 Shown as follows for CNT_BURST = 0 (Burst waveform outputs one cycle only): generates a waveform schematic when corresponding to different values of a, b, and T.

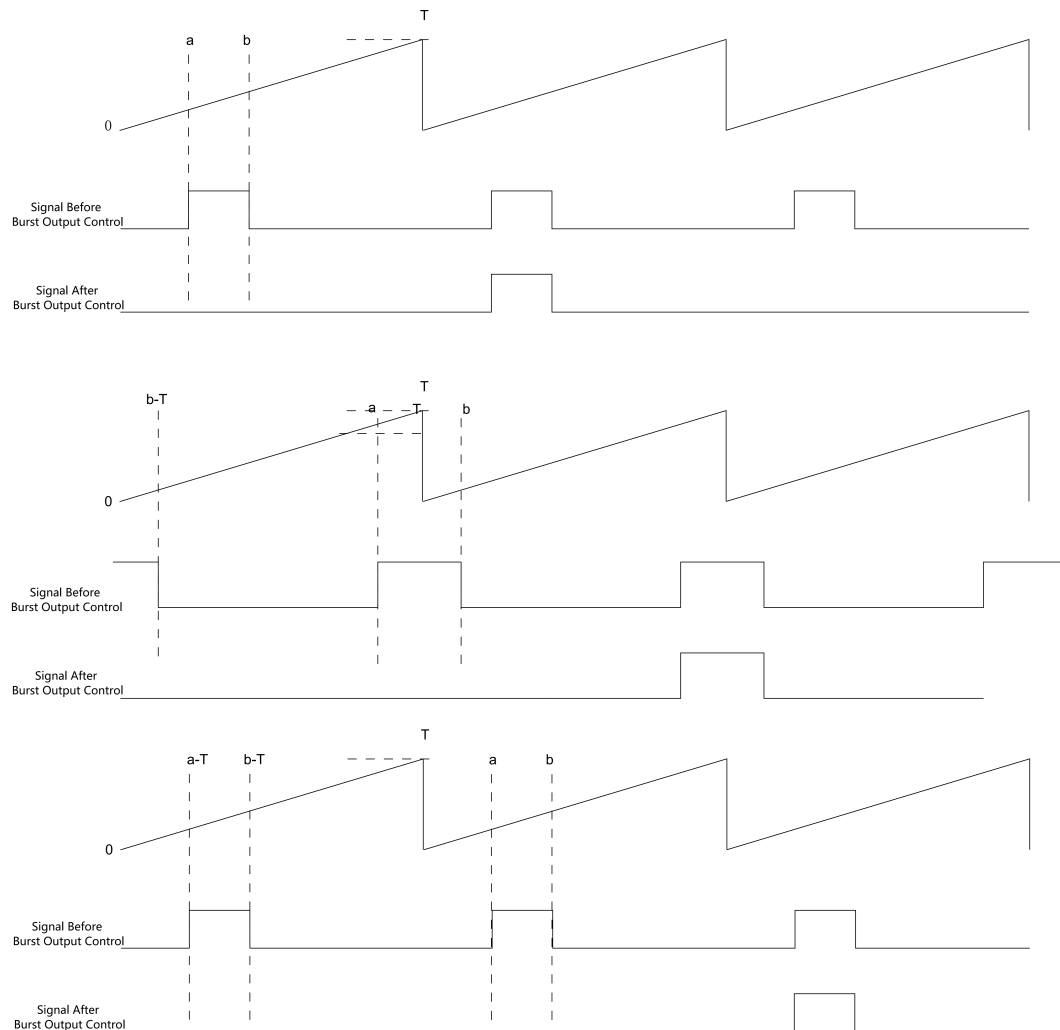


Figure 36: PWM Burst output

37.4.7 Complementary PWM and dead zone

PWM_CFG1 $\langle n \rangle$ [PAIR_MODE] position is 1 ($n = 0, 2, 4, 6$), output X and output Y are PWM pairs. At this time, the PWM output configuration of output Y is invalid, and output Y is the complement of output X.

When output X and output Y form a pair of complementary PWM outputs, the user can use the dead zone control module to insert a certain delay between the toggling of the pair of output reference signals to avoid simultaneous conduction of external switches controlled by the PWM output.

PWMv2 allows the user to insert different delays on 2 complementary reference signals. This allows users to optimize dead time based on the characteristics of off-chip switching devices.

The user can configure the length of the dead zone by configuring the PWM_DEAD $\langle n \rangle$ [DEAD_AREA] bit, and the corresponding dead zone control bits for pairs of X and Y must be configured at the same time to generate the dead zone correctly.

For example, when configuring PWM Output0 and Output1 as complementary PWM outputs, you need to configure: PWM_DEAD $\langle 0 \rangle$ [DEAD_AREA] can delay the rising edge of output 0 by several clock cycles. PWM_DEAD $\langle 1 \rangle$ [DEAD_AREA] can delay the rising edge of output 1 by several clock cycles.

The DEAD_AREA bit field is 24 bits long, of which the upper 16 bits are integers and the lower 8 bits are decimals. It is used for dead-band configuration of high-resolution PWM output. The minimum length of the dead-band is 2 clock cycles, i.e., the minimum configuration for DEAD_AREA is 0x200.

37.4.8 comparison value

PWMv2 supports 8 PWM outputs, outputs and counters, and the corresponding relationship between comparison values is as follows:

- outputs 0 and 1, corresponding to counter 0, and compares the values 0, 1, 2, and 3
- outputs 2 and 3, corresponding to counter 1, and compares values 4, 5, 6, 7
- outputs 4 and 5, corresponding to counter 2, and compares values 8, 9, 10, 11
- outputs 6 and 7, corresponding to counter 3, and compares values 12, 13, 14, 15

PWMv2 supports a total of 24 comparison values. The comparison values 0 - 15 are fixed in 4 PWM generation sub-modules and directly participate in PWM output waveform generation. In addition, when the counter count value matches the comparison value, a comparison event is generated, and the corresponding flag bit in the IRQ_STS_CMP register will be set to 1.

comparison values 16 - 23 can be arbitrarily mapped to one of the four counters. When the selected counter matches the comparison value, a comparison event will be generated, and the corresponding flag bit in the IRQ_STS_CMP register will be set to 1. The user configures the CMP_CFG <n>[CMP_CNT] bit to map the comparison value n(n = 16, 17..., 23) to one of the four counters. For example, CMP_CFG16 [CMP_CNT] = 0b00, when counter 0 counts to the comparison value of 16, a comparison event will be generated, and Bit 16 of the IRQ_STS_CMP register is set to 1.

Users can set the comparison value n through the CMP_CFG <n> register, and CMP_CFG <n>[CMP_IN_SEL] can configure the source of the comparison value n:

- 0x00 ~0x1B, from shadow registers 0~28
- 0x20 ~0x2F, from General Computing Unit 0~15
- 0x30 ~0x37, capture value from PWM input CAPTURE_POS[0:7]
- 0x38 ~0x3B, equal to 1/4 of the overload value CNT_RELOAD_WORK in counters 0~3<n>
- 0x3E, equal to 0xFFFF 000
- 0x3F, equal to 0xFFFFF00
- Other value, equal to 0

As mentioned above, users can configure the source of the comparison value to come from a shadow register and update the shadow register to update the comparison value. You can also set the comparison value source to a general purpose computing unit, or quickly fix it to 0 or a certain large number (such as 0xFFFFF00).

When the source of the comparison value is updated, the comparison value is not necessarily updated immediately, but is <n>updated at a specific time according to the setting of CMP_CFG [CMP_UPDATE_TIME].

- 000: WORK_CTRL1 [SHADOW_LOCK] Updated after position 1
- 001: Update now
- 010: Updated when the corresponding counter is overloaded
- 011: CMP_CFG <n>[CMP_TRIG_SEL] Updated when the selected comparison value is updated when the trigger signal has a rising edge
- 100: CNT_CFG0 <n>[RLD_CMP_SEL0] Updated when the selected comparison value matches the

counter value

- 101: CNT_CFG0 <n>[RLD_CMP_SEL1] Updated when the selected comparison value matches the counter value

37.4.9 General Purpose Computing Unit

PWMv2 contains 16 general purpose computing units, the outputs of these computing units are used to update the comparison values and counter values of the PWM generation sub-module for complex and flexible waveform generation.

The working modes of the general computing unit are [Figure 37](#) :

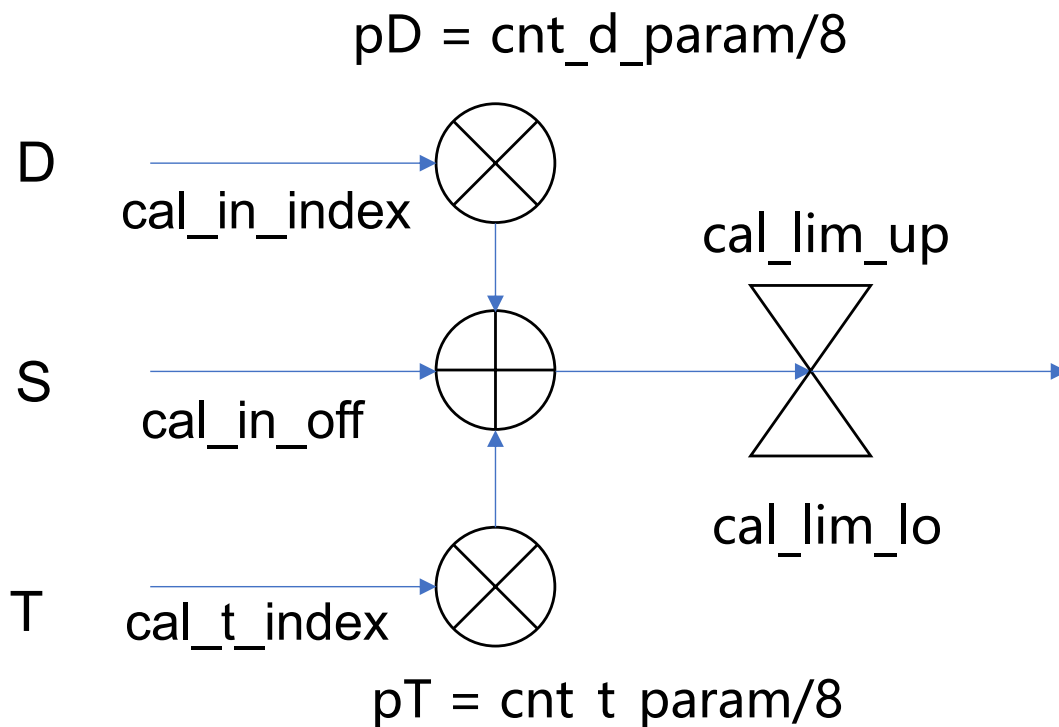


Figure 37: PWMv2 General Purpose Computing Unit

Where T can be configured by CAL_CFG1 <n>[CAL_T_INDEX], select one of the current overloaded values of 4 counters CNT_RELOAD_WORK as the T input calculation unit. If CAL_T_INDEX = 3, then the current overloaded value of counter 3 CNT_RELOAD_WORK [3] is used as the T input of the general purpose calculation unit.

pT is a multiplication coefficient and is configured by CAL_CFG0 <n>[CAL_T_PARAM]. The CAL_T_PARAM bit field is 5 bits, of which the most significant bit is the sign bit, CAL_T_PARAM [4] = 0b0, pT is a positive number, $pT = CAL\_T\_PARAM [3:0] / 8$, CAL_T_PARAM [4] = 0b1, pT is a negative number, and $pT = - CAL\_T\_PARAM [3:0] / 8$. Therefore, the range of values for pT is -15/8, -14/8,...-1/8, 0, 1/8, 2/8,... 15/8.

D can be configured by CAL_CFG1 <n>[CAL_IN_INDEX]. When taking values from 0 to 3, when taking values from 4 to 7 from DAC_VALUE<x>_SV, select 1 of the current values CNT_VAL of 4 counters as the <y>D input calculation unit. For example, CAL_IN_INDEX = 1, then DAC_VALUE1_SV will be used as the D input of the counter calculation unit. For example, CAL_IN_VAL =6, then CNT2_INDEX will be used as the D input of the counter calculation unit.

pD is a multiplication coefficient and is configured by CAL_CFG0 <n>[CAL_D_PARAM]. The CAL_D_PARAM bit field is 5 bits, of which the most significant bit is the sign bit, CAL_D_PARAM [4] = 0b0, pD is a positive number, $pD = \text{CAL\_D\_PARAM}[3:0] / 8$, CAL_D_PARAM [4] = 0b1, pD is a negative number, $pD = -\text{CAL\_D\_PARAM}[3:0] / 8$. Therefore, the range of values for pD is -15/8, -14/8, ..., -1/8, 0, 1/8, 2/8, ..., 15/8.

S can be configured by CAL_CFG1 <n>[CAL_IN_OFF] to select one from the shadow register SHADOW_VAL[0:27] as the S-input computing unit. If CAL_IN_OFF = 3, then SHADOW_VAL[3] is used as the S-input of the general-purpose computing unit.

general computing unit supports checking and clipping the output result. CAL_CFG1 <n>[CAL_LU_EN] position 1 can limit the upper limit of the calculation result. If the result is greater than the upper limit, the output is limited to a specific upper limit value. CAL_CFG1 <n>[CAL_LL_EN] position 1 can limit the lower limit of the calculation result. If the result is less than the lower limit, the output is limited to a specific lower limit value.

The upper and lower limits of the general computing unit are set as follows: [Figure 38](#) :

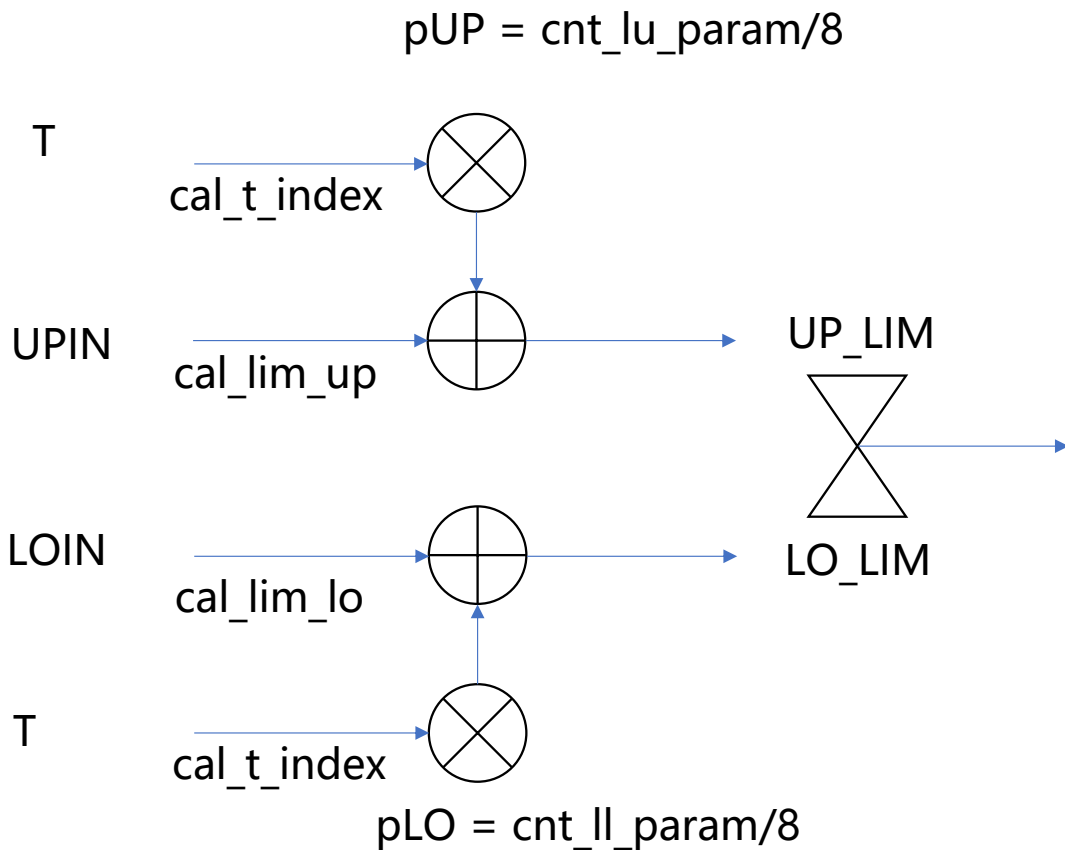


Figure 38: PWMv2 General Purpose Computing Unit Limiting

$$UP\_LIM = UPIN + T * pUP$$

$$LO\_LIM = LOIN + T * pLO$$

Where T is configured by CAL_CFG1 <n>[CAL_T_INDEX], and one of the current overloaded values of 4 counters, CNT_RELOAD_WORK, is selected as the calculation of T participation in the upper and lower limits.

pUP and pLO are multiplication coefficients and are configured by CAL_CFG0 <n>[CAL_LU_PARAM] and [CAL_LL_PARAM]. Taking pUP as an example, the CAL_LU_PARAM bit field is 5 bits, of which the highest bit is the sign bit, CAL_LU_PARAM [4] = 0b0, pUP is a positive number, $pUP = \text{CAL\_LU\_PARAM}[3:0] / 8$,

CAL_LU_PARAM [4] = 0b1, pUP is a negative number, and $pUP = -CAL\_LU\_PARAM [3:0] / 8$. Therefore, the range of values for pUP and pLO is -15/8, -14/8, ..., -1/8, 0, 1/8, 2/8, ..., 15/8.

UPIN and LOIN are configured by CAL_CFG1 <n>[CAL_LIM_UP] and [CAL_LIM_LO] respectively, and one is selected from the shadow register SHADOW_VAL[0:27] as the UPIN and LOIN to participate in the calculation of the upper and lower limits.

37.4.10 Forced output

Users can configure the forced output module to force the PWM output to a certain state. Whether the forced output takes effect can be controlled by a forced effect signal, which can come from software or hardware.

PWM_CFG1 <n>[SW_FORCE_EN] bit controls whether the force validation signal comes from hardware or software:

- When this bit is 0, FORCE_ACITVE comes from hardware, configure PWM_CFG1 <n>[PWM_FORCE_SEL] You can select one of the trigger inputs as a hardware enforcement signal
- When this bit is 1, FORCE_ACITVE comes from software, GLB_CTRL register can configure the software enforced signal state for each PWM output

By configuring the FORCE_MODE[15:0] bit field of the FORCE_MODE register, you can configure the status of the PWM output when the force takes effect. The width of the FORCE_MODE bit field is 16 bits, and every 2 bits corresponds to 1 PWM output:

- 2'b00, forcing output logic 0
- 2'b01, forced output logic 1
- 2'b10, turn off output, pin changes to high-impedance HiZ
- 2'b11, no forced output, that is, the output PWM original output remains unchanged

Forced output affects the PWM output signal. To ensure safety, PWMv2 allows the user to configure the timing of output PWM updates after the forced output takes effect or fails. Through the configuration register PWM_CFG1 <n>[FORCE_TIME], the user can select the timing of output updates.

- 2'b00, instant update
- 2'b01, updated when counter is overloaded
- 2'b10, hardware update, configured by PWM_CFG1 <n>[FORCE_ACT_SEL], select one of the universal trigger inputs, update on rising edge
- 2'b11, not updated

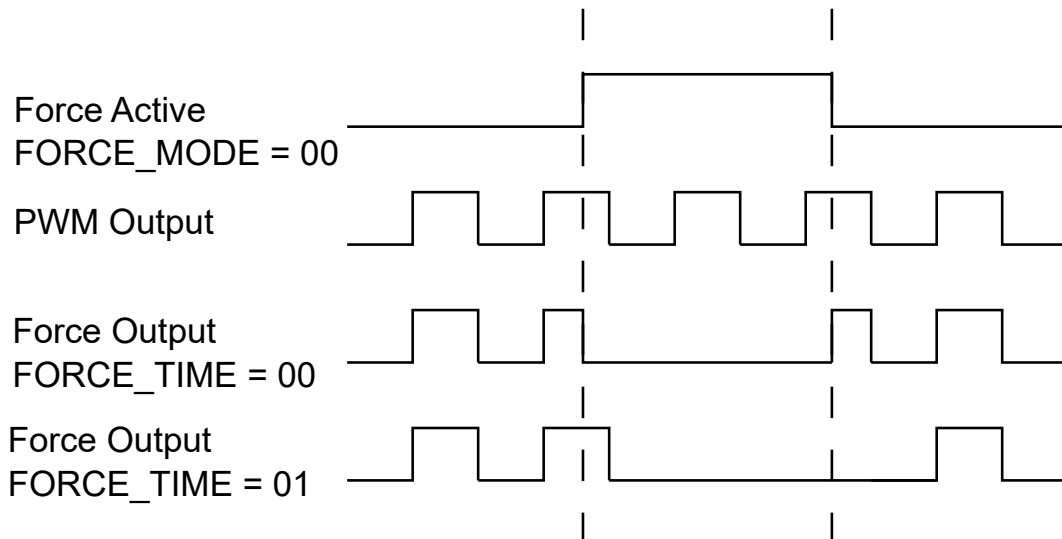


Figure 39: PWMv2 forced output schematic diagram

Similarly, after the `FORCE_MODE` bit field of the `FORCE_MODE` register is updated, whether the new mandatory mode takes effect immediately requires the `PWM_CFG1 <n>[FORCE_UPDATE_TIME]` setting:

- 00: `WORK_CTRL1 [SHADOW_LOCK]` takes effect after position 1
- 01: `<n>` Takes effect when the selected comparison value matches the counter value of `CNT_CFG0 [RLD_CMP_SEL0]` or `[RLD_CM_SEL1]`
- 10: Effective when the corresponding counter is overloaded
- 11: Configured by `PWM_CFG1 <n>[FORCE_TRIG_SEL]`, select one of the universal trigger inputs, update on rising edge

PWMv2 supports forced output modules operated by external modules.

37.4.11 Fault Protection

PWMv2 allows the user to quickly cut off the PWM output signal in real time in an emergency. Each PWM waveform generation sub-module of PWMv2 has its own independent fault protection control unit, which supports synchronous and asynchronous fault protection signals

- Synchronous fault input signal, when the `PWM_CFG0 <n>[FAULT_EN_SYNC]` position is 1, the synchronous fault input to PWM output `n` is effective. Note that the synchronous fault inputs of PWM outputs 0 and 1 are fixed to universal trigger input 0, the synchronous fault inputs of PWM outputs 2 and 3 are fixed to universal trigger input 1, the synchronous fault inputs of PWM outputs 4 and 5 are fixed to universal trigger input 2, and the synchronous fault inputs of PWM outputs 6 and 7 are fixed to universal trigger input 3.
- The asynchronous fault input signal can also come into play if the clock is lost.
When `PWM_CFG0 [FAULT_EN_ASYNC]` bit is 1, the asynchronous fault input to PWM output `n` is active.
`PWM_CFG0 [FAULT_SEL_ASYNC]` can be used to select one of all 16 asynchronous fault input pins, `PWM_CFG0 [FAULT_POL_ASYNC]` can be used to configure the effective polarity of asynchronous fault inputs
`PWM_CFG3 [ASYNC_FAULT_SEL]` can be used to receive the data from 1.1 Select one or more of the

6 asynchronous fault input pins (One asynchronous fault input per bit)

GLB_CTRL3 [ASYNC_FAULT_POL] can be used to configure the effective polarity of the asynchronous fault input selected in the PWM_CFG3 register

PWPWThe fault inputs selected in the M_CFG0 register and PWM_CFG3 register are ultimately or together used as asynchronous fault inputs

<n><n><n><n>

debugging mode can be used as a special fault protection input.

GLB_CTRL2 [DEBUG_IN_EN] bit can be enabled. When set to 1, fault protection can be exited immediately when exiting debugging mode. When set to 0, it is the same as other fault recovery opportunities.

When a fault input is in effect, PWMv2 can force the output to a safe, predetermined state.

<n>determines the state of output n at fault by configuring PWM_CFG0 [FAULT_MODE]:

- 2'b00, when a fault occurs, the output becomes logic 0
- 2'b01, when a fault occurs, the output becomes logic 1
- 2'b1x, when a fault occurs, the output is turned off and the pin becomes high-impedance HiZ

Once a fault occurs, the fault input signal is immediately eliminated, and PWM may not immediately resume normal output. The user can <n>select the timing of PWM recovery by setting PWM_CFG1 [FAULT_REC_TIME]:

- 2'b00, output is restored immediately after the fault is eliminated.
- 2'b01, after the fault is eliminated, the output will resume when the corresponding counter is overloaded.
- 2'b10, after the fault is eliminated, the PWM_CFG1 <n>[FAULT_REC_SEL] bit selects one of the universal trigger inputs as a signal to restore normal output.
- 2'b11, after the fault is recovered, the software restores the output after 1 corresponding position in the GLB_CTRL2[FAULT_CLEAR] bit field

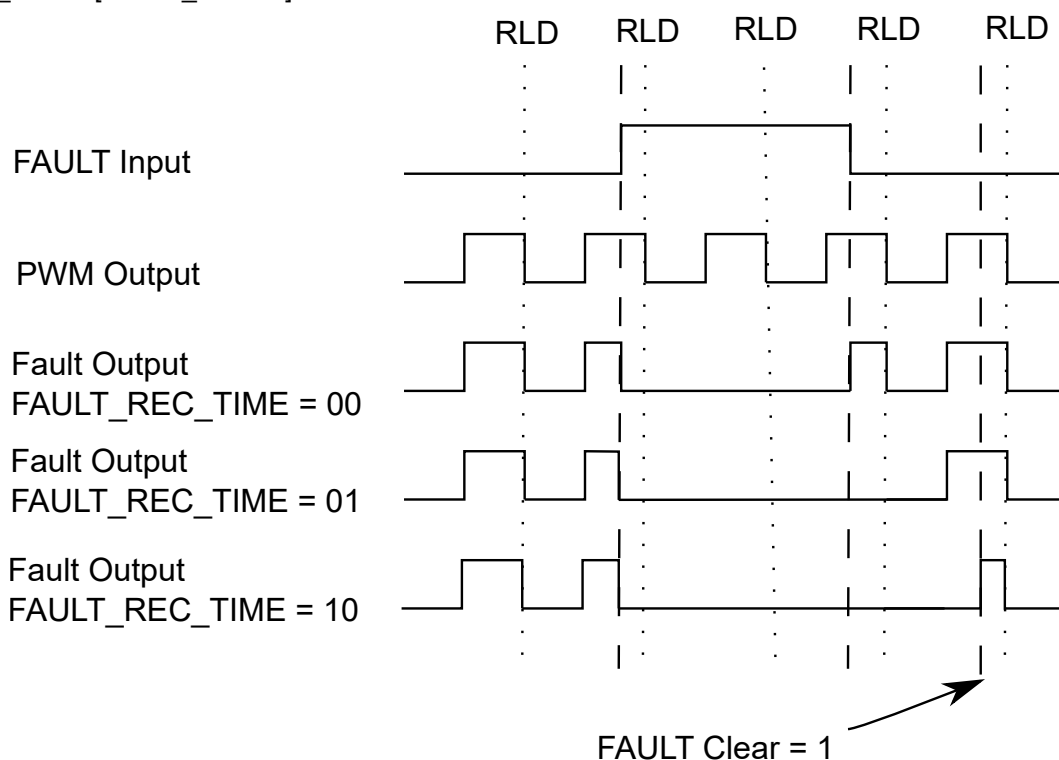


Figure 40: PWMv2 fail-safe recovery

37.4.12 Input Capture

PWMv2 supports 8 input capture modules, which can capture inputs, that is, detect rising and falling edges of the input signal. When a rising edge is detected, the current counter value is saved to the CAPTURE_POS <n>register. When a falling edge is detected, the current counter value is saved to the CAPTURE_NEG <n>register.

When the CAPTURE_POS <n>[CAPTURE_SELGPIO] bit is cleared to 0, the input signal comes from the input and output port P[n] of PWMv2. When set to 1, the input signal comes from other IO.

On this product, other IO refers to:

- TRGM.P[7:0] Input capture connected to PWM0 [7:0]
- TRGM.P[15:8] Input capture connected to PWM1 [7:0]
- TRGM.P[23:16] Connected to input capture of PWM2 [7:0]
- TRGM.P[31:24] Connected to input capture of PWM3 [7:0]

37.4.13 Trigger input

PWMv2 hosts eight universal trigger input signals (TRIG_IN). These universal departure input signals can come from modules other than PWMv2 and can be used to coordinate between PWMv2 and other peripherals.

Universal trigger input for PWMv2 can be used for the following purposes:

- Trigger signal for fault recovery fault_rec_sel
- Synchronous fault input signal, note that trigger inputs 0, 1, 2, and 3 are fixed as synchronous trigger inputs for PWM0/1, PWM2/3, PWM4/5, and PWM6/7 respectively. does not support flexible pairing
- Hardware forced effect signal pwm_force_sel for forced output mode
- Force the output mode FORCE_MODE hardware trigger for updating from shadow register to working register force_trig_sel
- Force output to take effect/fail hardware opportunity force_act_sel
- The hardware trigger opportunity for the counter calculation unit to update the overload value rld_trig_sel
- Use the general computing unit to update the trigger signal of the counter cnt_update_trig 0, cnt_update_trig 1
- Reset counter synchronous trigger signal cnt_reload_trig
- Hardware trigger for comparison value update timing, cmp_trig_sel

37.4.14 Trigger output

PWMv2 supports 8 trigger outputs, which can be configured by the user through the TRIGGER_CFG register. The user can select the 24 comparison values of PWMv2. When the counter count value corresponding to the comparison values matches them, the generated comparison event will be used as a trigger output and sent to other modules and even IO in the form of pulses. The TRIGGER_CFG [TRIGGER_CFG_SEL] bit field serves as a comparison value index and specifies the comparison value used as the trigger output.

37.4.15 interrupt

PWMv2 supports generating the following interrupts:

- Compare values to match interrupts
- Counter overload interrupt
- Input Capture Interrupt
- Fault protection interrupt
- Burst completion interrupt

Comparison Value Match interrupt means that when the counter counts the comparison value, a comparison value match event is generated, and the flag position corresponding to the IRQ_STS_CMP register is 1. Obviously, when the CMPx matches, bit x of IRQ_STS_CMP is set to 1. At the same time, the IRQ_STS [IRQ_CMP] flag bit will also be set to 1. The flag bit in the IRQ_STS_CMP register needs to be written to be cleared. The IRQ_STS [IRQ_CMP] flag bit will be cleared only when all flag bits in the IRQ_STS_CMP register are cleared.

When a comparison value match event occurs, if the corresponding interrupt enable bit in the IRQ_EN_CMP register is 1, a comparison value match interrupt can be generated.

The counter reload interrupt means that when the counter counts to the reload value, a counter reload event is generated, and the corresponding flag position in the IRQ_STS_RELOAD register is 1. Obviously, when CNTx is overloaded, bit x of IRQ_STS_RELOAD is set to 1. At the same time, the IRQ_STS [IRQ_RELOAD] flag bit will also be set to 1. The flag bit in the IRQ_STS_RELOAD register needs to be written to be cleared. The IRQ_STS [IRQ_RELOAD] flag bit will be cleared only when all flag bits in the IRQ_STS_RELOAD register are cleared.

When a counter reload event occurs, a counter reload interrupt can be generated if the corresponding interrupt enable bit in the IRQ_EN_RELOAD register is 1.

The input capture interrupt means that the input capture module of PWMv2 generates an input capture event when the rising edge or falling edge of the input signal is captured. When the rising edge is captured, the corresponding flag bit in the IRQ_STS_CAP_POS register will be set, and at the same time, the IRQ_STS [IRQ_CAPTURE_POS] flag bit will also be set. The flag bit in the IRQ_STS_CAP_POS register needs to be written to be cleared to 0. The IRQ_STS [IRQ_CAPTURE_POS] flag bit will be cleared to 0 only when all flag bits in the IRQ_STS_CAP_POS register are cleared to 0.

When the input capture module of PWMv2 captures a falling edge, the corresponding flag bit in the IRQ_STS_CAP_NEG register will be set, and the IRQ_STS [IRQ_CAPTURE_NEG] flag bit will also be set. The flag bit in the IRQ_STS_CAP_NEG register needs to be written to be cleared to 0. The IRQ_STS [IRQ_CAPTURE_NEG] flag bit will be cleared to 0 only when all flag bits in the IRQ_STS_CAP_NEG register are cleared to 0.

When an input capture event occurs, an input capture interrupt can be generated if the corresponding interrupt enable bit in the IRQ_EN_CAP_POS or IRQ_EN_CAP_NEG register is 1.

Fault protection interrupt means that when a fault occurs, a fault protection event is generated, and the flag position corresponding to the IRQ_STS_FAULT register is set to 1. Obviously, when fault protection for PWM output channel x takes effect, bit x of IRQ_STS_FAULT is set to 1. At the same time, the IRQ_STS [IRQ_FAULT] flag bit will also be set to 1. The flag bit in the IRQ_STS_FAULT register needs to be written to be cleared. The IRQ_STS [IRQ_FAULT] flag bit will be cleared only when all flag bits in the IRQ_STS_FAULT register are cleared.

When a failsafe event occurs, a failsafe interrupt can be generated if the corresponding interrupt enable bit in the IRQ_EN_FAULT register is set to 1.

The burst completion interrupt means that when the counter has generated a specified number of PWM cycles, that is, after the specified number of overloads have occurred, a burst completion event is generated, and

the corresponding flag position in the IRQ_STS_BURSTEND register is 1. Obviously, when the CNTx BURST is completed, bit x of IRQ_STS_BURSTEND is set to 1. At the same time, the IRQ_STS [IRQ_BURSTEND] flag bit will also be set to 1. The flag bit in the IRQ_STS_BURSTEND register needs to be written to be cleared. The IRQ_STS [IRQ_BURSTEND] flag bit will be cleared only when all flag bits in the IRQ_STS_BURSTEND register are cleared.

When a Burst completion event occurs, if the corresponding interrupt enable bit in the IRQ_EN_BURSTEND register is 1, a Burst completion interrupt can be generated.

37.4.16 DMA Request

PWMv2 module supports the generation of 4 DMA requests, which can be configured by the DMA_EN register. When the

DMAx_EN bit is 1, the corresponding DMA request is opened. The PWMv2 module supports generating a DMA request when a comparison value match event or a counter overload event occurs. Users can select the DMAx_SEL bit and set the value of DMAx_SEL to y. When y is between 0 and 23, select the comparison value y match event. When y is between 24 and 27, each DMA request corresponds to a counter overload event, such as a DMA1 request, the counter 1 overload event is selected, and when counter 1 is overloaded, a DMA1 request is generated.

37.5 PWMv2 register

PWMv2 is as follows:

PWM0 base address: 0xF0420000

PWM1 base address: 0xF0424000

Offset	Register Name	Description	Reset Value
0x0000	WORK_CTRL0		0x00000000
0x0004	UNLOCK		0xFFFFFFFF
0x0008	SHADOW_VAL[0]		0x00000000
0x000C	SHADOW_VAL[1]		0x00000000
0x0010	SHADOW_VAL[2]		0x00000000
0x0014	SHADOW_VAL[3]		0x00000000
0x0018	SHADOW_VAL[4]		0x00000000
0x001C	SHADOW_VAL[5]		0x00000000
0x0020	SHADOW_VAL[6]		0x00000000
0x0024	SHADOW_VAL[7]		0x00000000
0x0028	SHADOW_VAL[8]		0x00000000
0x002C	SHADOW_VAL[9]		0x00000000
0x0030	SHADOW_VAL[10]		0x00000000
0x0034	SHADOW_VAL[11]		0x00000000
0x0038	SHADOW_VAL[12]		0x00000000
0x003C	SHADOW_VAL[13]		0x00000000

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

37 PULSE WIDTH MODULATION MODULE

PWMV2

Offset	Register Name	Description	Reset Value
0x0040	SHADOW_VAL[14]		0x00000000
0x0044	SHADOW_VAL[15]		0x00000000
0x0048	SHADOW_VAL[16]		0x00000000
0x004C	SHADOW_VAL[17]		0x00000000
0x0050	SHADOW_VAL[18]		0x00000000
0x0054	SHADOW_VAL[19]		0x00000000
0x0058	SHADOW_VAL[20]		0x00000000
0x005C	SHADOW_VAL[21]		0x00000000
0x0060	SHADOW_VAL[22]		0x00000000
0x0064	SHADOW_VAL[23]		0x00000000
0x0068	SHADOW_VAL[24]		0x00000000
0x006C	SHADOW_VAL[25]		0x00000000
0x0070	SHADOW_VAL[26]		0x00000000
0x0074	SHADOW_VAL[27]		0x00000000
0x0078	FORCE_MODE		0x00000000
0x007C	WORK_CTRL1		0x00000000
0x0100	PWM[0][CFG0]		0x00000000
0x0104	PWM[0][CFG1]		0x00000000
0x0108	PWM[0][DEAD_AREA]		0x00000000
0x010C	PWM[0][CFG3]		0x00000000
0x0110	PWM[1][CFG0]		0x00000000
0x0114	PWM[1][CFG1]		0x00000000
0x0118	PWM[1][DEAD_AREA]		0x00000000
0x011C	PWM[1][CFG3]		0x00000000
0x0120	PWM[2][CFG0]		0x00000000
0x0124	PWM[2][CFG1]		0x00000000
0x0128	PWM[2][DEAD_AREA]		0x00000000
0x012C	PWM[2][CFG3]		0x00000000
0x0130	PWM[3][CFG0]		0x00000000
0x0134	PWM[3][CFG1]		0x00000000
0x0138	PWM[3][DEAD_AREA]		0x00000000
0x013C	PWM[3][CFG3]		0x00000000
0x0140	PWM[4][CFG0]		0x00000000
0x0144	PWM[4][CFG1]		0x00000000
0x0148	PWM[4][DEAD_AREA]		0x00000000
0x014C	PWM[4][CFG3]		0x00000000
0x0150	PWM[5][CFG0]		0x00000000
0x0154	PWM[5][CFG1]		0x00000000
0x0158	PWM[5][DEAD_AREA]		0x00000000
0x015C	PWM[5][CFG3]		0x00000000

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

37 PULSE WIDTH MODULATION MODULE

PWMV2

Offset	Register Name	Description	Reset Value
0x0160	PWM[6][CFG0]		0x00000000
0x0164	PWM[6][CFG1]		0x00000000
0x0168	PWM[6][DEAD_AREA]		0x00000000
0x016C	PWM[6][CFG3]		0x00000000
0x0170	PWM[7][CFG0]		0x00000000
0x0174	PWM[7][CFG1]		0x00000000
0x0178	PWM[7][DEAD_AREA]		0x00000000
0x017C	PWM[7][CFG3]		0x00000000
0x0180	TRIGGER_CFG[0]		0x00000000
0x0184	TRIGGER_CFG[1]		0x00000000
0x0188	TRIGGER_CFG[2]		0x00000000
0x018C	TRIGGER_CFG[3]		0x00000000
0x0190	TRIGGER_CFG[4]		0x00000000
0x0194	TRIGGER_CFG[5]		0x00000000
0x0198	TRIGGER_CFG[6]		0x00000000
0x019C	TRIGGER_CFG[7]		0x00000000
0x01F0	GLB_CTRL		0x00000000
0x01F4	GLB_CTRL2		0x00000000
0x01F8	GLB_CTRL3		0x00000000
0x0200	CNT_RELOAD_WORK[0]		0x00000000
0x0204	CNT_RELOAD_WORK[1]		0x00000000
0x0208	CNT_RELOAD_WORK[2]		0x00000000
0x020C	CNT_RELOAD_WORK[3]		0x00000000
0x0210	CMP_VAL_WORK[0]		0x00000000
0x0214	CMP_VAL_WORK[1]		0x00000000
0x0218	CMP_VAL_WORK[2]		0x00000000
0x021C	CMP_VAL_WORK[3]		0x00000000
0x0220	CMP_VAL_WORK[4]		0x00000000
0x0224	CMP_VAL_WORK[5]		0x00000000
0x0228	CMP_VAL_WORK[6]		0x00000000
0x022C	CMP_VAL_WORK[7]		0x00000000
0x0230	CMP_VAL_WORK[8]		0x00000000
0x0234	CMP_VAL_WORK[9]		0x00000000
0x0238	CMP_VAL_WORK[10]		0x00000000
0x023C	CMP_VAL_WORK[11]		0x00000000
0x0240	CMP_VAL_WORK[12]		0x00000000
0x0244	CMP_VAL_WORK[13]		0x00000000
0x0248	CMP_VAL_WORK[14]		0x00000000
0x024C	CMP_VAL_WORK[15]		0x00000000
0x0250	CMP_VAL_WORK[16]		0x00000000

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

37 PULSE WIDTH MODULATION MODULE

PWMV2

Offset	Register Name	Description	Reset Value
0x0254	CMP_VAL_WORK[17]		0x00000000
0x0258	CMP_VAL_WORK[18]		0x00000000
0x025C	CMP_VAL_WORK[19]		0x00000000
0x0260	CMP_VAL_WORK[20]		0x00000000
0x0264	CMP_VAL_WORK[21]		0x00000000
0x0268	CMP_VAL_WORK[22]		0x00000000
0x026C	CMP_VAL_WORK[23]		0x00000000
0x027C	FORCE_WORK		0x00000000
0x02A0	CNT_VAL[0]		0x00000000
0x02A4	CNT_VAL[1]		0x00000000
0x02A8	CNT_VAL[2]		0x00000000
0x02AC	CNT_VAL[3]		0x00000000
0x02B0	DAC_VALUE_SV[0]		0x00000000
0x02B4	DAC_VALUE_SV[1]		0x00000000
0x02B8	DAC_VALUE_SV[2]		0x00000000
0x02BC	DAC_VALUE_SV[3]		0x00000000
0x0300	CAPTURE_POS[0]		0x00000000
0x0304	CAPTURE_POS[1]		0x00000000
0x0308	CAPTURE_POS[2]		0x00000000
0x030C	CAPTURE_POS[3]		0x00000000
0x0310	CAPTURE_POS[4]		0x00000000
0x0314	CAPTURE_POS[5]		0x00000000
0x0318	CAPTURE_POS[6]		0x00000000
0x031C	CAPTURE_POS[7]		0x00000000
0x0380	CAPTURE_NEG[0]		0x00000000
0x0384	CAPTURE_NEG[1]		0x00000000
0x0388	CAPTURE_NEG[2]		0x00000000
0x038C	CAPTURE_NEG[3]		0x00000000
0x0390	CAPTURE_NEG[4]		0x00000000
0x0394	CAPTURE_NEG[5]		0x00000000
0x0398	CAPTURE_NEG[6]		0x00000000
0x039C	CAPTURE_NEG[7]		0x00000000
0x0400	IRQ_STS		0x00000000
0x0404	IRQ_EN		0x00000000
0x0410	IRQ_STS_CMP		0x00000000
0x0414	IRQ_STS_RELOAD		0x00000000
0x0418	IRQ_STS_CAP_POS		0x00000000
0x041C	IRQ_STS_CAP_NEG		0x00000000
0x0420	IRQ_STS_FAULT		0x00000000
0x0424	IRQ_STS_BURSTEND		0x00000000

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

37 PULSE WIDTH MODULATION MODULE

PWMV2

Offset	Register Name	Description	Reset Value
0x0430	IRQ_EN_CMP		0x00000000
0x0434	IRQ_EN_RELOAD		0x00000000
0x0438	IRQ_EN_CAP_POS		0x00000000
0x043C	IRQ_EN_CAP_NEG		0x00000000
0x0440	IRQ_EN_FAULT		0x00000000
0x0444	IRQ_EN_BURSTEND		0x00000000
0x0480	DMA_EN		0x00000000
0x0500	CNT[0][CFG0]		0x00000000
0x0504	CNT[0][CFG1]		0x00000000
0x0508	CNT[0][CFG2]		0x00000000
0x050C	CNT[0][CFG3]		0x00000000
0x0510	CNT[1][CFG0]		0x00000000
0x0514	CNT[1][CFG1]		0x00000000
0x0518	CNT[1][CFG2]		0x00000000
0x051C	CNT[1][CFG3]		0x00000000
0x0520	CNT[2][CFG0]		0x00000000
0x0524	CNT[2][CFG1]		0x00000000
0x0528	CNT[2][CFG2]		0x00000000
0x052C	CNT[2][CFG3]		0x00000000
0x0530	CNT[3][CFG0]		0x00000000
0x0534	CNT[3][CFG1]		0x00000000
0x0538	CNT[3][CFG2]		0x00000000
0x053C	CNT[3][CFG3]		0x00000000
0x0540	CNT_GLBCFG		0x00000000
0x0600	CAL[0][CFG0]		0x00000000
0x0604	CAL[0][CFG1]		0x00000000
0x0610	CAL[1][CFG0]		0x00000000
0x0614	CAL[1][CFG1]		0x00000000
0x0620	CAL[2][CFG0]		0x00000000
0x0624	CAL[2][CFG1]		0x00000000
0x0630	CAL[3][CFG0]		0x00000000
0x0634	CAL[3][CFG1]		0x00000000
0x0640	CAL[4][CFG0]		0x00000000
0x0644	CAL[4][CFG1]		0x00000000
0x0650	CAL[5][CFG0]		0x00000000
0x0654	CAL[5][CFG1]		0x00000000
0x0660	CAL[6][CFG0]		0x00000000
0x0664	CAL[6][CFG1]		0x00000000
0x0670	CAL[7][CFG0]		0x00000000
0x0674	CAL[7][CFG1]		0x00000000

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

37 PULSE WIDTH MODULATION MODULE

PWMV2

Offset	Register Name	Description	Reset Value
0x0680	CAL[8][CFG0]		0x00000000
0x0684	CAL[8][CFG1]		0x00000000
0x0690	CAL[9][CFG0]		0x00000000
0x0694	CAL[9][CFG1]		0x00000000
0x06A0	CAL[10][CFG0]		0x00000000
0x06A4	CAL[10][CFG1]		0x00000000
0x06B0	CAL[11][CFG0]		0x00000000
0x06B4	CAL[11][CFG1]		0x00000000
0x06C0	CAL[12][CFG0]		0x00000000
0x06C4	CAL[12][CFG1]		0x00000000
0x06D0	CAL[13][CFG0]		0x00000000
0x06D4	CAL[13][CFG1]		0x00000000
0x06E0	CAL[14][CFG0]		0x00000000
0x06E4	CAL[14][CFG1]		0x00000000
0x06F0	CAL[15][CFG0]		0x00000000
0x06F4	CAL[15][CFG1]		0x00000000
0x0800	CMP[0][CFG]		0x00000000
0x0810	CMP[1][CFG]		0x00000000
0x0820	CMP[2][CFG]		0x00000000
0x0830	CMP[3][CFG]		0x00000000
0x0840	CMP[4][CFG]		0x00000000
0x0850	CMP[5][CFG]		0x00000000
0x0860	CMP[6][CFG]		0x00000000
0x0870	CMP[7][CFG]		0x00000000
0x0880	CMP[8][CFG]		0x00000000
0x0890	CMP[9][CFG]		0x00000000
0x08A0	CMP[10][CFG]		0x00000000
0x08B0	CMP[11][CFG]		0x00000000
0x08C0	CMP[12][CFG]		0x00000000
0x08D0	CMP[13][CFG]		0x00000000
0x08E0	CMP[14][CFG]		0x00000000
0x08F0	CMP[15][CFG]		0x00000000
0x0900	CMP[16][CFG]		0x00000000
0x0910	CMP[17][CFG]		0x00000000
0x0920	CMP[18][CFG]		0x00000000
0x0930	CMP[19][CFG]		0x00000000
0x0940	CMP[20][CFG]		0x00000000
0x0950	CMP[21][CFG]		0x00000000
0x0960	CMP[22][CFG]		0x00000000
0x0970	CMP[23][CFG]		0x00000000

Offset	Register Name	Description	Reset Value

Table 176: PWMV2 Register List

37.6 PWM register details

The register details of PWMv2 are as follows:

37.6.1 WORK_CTRL0 (0x0)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SHADOW_UNLOCK																RSVD															
RW																N/A															
0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

WORK_CTRL0 [31:0]

Bits	Name	Description
31	SHADOW_UNLOCK	write 0x...first to unlock, then set related bits in unlock_sel to unlock following shadow registers(from 0x04 to 0x78), otherwise the shadow registers can not be written. The shadow registers will be loaded to work registers only when shadow_lock is 1 or lock is not enabled This bit can be cleared by set shadow_lock bit in work_ctrl1

WORK_CTRL0 Bits

37.6.2 UNLOCK (0x4)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																UNLOCK_BIT															
																RW															
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

UNLOCK [31:0]

Bits	Name	Description
31-0	UNLOCK_BIT	bit2 to bit 29 for value_shadow, bit30 for force_mode the shadow registers can be updated only when related unlock_bit is set; this register can only be updated after unlock

UNLOCK Bits

37.6.3 SHADOW_VAL (0x8 + 0x4 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																VALUE															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SHADOW_VAL [31:0]

Bits	Name	Description
31-0	VALUE	shadow registers, if used as reload or compare point, shall be 24bit clock cycles plus 1bit half cycle and 7bit high-resolution delay

SHADOW_VAL Bits

37.6.4 FORCE_MODE (0x78)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								POLARITY								FORCE_MODE															
N/A								RW								RW															
x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FORCE_MODE [31:0]

Bits	Name	Description
23-16	POLARITY	one bit for one pwm channel, it's used as shadow register when pwm_cfg0.polarity_opt0 is set. output polarity, set to 1 will invert the output

Bits	Name	Description
15-0	FORCE_MODE	2bit for each PWM channel(0~7); 00: force output 0 01: force output 1 10: output highz(pad_oe_*=0) 11: no force this field may be changed by software as shadow register , the update time should be defined by chan_cfg.load, only for PWM channels.

FORCE_MODE Bits

37.6.5 WORK_CTRL1 (0x7C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SHADOW_LOCK	RSVD																														
RW	N/A																														
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

WORK_CTRL1 [31:0]

Bits	Name	Description
31	SHADOW_LOCK	one to lock, software can't write any shadow registers Software have to write 0x... to work_ctrl0 to clear this bit.

WORK_CTRL1 Bits

37.6.6 PWM[CFG0] (0x100 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD							TRIG_SEL4	RSVD												FAULT_SEL_ASYNC				RSVD	FAULT_POL_ASYNC	FAULT_EN_ASYNC	FAULT_EN_SYNC	RSVD	POL_UPDATE_SEL	OUT_POLARITY	POLARITY_OPT0
N/A							RW	N/A												RW	N/A	RW	RW	RW	N/A	RW	RW	RW	RW	RW	RW
x	x	x	x	x	x	x	0	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	x	0	0	0	x	0	0	0

PWM[CFG0] [31:0]

Bits	Name	Description
24	TRIG_SEL4	for N=0/2/4/6, clear to select 2 compare point(N*2~N*2+1); set to select 4 compare point(N*2~N*2+3); or use 2 compare point(N*2+2~N*2+3); for N=1/3/5/7, this bit is no means, it can work on pair mode, or use 2 compare point (N*2+2~N*2+3); assume select ab or abcd, abcd can between 0 and 2T. output will be 1 when counter value between a and b; if b<=a then output all 0; if b>=(T+a), then output all 1;
11-8	FAULT_SEL_ASYNC	select from 16bit async fault from pad
6	FAULT_POL_ASYNC	fault polarity for input fault from pad, 1-active low; 0-active high;
5	FAULT_EN_ASYNC	set to enable the input async faults from pad directly
4	FAULT_EN_SYNC	set to enable the input faults from trig_mux(trigger_in[0] for channel0/1, 1 for 23, 2 for 45, 3 for 67)
2	POL_UPDATE_SE	used when polarity_opt0 is set, define when to update polarity working register. 0: software set work_ctrl1.shadow_lock bit 1: update at reload point;
1	OUT_POLARITY	output polarity, set to 1 will invert the output. when polarity_opt0 is set, this bit is controlled by shadow register, can't be writable; read as working register use compare channel settings(in cmp_cfg) as shadow register update
0	POLARITY_OPT0	set to use shadow polarity

PWM[CFG0] Bits

37.6.7 PWM[CFG1] (0x104 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD			HIGHZ_EN_N	FORCE_UPDATE_TIME		FAULT_MODE	FAULT_REC_TIME	SW_FORCE_EN	PAIR_MODE	PWM_LOGIC	FORCE_TIME	RSVD	FORCE_TRIG_SEL		RSVD	FORCE_ACT_SEL		RSVD	PWM_FORCE_SEL		RSVD	FAULT_REC_SEL									
N/A			RW	RW		RW	RW	RW	RW	RW	RW	N/A	RW		N/A	RW		N/A	RW		N/A	RW									
x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	x	0	0	0	x	0	0	0	x	0	0	0	x	0	0	0

PWM[CFG1] [31:0]

Bits	Name	Description
28	HIGHZ_EN_N	0 to highz pwm outputs(pad_oe*=0), software need set this bit to 1 to enable pwm output
27-26	FORCE_UPDATE_TIME	define when to use the shadow register value for working register(force_mode) 00: software set work_ctrl1.shadow_lock bit 01: use the related counter rld_cmp_sel0 and rld_cmp_sel1, to select one compare point 10: related counter reload time(selected by pwm_cnt) 11: use force_trig_sel to select one of the input trigger NOTE: 00/01 are not recommended since the update time is not controllable, may cause error in complex application. 00 is used for initialization or debug, not suggest for real time update
25-24	FAULT_MODE	00: force output 0 01: force output 1 1x: output highz(pad_oe_*=0)
23-22	FAULT_REC_TIME	00: immediately 01: after main counter reload time 10: use fault_rec_sel to select one of the input trigger 11: software write fault_clear in glb_ctrl2, no effort if pwm_fault is still assert
21	SW_FORCE_EN	0 for hardware force, from trig_mux selected by pwm_force_sel 1 for software force, from glb_ctrl.sw_force
20	PAIR_MODE	if set to 1, PWM work at pair mode, pwm_cfg for channel 2m is used for channel 2m+1(m=0,1,2,3), except the dead area, which is separate for each channel even in pair mode software need set this bit for both channel of one pair, otherwise result unknown.
19-18	PWM_LOGIC	valid only for pwm0/2/4/6 when trig_sel4 is set 00: ab OR cd; 01: ab AND cd; 10: ab XOR cd; 11: cd
17-16	FORCE_TIME	00: force immediately 01: force at main counter reload time 10: force at trig signal selected by force_act_sel 11: no force the force assert/deassert will happen at the force_time; geo force and value also latched at this time

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

37 PULSE WIDTH MODULATION MODULE

PWMV2

Bits	Name	Description
14-12	FORCE_TRIG_SE L	select one trigger from 8, should set to pulse in trig_mux, will load shadow register(force)mode) to force_mode_work at this time
10-8	FORCE_ACT_SEL	select one trigger from 8, should set to pulse in trig_mux, will load hw/sw force at this time
6-4	PWM_FORCE_SE L	select one trigger from 8 as force signal, should be level signal, 1 for force active, 0 for no force
2-0	FAULT_REC_SEL	select one trigger from 8, should set to pulse in trig_mux, used for fault recovery if fault_rec_time is set to 2'b10

PWM[CFG1] Bits

37.6.8 PWM[DEAD_AREA] (0x108 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								DEAD_AREA																							
N/A								RW																							
x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

PWM[DEAD_AREA] [31:0]

Bits	Name	Description
23-0	DEAD_AREA	16bit cycle delay plus 8bit hr_delay min value is 2 cycles, less than 0x200 will be treated as no dead area; NOTE: dead insertion must be configured with pair, that is, for pwm 01/23/45/67. otherwise the result maybe UNKNOWN!!!

PWM[DEAD_AREA] Bits

37.6.9 PWM[CFG3] (0x10C + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																ASYNC_FAULT_SEL															
N/A																RW															

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

37 PULSE WIDTH MODULATION MODULE

PWMV2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PWM[CFG3] [31:0]

Bits	Name	Description
15-0	ASYNC_FAULT_SEL	one bit select to enable one async fault from pad, all enabled fault will be OR together with the old async fault logic. (in old version, user can select one async fault by pwm_cfg0.fault_sel_async, and enabled by pwm_cfg0.fault_en_async) each async fault has dedicaate polarity definition in glb_ctrl3.async_fault_pol

PWM[CFG3] Bits

37.6.10 TRIGGER_CFG (0x180 + 0x4 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																									TRIGGER_OUT_SEL						
N/A																									RW						
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	

TRIGGER_CFG [31:0]

Bits	Name	Description
4-0	TRIGGER_OUT_SEL	select one from 24 compare result as trigger out, set at compare point, clear at reload point.

TRIGGER_CFG Bits

37.6.11 GLB_CTRL (0x1F0)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								SW_FORCE								RSVD						OUTPUT_DELAY		RSVD			HR_PWM_EN	FRAC_DISABLE	RSVD		
N/A								RW								N/A						RW		N/A			RW	RW	N/A		
x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	x	x	x	x	x	x	0	0	x	x	x	0	0	x	x	x

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

37 PULSE WIDTH MODULATION MODULE

PWMV2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

GLB_CTRL [31:0]

Bits	Name	Description
23-16	SW_FORCE	software write 1 to start software force, if the pwm_cfg<n>.sw_force_en is set, force will take effort
9-8	OUTPUT_DELAY	add delay after dead_area insertiong logic, for hr_pwm
4	HR_PWM_EN	set to enable hr pwm, clear to bypass delay chain.
3	FRAC_DISABLE	set to disable bit[7:0] in DAC value when Calculation Unit use it.

GLB_CTRL Bits

37.6.12 GLB_CTRL2 (0x1F4)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD				DAC_SW_MODE				RSVD	DEBUG_OPT	DEBUG_IN_EN	RSVD				FAULT_CLEAR								RSVD						SHADOW_LOCK_EN		
N/A				RW				N/A	RW	RW	N/A				RW								N/A						RW		
x	x	x	x	0	0	0	0	x	0	0	x	x	x	x	x	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	0

GLB_CTRL2 [31:0]

Bits	Name	Description
27-24	DAC_SW_MODE	set for software DAC mode, software can write dac_value*_sv directly, and dac_valid from moto system is ignored
22	DEBUG_OPT	set to disable debug fault immediately when exit debug mode
21	DEBUG_IN_EN	set to enable debug_in signal as fault signal, generally disable pwm output
15-8	FAULT_CLEAR	software write 1 to clear fault event if pwm_cfg.fault_rec_time is 2'b11. software need to clear it after the fault signal is de-assert and before next fault one bit for one pwm channel
0	SHADOW_LOCK_EN	enable shadow_lock feature, if cleared, shadow_lock will be always 0

GLB_CTRL2 Bits

37.6.13 GLB_CTRL3 (0x1F8)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																ASYNC_FAULT_POL															
N/A																RW															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GLB_CTRL3 [31:0]

Bits	Name	Description
15-0	ASYNC_FAULT_POL	async fault polarity, default 0 for active high; set to 1 for active low

GLB_CTRL3 Bits

37.6.14 CNT_RELOAD_WORK (0x200 + 0x4 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																RO															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CNT_RELOAD_WORK [31:0]

Bits	Name	Description
31-0	VALUE	counter0 reload working register

CNT_RELOAD_WORK Bits

37.6.15 CMP_VAL_WORK (0x210 + 0x4 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																RO															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CMP_VAL_WORK [31:0]

Bits	Name	Description
31-0	VALUE	compare point working register

Bits	Name	Description

CMP_VAL_WORK Bits

37.6.16 FORCE_WORK (0x27C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								OUT_POLARITY								FORCE_MODE															
N/A								RO								RO															
x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FORCE_WORK [31:0]

Bits	Name	Description
23-16	OUT_POLARITY	force working register
15-0	FORCE_MODE	force_mode work register

FORCE_WORK Bits

37.6.17 CNT_VAL (0x2A0 + 0x4 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																RO															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CNT_VAL [31:0]

Bits	Name	Description
31-0	VALUE	main counter value

CNT_VAL Bits

37.6.18 DAC_VALUE_SV (0x2B0 + 0x4 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

DAC_VALUE_SV [31:0]

Bits	Name	Description
31-0	VALUE	save dac0_value when dac0_valid if dac_sw_mode is 0; software write dac_value directly if dac_sw_mode is 1

DAC_VALUE_SV Bits

37.6.19 CAPTURE_POS (0x300 + 0x4 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPTURE_POS																								RSVD			CAPTURE_SELGPIO	RSVD		CNT_INDEX	
RO																								N/A			RW	N/A		RW	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x	x	0	x	x	0	0

CAPTURE_POS [31:0]

Bits	Name	Description
31-8	CAPTURE_POS	related counter value captured at input negedge
4	CAPTURE_SELGPIO	0: result from CAP[7:0], from trgm 1: result from CAP[15:8], from gpio
1-0	CNT_INDEX	related counter

CAPTURE_POS Bits

37.6.20 CAPTURE_NEG (0x380 + 0x4 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPTURE_NEG																								RSVD							
RO																								N/A							
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

CAPTURE_NEG [31:0]

Bits	Name	Description
31-8	CAPTURE_NEG	counter value captured at input negedge

Bits	Name	Description

CAPTURE_NEG Bits

37.6.21 IRQ_STS (0x400)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
IRQ_CAL_OVERFLOW	RSVD																										IRQ_BURSTEND	IRQ_FAULT	IRQ_CAPTURE_NEG	IRQ_CAPTURE_POS	IRQ_RELOAD	IRQ_CMP
W1C	N/A																										RO	RO	RO	RO	RO	RO
0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	

IRQ_STS [31:0]

Bits	Name	Description
31	IRQ_CAL_OVERFLOW	end of output burst
5	IRQ_BURSTEND	end of output burst
4	IRQ_FAULT	for external fault event
3	IRQ_CAPTURE_NEG	capture negedge status
2	IRQ_CAPTURE_POS	capture posedge status
1	IRQ_RELOAD	when clock counter reach the reload time
0	IRQ_CMP	for 24 channel, compare event

IRQ_STS Bits

37.6.22 IRQ_EN (0x404)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRQ_EN_OVERFLOW	RSVD																														
0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

IRQ_EN [31:0]

Bits	Name	Description
31	IRQ_EN_OVERFLOW	enable interrupt when calculation unit overflow

IRQ_EN Bits

37.6.23 IRQ_STS_CMP (0x410)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								IRQ_STS_CMP																							
N/A								W1C																							
x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

IRQ_STS_CMP [31:0]

Bits	Name	Description
23-0	IRQ_STS_CMP	interrupt flag for compare point match event, and each bit means one compare point.

IRQ_STS_CMP Bits

37.6.24 IRQ_STS_RELOAD (0x414)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																												IRQ_STS_RELOAD			
N/A																												W1C			
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

IRQ_STS_RELOAD [31:0]

Bits	Name	Description
3-0	IRQ_STS_RELOAD	interrupt flag for reload event , and each bit means one main counter.

IRQ_STS_RELOAD Bits

37.6.25 IRQ_STS_CAP_POS (0x418)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								IRQ_STS_CAP_POS							
N/A																								W1C							
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0

IRQ_STS_CAP_POS [31:0]

Bits	Name	Description
7-0	IRQ_STS_CAP_POS	interrupt flag for posedge capture event , and each bit means one capture channel.

IRQ_STS_CAP_POS Bits

37.6.26 IRQ_STS_CAP_NEG (0x41C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								IRQ_STS_CAP_NEG							
N/A																								W1C							
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0

IRQ_STS_CAP_NEG [31:0]

Bits	Name	Description
7-0	IRQ_STS_CAP_NEG	interrupt flag for negedge capture event , and each bit means one capture channel.

IRQ_STS_CAP_NEG Bits

37.6.27 IRQ_STS_FAULT (0x420)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								IRQ_STS_FAULT							

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
N/A																							WTC								
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	

IRQ_STS_FAULT [31:0]

Bits	Name	Description
7-0	IRQ_STS_FAULT	interrupt flag for external fault event , and each bit means one external fault channel.

IRQ_STS_FAULT Bits

37.6.28 IRQ_STS_BURSTEND (0x424)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																												IRQ_STS_BURSTEND			
N/A																												W1C			
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	

IRQ_STS_BURSTEND [31:0]

Bits	Name	Description
3-0	IRQ_STS_BURSTEND	interrupt flag for output burst done event , and each bit means one main counter.

IRQ_STS_BURSTEND Bits

37.6.29 IRQ_EN_CMP (0x430)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								IRQ_EN_CMP																							
N/A								RW																							
x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IRQ_EN_CMP [31:0]

Bits	Name	Description
23-0	IRQ_EN_CMP	interrupt enable field for compare point match event, and each bit means one compare point.

IRQ_EN_CMP Bits

37.6.30 IRQ_EN_RELOAD (0x434)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																												IRQ_EN_RELOAD			
N/A																												RW			
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

IRQ_EN_RELOAD [31:0]

Bits	Name	Description
3-0	IRQ_EN_RELOAD	interrupt enable field for reload event , and each bit means one main counter.

IRQ_EN_RELOAD Bits

37.6.31 IRQ_EN_CAP_POS (0x438)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																												IRQ_EN_CAP_POS			
N/A																												RW			
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0

IRQ_EN_CAP_POS [31:0]

Bits	Name	Description
7-0	IRQ_EN_CAP_POS	interrupt enable field for posedge capture event , and each bit means one capture channel.

IRQ_EN_CAP_POS Bits

37.6.32 IRQ_EN_CAP_NEG (0x43C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								IRQ_EN_CAP_NEG							
N/A																								RW							
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0

IRQ_EN_CAP_NEG [31:0]

Bits	Name	Description
7-0	IRQ_EN_CAP_NEG	interrupt enable field for negedge capture event , and each bit means one capture channel.

IRQ_EN_CAP_NEG Bits

37.6.33 IRQ_EN_FAULT (0x440)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								IRQ_EN_FAULT							
N/A																								RW							
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0

IRQ_EN_FAULT [31:0]

Bits	Name	Description
7-0	IRQ_EN_FAULT	interrupt enable field for external fault event , and each bit means one external fault channel.

IRQ_EN_FAULT Bits

37.6.34 IRQ_EN_BURSTEND (0x444)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								IRQ_EN_BURSTEND							
N/A																								RW							

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

IRQ_EN_BURSTEND [31:0]

Bits	Name	Description
3-0	IRQ_EN_BURSTEND	interrupt enable field for output burst done event , and each bit means one main counter.

IRQ_EN_BURSTEND Bits

37.6.35 DMA_EN (0x480)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMA3_EN		RSVD		DMA3_SEL				DMA2_EN		RSVD		DMA2_SEL				DMA1_EN		RSVD		DMA1_SEL				DMA0_EN		RSVD		DMA0_SEL			
RW		N/A		RW				RW		N/A		RW				RW		N/A		RW				RW		N/A		RW			
0	x	x	0	0	0	0	0	0	x	x	0	0	0	0	0	0	x	x	0	0	0	0	0	0	x	x	0	0	0	0	0

DMA_EN [31:0]

Bits	Name	Description
31	DMA3_EN	enable dma3
28-24	DMA3_SEL	select one of compare point(0~23) or one reload point(24~27) as dma0
23	DMA2_EN	enable dma2
20-16	DMA2_SEL	select one of compare point(0~23) or one reload point(24~27) as dma0
15	DMA1_EN	enable dma1
12-8	DMA1_SEL	select one of compare point(0~23) or one reload point(24~27) as dma0
7	DMA0_EN	enable dma0
4-0	DMA0_SEL	select one of compare point(0~23) or one reload point(24~27) as dma0

DMA_EN Bits

37.6.36 CNT[CFG0] (0x500 + 0x10 * n)

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

37 PULSE WIDTH MODULATION MODULE

PWMV2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD			RLD_CMP_SEL1					RSVD			RLD_CMP_SEL0					RSVD	RLD_TRIG_SEL			RSVD		RLD_UPDATE_TIME		RSVD			CNT_D_PARAM				
N/A			RW					N/A			RW					N/A	RW			N/A		RW		N/A			RW				
x	x	x	0	0	0	0	0	x	x	x	0	0	0	0	0	x	0	0	0	x	x	0	0	x	x	x	0	0	0	0	0

CNT[CFG0] [31:0]

Bits	Name	Description
28-24	RLD_CMP_SEL1	select one compare point from 24, set to 0x1F to disable current selection, used for reload value, compare value, force value update
20-16	RLD_CMP_SEL0	select one compare point from 24, set to 0x1F to disable current selection
14-12	RLD_TRIG_SEL	select one trigger from 8, should set to pulse in trig_mux
9-8	RLD_UPDATE_TIME	define when to use the calculation output value as reload time 00: software set work_ctrl1.shadow_lock bit 01: use compare point selected by rld_cmp_sel0 or rld_cmp_sel1 10: counter reload time 11: use rld_trig_sel to select one of the input trigger NOTE: 00 is not recommended since the update time is not controllable, may cause error in complex application.
4-0	CNT_D_PARAM	input dac data parameter

CNT[CFG0] Bits

37.6.37 CNT[CFG1] (0x504 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RSVD						CNT_DAC_INDEX		CNT_LU_EN		RSVD		CNT_LIM_UP				CNT_LL_EN		RSVD		CNT_LIM_LO				RSVD			CNT_IN_OFF					
N/A						RW		RW		N/A		RW				RW		N/A		RW				N/A			RW					
x	x	x	x	x	x	0	0	0	x	x	0	0	0	0	0	0	x	x	0	0	0	0	0	x	x	x	0	0	0	0	0	

CNT[CFG1] [31:0]

Bits	Name	Description
25-24	CNT_DAC_INDEX	select one of the dac value
23	CNT_LU_EN	set to enable up limit, use cnt_lu_off to select one of the shadow register value as limitation
20-16	CNT_LIM_UP	up limit offset selection, from one of the shadow_val
15	CNT_LL_EN	set to enable low limit
12-8	CNT_LIM_LO	low limit offset selection, from one of the shadow_val
4-0	CNT_IN_OFF	input data offset selection, from one of the shadow_val, default just shadow reload time

CNT[CFG1] Bits

37.6.38 CNT[CFG2] (0x508 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT_RELOAD_EN	RSVD				CNT_RELOAD_TRIG			RSVD	CNT_UPDATE_TRIG1			CNT_UPDATE_EN1	RSVD			CNT_TRIG1				RSVD	CNT_UPDATE_TRIG0			CNT_UPDATE_EN0	RSVD			CNT_TRIG0			
RW	N/A				RW			N/A	RW			RW	N/A			RW				N/A	RW			RW	N/A			RW			
0	x	x	x	x	0	0	0	x	0	0	0	0	x	x	x	0	0	0	0	x	0	0	0	0	x	x	x	0	0	0	0

CNT[CFG2] [31:0]

Bits	Name	Description
31	CNT_RELOAD_EN	set to use input signal(selected by cnt_reload_trig) to reload timer
26-24	CNT_RELOAD_TRIG	select one trigger from 8, should set to pulse in trig_mux
22-20	CNT_UPDATE_TRIG1	select one trigger from 8, should set to pulse in trig_mux
19	CNT_UPDATE_EN1	set to enable using trig1 to load calculation cell output to counter
15-12	CNT_TRIG1	change counter value to one of the calculation cell output when cnt_update_triger1 issued
10-8	CNT_UPDATE_TRIG0	select one trigger from 8, should set to pulse in trig_mux
7	CNT_UPDATE_EN0	set to enable using trig0 to load calculation cell output to counter
3-0	CNT_TRIG0	change counter value to one of the calculation cell output when cnt_update_triger0 issued

Bits	Name	Description

CNT[CFG2] Bits

37.6.39 CNT[CFG3] (0x50C + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RSVD									CNT_START_SEL			RSVD		CNT_HW_START_EN	RSVD	CNT_BURST																
N/A									RW			N/A		RW	N/A	RW																
x	x	x	x	x	x	x	x	x	0	0	0	x	x	0	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

CNT[CFG3] [31:0]

Bits	Name	Description
22-20	CNT_START_SEL	select one trigger from 8, should set to pulse in trig_mux
17	CNT_HW_START_EN	enable use trigger to start pwm output(at next reload point), by cnt_start_sel
15-0	CNT_BURST	output pwm wave for configured burst(timer period), 0 for one burst; 1 for two burst. set to 0xFFFF for always output pwm wave bit's only used when setting cnt_sw_start or trigger selected by cnt_start_sel

CNT[CFG3] Bits

37.6.40 CNT_GLBCFG (0x540)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD												CNT_SW_START				RSVD				TIMER_RESET				RSVD				TIMER_ENABLE			
N/A												WO				N/A				WO				N/A				RW			
x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	x	x	x	x	0	0	0	0	x	x	x	x	0	0	0	0

CNT_GLBCFG [31:0]

Bits	Name	Description
19-16	CNT_SW_START	set to start pwm output(at next reload point), write only, Auto clear. User can disable pwm output before burst end by start again with cnt_burst=0
11-8	TIMER_RESET	set to clear current timer. Auto clear
3-0	TIMER_ENABLE	1 to enable the main cycle counter; 0 to stop the counter; NOTE: when counter stopped, the related trigger_out will be cleared to 0, the related pwm output will keep value not changed.

CNT_GLBCFG Bits

37.6.41 CAL[CFG0] (0x600 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD			CAL_LU_PARAM					RSVD			CAL_LL_PARAM					RSVD			CAL_T_PARAM					RSVD			CAL_D_PARAM				
N/A			RW					N/A			RW					N/A			RW					N/A			RW				
x	x	x	0	0	0	0	0	x	x	x	0	0	0	0	0	x	x	x	0	0	0	0	0	x	x	x	0	0	0	0	0

CAL[CFG0] [31:0]

Bits	Name	Description
28-24	CAL_LU_PARAM	up limit parameter
20-16	CAL_LL_PARAM	low limit parameter
12-8	CAL_T_PARAM	period parameter
4-0	CAL_D_PARAM	dac/counter value parameter

CAL[CFG0] Bits

37.6.42 CAL[CFG1] (0x604 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD		CAL_T_INDEX		RSVD	CAL_IN_INDEX			CAL_LU_EN	RSVD		CAL_LIM_UP					CAL_LL_EN	RSVD		CAL_LIM_LO					RSVD			CAL_IN_OFF				
N/A		RW		N/A	RW			RW	N/A		RW					RW	N/A		RW					N/A			RW				
x	x	0	0	x	0	0	0	0	x	x	0	0	0	0	0	0	x	x	0	0	0	0	0	x	x	x	0	0	0	0	0

CAL[CFG1] [31:0]

Bits	Name	Description
29-28	CAL_T_INDEX	select one of 4 counter reload time
26-24	CAL_IN_INDEX	0~3 to select one of the dac input value; 4~7 to select one of the current counter value
23	CAL_LU_EN	set to enable up limit
20-16	CAL_LIM_UP	up limit offset selection, select from one of the shadow_val
15	CAL_LL_EN	set to enable low limit
12-8	CAL_LIM_LO	low limit offset selection, select from one of the shadow_val
4-0	CAL_IN_OFF	offset for calculation unit, select from one of the shadow_val.

CAL[CFG1] Bits

37.6.43 CMP[CFG] (0x800 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD		CMP_TRIG_SEL			RSVD		CMP_UPDATE_TIME			RSVD		CMP_IN_SEL					CMP_CNT		RSVD												
N/A		RW			N/A		RW			N/A		RW					RW		N/A												
x	0	0	0	x	0	0	0	x	x	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	

CMP[CFG] [31:0]

Bits	Name	Description
30-28	CMP_TRIG_SEL	select one trigger from 8, should set to pulse in trig_mux
26-24	CMP_UPDATE_TIME	define when to use the shadow register value for working register(trig_cmp) 000: software set work_ctrl1.shadow_lock bit 001: update immediately(at next cycle) 010: related counter reload time 011: use cmp_update_trigger(from trig_mux, selected by cmp_trig_sel) 100: use the related counter rld_cmp_sel0 to select one compare point 101: use the related counter rld_cmp_sel1, to select one compare point 11x: reserved, no update.

Bits	Name	Description
21-16	CMP_IN_SEL	0x00~0x1B select one of the shadow_val directly 0x20~0x2F select one of the calculation cell output 0x30~0x37 select one of capture_pos value(low 8bit are 0) 0x38+k select T/4 0x3E select 0xFFFFF000 0x3F select 0xFFFFF00 others select 0
15-14	CMP_CNT	select one from 4 counters, only for N>=16. for N<16, this field is0, every 4 compare point related to one counter(0123 for counter0, 4567 for counter1....)

CMP[CFG] Bits

38 Internet Manager TRGM

On this product, the input and output signals of multiple peripherals can be connected to each other, so that multiple peripherals can be used in conjunction with each other. The interconnection manager can configure these connections. The

interconnection manager supports a number of inputs and outputs (IOs) that can be assigned to peripherals connected to the interconnection manager and used as inputs or outputs for them.

For input signal allocation of interconnection manager on this product, please refer to [Table 169](#).

For the output signal distribution of the interconnection manager on this product, please refer to [Table 170](#). The

interconnection manager input is equipped with a digital filter that can filter some signals.

For the digital filter allocation of the interconnection manager on this product, please refer to [Table 172](#). The interconnection manager also manages DMA requests from peripherals and sends selected DMA requests to the DMA controller.

For this product, please refer to [Table 171](#) for DMA request signal allocation by interconnection manager.

38.1 Feature Summary

This chapter introduces the main features of the interconnection manager TRGM (Trigger Manager):

- Multiplexer (MUX) array
- supports multiple inputs and multiple outputs
- Each output can be configured individually, choosing from a number of inputs
- Input Signal Digital Filter
- The polarity of the output signal is inverted
- Signal edge-to-pulse conversion
- DMA Request Management

38.2 Function Description

This chapter describes the functions of the interconnection manager TRGM.

38.2.1 Manager Signal Input Output Checkbox

By configuring the interconnection manager register, the user can connect any input signal of the TRGM to the output of the mutual manager, and configure the TRGOCFGx[OCFG] bit to select one of the inputs of the TRGM as the output of OUTx.

38.2.2 Output Configuration

For the output of the interconnection manager, after selecting the signal input, the user can also invert the output by configuring the TRGOCFGx[OPOL] bit. When TRGOCFGx[OPOL] is 0, the output signal remains unchanged, and when set to 1, the output signal is inverted.

In addition, the output configuration of the interconnection manager supports edge-to-pulse conversion of the signal. When TRGOCFGx[REDG2PEN] is set to 1, a pulse with a width of one interconnect manager clock

cycle is output on each rising edge of the input signal. When the

TRGOCFGx[FEDG2PEN] bit is 1, a pulse with a width of one interconnect manager clock cycle is output for each falling edge of the input signal.

38.2.3 Digital Filter

The interconnection manager TRGM provides digital filters for some input signals. Users can use the FILTCFGx[MODE] register filter working mode:

- 3'b000, bypass mode, digital filter off
- 3'b100, filter piercing mode, after the filter input is flipped, the output will also be flipped immediately, and then the filter input will be ignored for a certain period of time. In this mode, the filter output will closely follow the input, while avoiding glitches in the output signal.
- 3'b101, delay filter, the filter input needs to be held for a certain period of time before the filter output can flip
- 3'b110, peak filtering mode, after the filter input is set to logic 0, it will take a certain period of time before the filter output is set to logic 0. If the filter input is set to 1, the filter output will be set to 1 immediately. The purpose of this mode is to filter out input peaks that are not long enough.
- 3'b111, valley filtering mode, after the filter input is set to logic 1, it will take a certain period of time before the filter output is set to logic 1. When the filter input is set to 0, the filter output will be set to 0 immediately. The purpose of this mode is to filter out input troughs that are not long enough.

The user can configure the filter width through the FILTCFGx[FILTLEN] register bits. The unit is the clock cycle of the interconnection manager.

The user can negate the output of the filter by setting the FILTCFGx[OUTINV] bit to 1.

38.2.4 DMA Request Management

Interconnect Manager supports managing DMA requests from some of the peripherals it is interconnected with, and these peripherals' DMA requests are forwarded to DMAMUX through the Interconnect Manager.

The interconnection manager supports 4 DMA requests. The user can configure the DMMACFG0 ~ DMMACFG3 register to select the DMA request output to the DMA controller from the table below.

38.3 TRGM register list

TRGM is as follows:

TRGM0 base address: 0xF047C000

Offset	Register Name	Description	Reset Value
0x0000	FILTCFG[PWM0_IN0]	Filter configure register	0x00000000
0x0004	FILTCFG[PWM0_IN1]	Filter configure register	0x00000000
0x0008	FILTCFG[PWM0_IN2]	Filter configure register	0x00000000
0x000C	FILTCFG[PWM0_IN3]	Filter configure register	0x00000000
0x0010	FILTCFG[PWM0_IN4]	Filter configure register	0x00000000
0x0014	FILTCFG[PWM0_IN5]	Filter configure register	0x00000000
0x0018	FILTCFG[PWM0_IN6]	Filter configure register	0x00000000

Offset	Register Name	Description	Reset Value
0x001C	FILTCFG[PWM0_IN7]	Filter configure register	0x00000000
0x0020	FILTCFG[PWM1_IN0]	Filter configure register	0x00000000
0x0024	FILTCFG[PWM1_IN1]	Filter configure register	0x00000000
0x0028	FILTCFG[PWM1_IN2]	Filter configure register	0x00000000
0x002C	FILTCFG[PWM1_IN3]	Filter configure register	0x00000000
0x0030	FILTCFG[PWM1_IN4]	Filter configure register	0x00000000
0x0034	FILTCFG[PWM1_IN5]	Filter configure register	0x00000000
0x0038	FILTCFG[PWM1_IN6]	Filter configure register	0x00000000
0x003C	FILTCFG[PWM1_IN7]	Filter configure register	0x00000000
0x0040	FILTCFG[TRGM_P_00]	Filter configure register	0x00000000
0x0044	FILTCFG[TRGM_P_01]	Filter configure register	0x00000000
0x0048	FILTCFG[TRGM_P_02]	Filter configure register	0x00000000
0x004C	FILTCFG[TRGM_P_03]	Filter configure register	0x00000000
0x0050	FILTCFG[TRGM_P_04]	Filter configure register	0x00000000
0x0054	FILTCFG[TRGM_P_05]	Filter configure register	0x00000000
0x0058	FILTCFG[TRGM_P_06]	Filter configure register	0x00000000
0x005C	FILTCFG[TRGM_P_07]	Filter configure register	0x00000000
0x0060	FILTCFG[TRGM_P_08]	Filter configure register	0x00000000
0x0064	FILTCFG[TRGM_P_09]	Filter configure register	0x00000000
0x0068	FILTCFG[TRGM_P_10]	Filter configure register	0x00000000
0x006C	FILTCFG[TRGM_P_11]	Filter configure register	0x00000000
0x0070	FILTCFG[TRGM_P_12]	Filter configure register	0x00000000
0x0074	FILTCFG[TRGM_P_13]	Filter configure register	0x00000000
0x0078	FILTCFG[TRGM_P_14]	Filter configure register	0x00000000
0x007C	FILTCFG[TRGM_P_15]	Filter configure register	0x00000000
0x0400	DMACFG[0]	DMA request configure register	0x00000000
0x0404	DMACFG[1]	DMA request configure register	0x00000000
0x0408	DMACFG[2]	DMA request configure register	0x00000000
0x040C	DMACFG[3]	DMA request configure register	0x00000000
0x0410	DMACFG[4]	DMA request configure register	0x00000000
0x0414	DMACFG[5]	DMA request configure register	0x00000000
0x0418	DMACFG[6]	DMA request configure register	0x00000000
0x041C	DMACFG[7]	DMA request configure register	0x00000000
0x0500	GCR		0x00000000
0x0540	ADC_MATRIX_SEL0	adc matrix select register0	0x00000000
0x0580	DAC_MATRIX_SEL0	dac matrix select register0	0x00000000
0x0584	DAC_MATRIX_SEL1	dac matrix select register1	0x00000000
0x0588	DAC_MATRIX_SEL2	dac matrix select register2	0x00000000
0x05C0	POS_MATRIX_SEL0	position matrix select register0	0x00000000
0x0600	TRGM_IN[0]	trigmux input read register0	0x00000000

Offset	Register Name	Description	Reset Value
0x0604	TRGM_IN[1]	trigmux input read register1	0x00000000
0x0608	TRGM_IN[2]	trigmux input read register2	0x00000000
0x060C	TRGM_IN[3]	trigmux input read register3	0x00000000
0x0610	TRGM_IN[4]	trigmux input read register4	0x00000000
0x0680	TRGM_OUT[0]	trigmux output read register0	0x00000000
0x0684	TRGM_OUT[1]	trigmux output read register1	0x00000000
0x0688	TRGM_OUT[2]	trigmux output read register2	0x00000000
0x068C	TRGM_OUT[3]	trigmux output read register3	0x00000000
0x0690	TRGM_OUT[4]	trigmux output read register4	0x00000000
0x0800	PWM_DELAY_CFG	pwm delay chain config register	0x00000000
0x0804	PWM_CALIB_CFG	pwm delay chain calibration control register	0x00000000
0x1000	TRGOCFG[TRGM_P_00]	Trigger manager output configure register	0x00000000
0x1004	TRGOCFG[TRGM_P_01]	Trigger manager output configure register	0x00000000
0x1008	TRGOCFG[TRGM_P_02]	Trigger manager output configure register	0x00000000
0x100C	TRGOCFG[TRGM_P_03]	Trigger manager output configure register	0x00000000
0x1010	TRGOCFG[TRGM_P_04]	Trigger manager output configure register	0x00000000
0x1014	TRGOCFG[TRGM_P_05]	Trigger manager output configure register	0x00000000
0x1018	TRGOCFG[TRGM_P_06]	Trigger manager output configure register	0x00000000
0x101C	TRGOCFG[TRGM_P_07]	Trigger manager output configure register	0x00000000
0x1020	TRGOCFG[TRGM_P_08]	Trigger manager output configure register	0x00000000
0x1024	TRGOCFG[TRGM_P_09]	Trigger manager output configure register	0x00000000
0x1028	TRGOCFG[TRGM_P_10]	Trigger manager output configure register	0x00000000
0x102C	TRGOCFG[TRGM_P_11]	Trigger manager output configure register	0x00000000
0x1030	TRGOCFG[TRGM_P_12]	Trigger manager output configure register	0x00000000
0x1034	TRGOCFG[TRGM_P_13]	Trigger manager output configure register	0x00000000

Offset	Register Name	Description	Reset Value
0x1038	TRGOCFG[TRGM_P_14]	Trigger manager output configure register	0x00000000
0x103C	TRGOCFG[TRGM_P_15]	Trigger manager output configure register	0x00000000
0x1040	TRGOCFG[PWM0_TRIG_IN0]	Trigger manager output configure register	0x00000000
0x1044	TRGOCFG[PWM0_TRIG_IN1]	Trigger manager output configure register	0x00000000
0x1048	TRGOCFG[PWM0_TRIG_IN2]	Trigger manager output configure register	0x00000000
0x104C	TRGOCFG[PWM0_TRIG_IN3]	Trigger manager output configure register	0x00000000
0x1050	TRGOCFG[PWM0_TRIG_IN4]	Trigger manager output configure register	0x00000000
0x1054	TRGOCFG[PWM0_TRIG_IN5]	Trigger manager output configure register	0x00000000
0x1058	TRGOCFG[PWM0_TRIG_IN6]	Trigger manager output configure register	0x00000000
0x105C	TRGOCFG[PWM0_TRIG_IN7]	Trigger manager output configure register	0x00000000
0x1060	TRGOCFG[PWM1_TRIG_IN0]	Trigger manager output configure register	0x00000000
0x1064	TRGOCFG[PWM1_TRIG_IN1]	Trigger manager output configure register	0x00000000
0x1068	TRGOCFG[PWM1_TRIG_IN2]	Trigger manager output configure register	0x00000000
0x106C	TRGOCFG[PWM1_TRIG_IN3]	Trigger manager output configure register	0x00000000
0x1070	TRGOCFG[PWM1_TRIG_IN4]	Trigger manager output configure register	0x00000000
0x1074	TRGOCFG[PWM1_TRIG_IN5]	Trigger manager output configure register	0x00000000
0x1078	TRGOCFG[PWM1_TRIG_IN6]	Trigger manager output configure register	0x00000000
0x107C	TRGOCFG[PWM1_TRIG_IN7]	Trigger manager output configure register	0x00000000
0x1080	TRGOCFG[ADC0_STRGI]	Trigger manager output configure register	0x00000000
0x1084	TRGOCFG[ADC1_STRGI]	Trigger manager output configure register	0x00000000

Offset	Register Name	Description	Reset Value
0x1088	TRGOCFG[ADCX_PTRGI0A]	Trigger manager output configure register	0x00000000
0x108C	TRGOCFG[ADCX_PTRGI0B]	Trigger manager output configure register	0x00000000
0x1090	TRGOCFG[ADCX_PTRGI0C]	Trigger manager output configure register	0x00000000
0x1094	TRGOCFG[ADCX_PTRGI1A]	Trigger manager output configure register	0x00000000
0x1098	TRGOCFG[ADCX_PTRGI1B]	Trigger manager output configure register	0x00000000
0x109C	TRGOCFG[ADCX_PTRGI1C]	Trigger manager output configure register	0x00000000
0x10A0	TRGOCFG[ADCX_PTRGI2A]	Trigger manager output configure register	0x00000000
0x10A4	TRGOCFG[ADCX_PTRGI2B]	Trigger manager output configure register	0x00000000
0x10A8	TRGOCFG[ADCX_PTRGI2C]	Trigger manager output configure register	0x00000000
0x10AC	TRGOCFG[ADCX_PTRGI3A]	Trigger manager output configure register	0x00000000
0x10B0	TRGOCFG[ADCX_PTRGI3B]	Trigger manager output configure register	0x00000000
0x10B4	TRGOCFG[ADCX_PTRGI3C]	Trigger manager output configure register	0x00000000
0x10B8	TRGOCFG[QEI0_TRIG_IN]	Trigger manager output configure register	0x00000000
0x10BC	TRGOCFG[QEI1_TRIG_IN]	Trigger manager output configure register	0x00000000
0x10C0	TRGOCFG[QEI0_PAUSE]	Trigger manager output configure register	0x00000000
0x10C4	TRGOCFG[QEI1_PAUSE]	Trigger manager output configure register	0x00000000
0x10C8	TRGOCFG[QEO0_TRIG_IN0]	Trigger manager output configure register	0x00000000
0x10CC	TRGOCFG[QEO0_TRIG_IN1]	Trigger manager output configure register	0x00000000
0x10D0	TRGOCFG[QEO1_TRIG_IN0]	Trigger manager output configure register	0x00000000
0x10D4	TRGOCFG[QEO1_TRIG_IN1]	Trigger manager output configure register	0x00000000

Offset	Register Name	Description	Reset Value
0x10D8	TRGOCFG[ACMP0_CH0_WIN]	Trigger manager output configure register	0x00000000
0x10DC	TRGOCFG[ACMP0_CH1_WIN]	Trigger manager output configure register	0x00000000
0x10E0	TRGOCFG[GPTMR0_IN2]	Trigger manager output configure register	0x00000000
0x10E4	TRGOCFG[GPTMR0_IN3]	Trigger manager output configure register	0x00000000
0x10E8	TRGOCFG[GPTMR1_IN2]	Trigger manager output configure register	0x00000000
0x10EC	TRGOCFG[GPTMR1_IN3]	Trigger manager output configure register	0x00000000
0x10F0	TRGOCFG[GPTMR2_IN2]	Trigger manager output configure register	0x00000000
0x10F4	TRGOCFG[GPTMR2_IN3]	Trigger manager output configure register	0x00000000
0x10F8	TRGOCFG[GPTMR3_IN2]	Trigger manager output configure register	0x00000000
0x10FC	TRGOCFG[GPTMR3_IN3]	Trigger manager output configure register	0x00000000
0x1100	TRGOCFG[GPTMR0_SYNCI]	Trigger manager output configure register	0x00000000
0x1104	TRGOCFG[GPTMR1_SYNCI]	Trigger manager output configure register	0x00000000
0x1108	TRGOCFG[GPTMR2_SYNCI]	Trigger manager output configure register	0x00000000
0x110C	TRGOCFG[GPTMR3_SYNCI]	Trigger manager output configure register	0x00000000
0x1110	TRGOCFG[CAN_PTPC0_CAP]	Trigger manager output configure register	0x00000000
0x1114	TRGOCFG[CAN_PTPC1_CAP]	Trigger manager output configure register	0x00000000
0x1118	TRGOCFG[UART_TRIG0]	Trigger manager output configure register	0x00000000
0x111C	TRGOCFG[UART_TRIG1]	Trigger manager output configure register	0x00000000
0x1120	TRGOCFG[SDM_PWM_SOC0]	Trigger manager output configure register	0x00000000
0x1124	TRGOCFG[SDM_PWM_SOC1]	Trigger manager output configure register	0x00000000

Offset	Register Name	Description	Reset Value
0x1128	TRGOCFG[SDM_PWM_SOC2]	Trigger manager output configure register	0x00000000
0x112C	TRGOCFG[SDM_PWM_SOC3]	Trigger manager output configure register	0x00000000
0x1130	TRGOCFG[SDM_PWM_SOC4]	Trigger manager output configure register	0x00000000
0x1134	TRGOCFG[SDM_PWM_SOC5]	Trigger manager output configure register	0x00000000
0x1138	TRGOCFG[SDM_PWM_SOC6]	Trigger manager output configure register	0x00000000
0x113C	TRGOCFG[SDM_PWM_SOC7]	Trigger manager output configure register	0x00000000
0x1140	TRGOCFG[SDM_PWM_SOC8]	Trigger manager output configure register	0x00000000
0x1144	TRGOCFG[SDM_PWM_SOC9]	Trigger manager output configure register	0x00000000
0x1148	TRGOCFG[SDM_PWM_SOC10]	Trigger manager output configure register	0x00000000
0x114C	TRGOCFG[SDM_PWM_SOC11]	Trigger manager output configure register	0x00000000
0x1150	TRGOCFG[SDM_PWM_SOC12]	Trigger manager output configure register	0x00000000
0x1154	TRGOCFG[SDM_PWM_SOC13]	Trigger manager output configure register	0x00000000
0x1158	TRGOCFG[SDM_PWM_SOC14]	Trigger manager output configure register	0x00000000
0x115C	TRGOCFG[SDM_PWM_SOC15]	Trigger manager output configure register	0x00000000
0x1160	TRGOCFG[PLB_IN_00]	Trigger manager output configure register	0x00000000
0x1164	TRGOCFG[PLB_IN_01]	Trigger manager output configure register	0x00000000
0x1168	TRGOCFG[PLB_IN_02]	Trigger manager output configure register	0x00000000
0x116C	TRGOCFG[PLB_IN_03]	Trigger manager output configure register	0x00000000
0x1170	TRGOCFG[PLB_IN_04]	Trigger manager output configure register	0x00000000
0x1174	TRGOCFG[PLB_IN_05]	Trigger manager output configure register	0x00000000

Offset	Register Name	Description	Reset Value
0x1178	TRGOCFG[PLB_IN_06]	Trigger manager output configure register	0x00000000
0x117C	TRGOCFG[PLB_IN_07]	Trigger manager output configure register	0x00000000
0x1180	TRGOCFG[PLB_IN_08]	Trigger manager output configure register	0x00000000
0x1184	TRGOCFG[PLB_IN_09]	Trigger manager output configure register	0x00000000
0x1188	TRGOCFG[PLB_IN_10]	Trigger manager output configure register	0x00000000
0x118C	TRGOCFG[PLB_IN_11]	Trigger manager output configure register	0x00000000
0x1190	TRGOCFG[PLB_IN_12]	Trigger manager output configure register	0x00000000
0x1194	TRGOCFG[PLB_IN_13]	Trigger manager output configure register	0x00000000
0x1198	TRGOCFG[PLB_IN_14]	Trigger manager output configure register	0x00000000
0x119C	TRGOCFG[PLB_IN_15]	Trigger manager output configure register	0x00000000
0x11A0	TRGOCFG[PLB_IN_16]	Trigger manager output configure register	0x00000000
0x11A4	TRGOCFG[PLB_IN_17]	Trigger manager output configure register	0x00000000
0x11A8	TRGOCFG[PLB_IN_18]	Trigger manager output configure register	0x00000000
0x11AC	TRGOCFG[PLB_IN_19]	Trigger manager output configure register	0x00000000
0x11B0	TRGOCFG[PLB_IN_20]	Trigger manager output configure register	0x00000000
0x11B4	TRGOCFG[PLB_IN_21]	Trigger manager output configure register	0x00000000
0x11B8	TRGOCFG[PLB_IN_22]	Trigger manager output configure register	0x00000000
0x11BC	TRGOCFG[PLB_IN_23]	Trigger manager output configure register	0x00000000
0x11C0	TRGOCFG[PLB_IN_24]	Trigger manager output configure register	0x00000000
0x11C4	TRGOCFG[PLB_IN_25]	Trigger manager output configure register	0x00000000

Offset	Register Name	Description	Reset Value
0x11C8	TRGOCFG[PLB_IN_26]	Trigger manager output configure register	0x00000000
0x11CC	TRGOCFG[PLB_IN_27]	Trigger manager output configure register	0x00000000
0x11D0	TRGOCFG[PLB_IN_28]	Trigger manager output configure register	0x00000000
0x11D4	TRGOCFG[PLB_IN_29]	Trigger manager output configure register	0x00000000
0x11D8	TRGOCFG[PLB_IN_30]	Trigger manager output configure register	0x00000000
0x11DC	TRGOCFG[PLB_IN_31]	Trigger manager output configure register	0x00000000
0x11E0	TRGOCFG[PLB_IN_32]	Trigger manager output configure register	0x00000000
0x11E4	TRGOCFG[PLB_IN_33]	Trigger manager output configure register	0x00000000
0x11E8	TRGOCFG[PLB_IN_34]	Trigger manager output configure register	0x00000000
0x11EC	TRGOCFG[PLB_IN_35]	Trigger manager output configure register	0x00000000
0x11F0	TRGOCFG[PLB_IN_36]	Trigger manager output configure register	0x00000000
0x11F4	TRGOCFG[PLB_IN_37]	Trigger manager output configure register	0x00000000
0x11F8	TRGOCFG[PLB_IN_38]	Trigger manager output configure register	0x00000000
0x11FC	TRGOCFG[PLB_IN_39]	Trigger manager output configure register	0x00000000
0x1200	TRGOCFG[PLB_IN_40]	Trigger manager output configure register	0x00000000
0x1204	TRGOCFG[PLB_IN_41]	Trigger manager output configure register	0x00000000
0x1208	TRGOCFG[PLB_IN_42]	Trigger manager output configure register	0x00000000
0x120C	TRGOCFG[PLB_IN_43]	Trigger manager output configure register	0x00000000
0x1210	TRGOCFG[PLB_IN_44]	Trigger manager output configure register	0x00000000
0x1214	TRGOCFG[PLB_IN_45]	Trigger manager output configure register	0x00000000

Offset	Register Name	Description	Reset Value
0x1218	TRGOCFG[PLB_IN_46]	Trigger manager output configure register	0x00000000
0x121C	TRGOCFG[PLB_IN_47]	Trigger manager output configure register	0x00000000
0x1220	TRGOCFG[PLB_IN_48]	Trigger manager output configure register	0x00000000
0x1224	TRGOCFG[PLB_IN_49]	Trigger manager output configure register	0x00000000
0x1228	TRGOCFG[PLB_IN_50]	Trigger manager output configure register	0x00000000
0x122C	TRGOCFG[PLB_IN_51]	Trigger manager output configure register	0x00000000
0x1230	TRGOCFG[PLB_IN_52]	Trigger manager output configure register	0x00000000
0x1234	TRGOCFG[PLB_IN_53]	Trigger manager output configure register	0x00000000
0x1238	TRGOCFG[PLB_IN_54]	Trigger manager output configure register	0x00000000
0x123C	TRGOCFG[PLB_IN_55]	Trigger manager output configure register	0x00000000
0x1240	TRGOCFG[PLB_IN_56]	Trigger manager output configure register	0x00000000
0x1244	TRGOCFG[PLB_IN_57]	Trigger manager output configure register	0x00000000
0x1248	TRGOCFG[PLB_IN_58]	Trigger manager output configure register	0x00000000
0x124C	TRGOCFG[PLB_IN_59]	Trigger manager output configure register	0x00000000
0x1250	TRGOCFG[PLB_IN_60]	Trigger manager output configure register	0x00000000
0x1254	TRGOCFG[PLB_IN_61]	Trigger manager output configure register	0x00000000
0x1258	TRGOCFG[PLB_IN_62]	Trigger manager output configure register	0x00000000
0x125C	TRGOCFG[PLB_IN_63]	Trigger manager output configure register	0x00000000
0x1260	TRGOCFG[SYNCTIMER_TRIG]	Trigger manager output configure register	0x00000000
0x1264	TRGOCFG[DCCS_TRIG]	Trigger manager output configure register	0x00000000

Offset	Register Name	Description	Reset Value
0x1268	TRGOCFG[TRGM_IRQ0]	Trigger manager output configure register	0x00000000
0x126C	TRGOCFG[TRGM_IRQ1]	Trigger manager output configure register	0x00000000
0x1270	TRGOCFG[TRGM_DMA0]	Trigger manager output configure register	0x00000000
0x1274	TRGOCFG[TRGM_DMA1]	Trigger manager output configure register	0x00000000
0x1278	TRGOCFG[ESC_TRIG_IN]	Trigger manager output configure register	0x00000000

Table 177: TRGM Register List

38.4 TRGM register description

TRGM are as follows:

38.4.1 FILTCFG (0x0 + 0x4 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RSVD															OUTINV	MODE				SYNCEN	FILTLEN_SHIFT				FILTLEN_BASE							
N/A															RW	RW				RW	RW				RW							
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

FILTCFG [31:0]

Bits	Name	Description
16	OUTINV	1- Filter will invert the output 0- Filter will not invert the output
15-13	MODE	This bitfields defines the filter mode 000-bypass; 100-rapid change mode; 101-delay filter mode; 110-stalbe low mode; 111-stable high mode
12	SYNCEN	set to enable sychronization input signal with TRGM clock
11-9	FILTLEN_SHIFT	
8-0	FILTLEN_BASE	This bitfields defines the filter counter length.

FILTCFG Bits

38.4.2 DMACFG (0x400 + 0x4 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAMUX_EN	RSVD																								DMASRCSEL						
RW	N/A																								RW						
0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0

DMACFG [31:0]

Bits	Name	Description
31	DMAMUX_EN	
5-0	DMASRCSEL	

DMACFG Bits

38.4.3 GCR (0x500)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRGOPEN																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

GCR [31:0]

Bits	Name	Description
31-0	TRGOPEN	

GCR Bits

38.4.4 ADC_MATRIX_SEL0 (0x540)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QE11_ADC1_SEL								QE11_ADC0_SEL								QE10_ADC1_SEL								QE10_ADC0_SEL							
RW								RW								RW								RW							
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

ADC_MATRIX_SEL0 [31:0]

Bits	Name	Description
31-24	QE11_ADC1_SEL	
23-16	QE11_ADC0_SEL	
15-8	QE10_ADC1_SEL	
7-0	QE10_ADC0_SEL	

ADC_MATRIX_SEL0 Bits

38.4.5 DAC_MATRIX_SEL0 (0x580)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWM0_DAC1_SEL								PWM0_DAC0_SEL								ACMP0_CH1_DAC_SEL								ACMP0_CH0_DAC_SEL							
RW								RW								RW								RW							
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DAC_MATRIX_SEL0 [31:0]

Bits	Name	Description
31-24	PWM0_DAC1_SE L	
23-16	PWM0_DAC0_SE L	
15-8	ACMP0_CH1_DA C_SEL	
7-0	ACMP0_CH0_DA C_SEL	

DAC_MATRIX_SEL0 Bits

38.4.6 DAC_MATRIX_SEL1 (0x584)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWM1_DAC1_SEL								PWM1_DAC0_SEL								PWM0_DAC3_SEL								PWM0_DAC2_SEL							
RW								RW								RW								RW							
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

DAC_MATRIX_SEL1 [31:0]

Bits	Name	Description
31-24	PWM1_DAC1_SE L	
23-16	PWM1_DAC0_SE L	
15-8	PWM0_DAC3_SE L	
7-0	PWM0_DAC2_SE L	

DAC_MATRIX_SEL1 Bits

38.4.7 DAC_MATRIX_SEL2 (0x588)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																PWM1_DAC3_SEL								PWM1_DAC2_SEL							
N/A																RW								RW							
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DAC_MATRIX_SEL2 [31:0]

Bits	Name	Description
15-8	PWM1_DAC3_SE L	
7-0	PWM1_DAC2_SE L	

DAC_MATRIX_SEL2 Bits

38.4.8 POS_MATRIX_SEL0 (0x5C0)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																QE01_POSIN_SEL								QE00_POSIN_SEL							

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
N/A																RW								RW							
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

POS_MATRIX_SEL0 [31:0]

Bits	Name	Description
15-8	QEO1_POSIN_SE L	
7-0	QEO0_POSIN_SE L	

POS_MATRIX_SEL0 Bits

38.4.9 TRGM_IN (0x600 + 0x4 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRGM_IN																RO															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TRGM_IN [31:0]

Bits	Name	Description
31-0	TRGM_IN	

TRGM_IN Bits

38.4.10 TRGM_OUT (0x680 + 0x4 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRGM_OUT																RO															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TRGM_OUT [31:0]

Bits	Name	Description
31-0	TRGM_OUT	

TRGM_OUT Bits

38.4.11 PWM_DELAY_CFG (0x800)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																										DELAY_CHAN_CALIB_SW					
N/A																										RW					
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	

PWM_DELAY_CFG [31:0]

Bits	Name	Description
5-0	DELAY_CHAN_CALIB_SW	

PWM_DELAY_CFG Bits

38.4.12 PWM_CALIB_CFG (0x804)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																CALIB_SW_START	RSVD										CALIB_PERIOD				
N/A																RW	N/A										RW				
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0

PWM_CALIB_CFG [31:0]

Bits	Name	Description
15	CALIB_SW_START	
4-0	CALIB_PERIOD	

PWM_CALIB_CFG Bits

38.4.13 TRGOCFG (0x1000 + 0x4 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD													OUTINV	FEDG2PEN	REDG2PEN	RSVD							TRIGOSEL								
N/A													RW	RW	RW	N/A							RW								
x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0

TRGOCFG [31:0]

Bits	Name	Description
18	OUTINV	1- Invert the output
17	FEDG2PEN	1- The selected input signal falling edge will be convert to an pulse on output. The output pulse can be stably used within the motor control system. When connecting the signal outside the motor system, due to the asynchronization of the clock systems, the clock frequency and signal active length need to be considered.
16	REDG2PEN	1- The selected input signal rising edge will be convert to an pulse on output. The output pulse can be stably used within the motor control system. When connecting the signal outside the motor system, due to the asynchronization of the clock systems, the clock frequency and signal active length need to be considered.
7-0	TRIGOSEL	This bitfield selects one of the TRGM inputs as output.

TRGOCFG Bits

39 Sync Timer SYNT

This chapter describes the main functions and features of the synchronous timer SYNT.

39.1 Feature Summary

This chapter describes the main features of the synchronous timer SYNT:

- supports output synchronization events
 - 32-bit counter and reload register
 - supports 16 channels, each channel supports a comparison register
 - supports Oneshot mode
- supports outputting a unified timestamp to the motion control system
 - 32-bit counter generates timestamp
 - supports recording the timestamp when an event is triggered
 - Support software update timestamp
- supports output DCCS_CLK to adjust DCDC voltage conversion timing and optimize ADC sampling performance
 - supports fixed ADC operating frequency mode
 - supports fixed DCCS_CLK frequency mode

39.2 Function Description

synchronization timer, which are used to generate synchronization events and generate timestamps respectively.

39.2.1 Output synchronization events

The synchronization timer uses a 32-bit counter to generate synchronization events, and the user can configure this counter overload value by overloading the register RLD.

When in use, the counter is enabled by setting the GCR[CEN] bit to 1. After enabling, the counter always starts counting from 0. When the value of the counter reaches the value of the overload register RLD, the counter is restored to 0 and continues counting. However, if oneshot mode is enabled through the GCR[TIMER_ONESHOT] register, the counter will stop counting after reaching the overloaded value, and the software needs to reset the counter before continuing to count again.

By setting the GCR[CRST] position to 1, the user can reset the counter to 0. After the reset is completed, GCR[CRST] needs to be cleared manually. If the GCR[COUNTER_DEBUG_EN] bit field is enabled, the counter will stop counting in debug mode.

The synchronous timer supports 16 channels, each channel having a corresponding 32-bit comparison register CMPx. When the counter reaches the comparison register value, a comparison event with a length of one synchronous timer clock pulse is output. This event can be connected to other modules of the motor system through the interconnection manager, such as PWM, QEI, etc., for synchronization. For example,

can allow four motor systems to start running accurately in the same clock cycle by connecting to the synchronous trigger input (SYNCI) of PWM, and can also accurately adjust the phase difference of the four motor systems.

Note: The synchronous timer is designed for synchronization between on-chip motor systems. The working clock is synchronized with the motor system and cannot be used for peripherals in other clock domains.

The synchronous timer does not support generating interrupts or DMA requests.

39.2.2 Output Timestamp

synchronous timer can output a 32-bit timestamp to other modules in the motion control system (motor system), providing a unified time for collaborative work between different modules in the motion control system. This timestamp is generated by a 32-bit counter.

When in use, the counter is enabled by setting the GCR[TIMESTAMP_ENABLE] bit to 1. After enabling, the counter always starts counting from 0. When the value of the counter reaches 0xFF_FFFF, the counter will automatically return to 0 and continue counting.

By setting the GCR[TIMESTAMP_RESET] bit to 1, the user can reset the counter to 0, and GCR[TIMESTAMP_RESET] will automatically clear after writing 1. If the GCR[TIMESTAMP_DEBUG_EN] bit field is enabled, the counter will stop counting in debug mode.

Users can update the value of this counter through software. Setting GCR[TIMESTAMP_SET_NEW] to 1 can directly update the counter value (timestamp) to the value written in the TIMESTAMP_NEW register; setting GCR[TIMESTAMP_DEC_NEW] to 1 can update the counter value to the current counter value minus the value written in the TIMESTAMP_NEW register; setting GCR[TIMESTAMP_INC_NEW] to 1 can update the counter value to the current counter value plus the value written in the TIMESTAMP_NEW register.

synchronous timer also supports recording the timestamp when an event is triggered. By configuring the interconnection manager to select events from the motion control system as triggers for SYNT, triggering the synchronous timer to save the current counter's timestamp in the TIMESTAMP_SAV register.

39.2.3 Outputs DCCS_CLK to DCDC

DCDC module provides different duty cycles through the DCDC_SWITCH signal to realize the function of converting 3.3V to 1.1V. If the ADC sampling window ends too close to the edge of DCDC_SWITCH change (order of magnitude: tens of nanoseconds), it will cause inaccurate ADC sampling and affect ADC performance. Therefore, the synchronous timer SYNT can output the DCCS_CLK clock to the DCCC through the internal DCCC (DCCC Clock Control System) function module to adjust the DCDC voltage conversion timing, thereby optimizing the sampling performance of the ADC.

The DCCS adjusts the rising edge time of DCCS_SWITCH by varying the period of DCCS_CLK, thereby avoiding the transition point of DCCS_SWITCH for the end time of the ADC sampling window. DCCS provides a clock DCCS_CLK with a frequency between 12 and 40MHz to the DCDC. Based on this clock, the DCDC outputs a cycle of the DCDC_SWITCH signal every 16 clock cycles. When the duty ratio is 1:2, a corresponding voltage of 1.1V (3.3V/3) is output.

39.2.3.1 Fixed ADC operating frequency mode

In this mode, the ADC works at a fixed frequency, and the DCCS module adjusts the DCCS_CLK to make the DCCS_SWITCH cycle consistent with the ADC cycle. Because the DCCS_CLK frequency is limited, the minimum ADC frequency in this mode is 750KHz (corresponding to 12MHz DCCS_CLK) and the maximum is 2.5 MHz (the actual maximum operating frequency of a 16-bit ADC is 2MHz, and it can work at 2.5 M at room

temperature).

Software configuration process:

- Configure DCDC, configure DCDC working mode as basic mode, and select the clock source of DCDC as RC24M;
- Configure the ADC and interconnection manager TRGM (generates a trigger signal and sends it to the DCCS module)
- Configure DCDC_PRD (configured as ADC period, unit is system clock period) and SAMPLE_DIS (unit is system time period) from the trigger signal to the ADC sampling end time point. It is recommended to configure it to be about half the size of DCCS_PRD. Too large (greater than DCCS_PRD8) or too small (less than 8) may cause the DCCS to function incorrectly.
- is configured in fixed ADC frequency mode (CFG0.MODE=1)
- Enable DCCS (CFG0.ENABLE=1)

39.2.3.2 Fixed DCCS_CLK frequency mode

In this mode, DCCS_CLK operates at 25MHz at steady state (configured DCDC_PRD=128). The interconnection manager TRGM needs to give a trigger signal (configuration SAMPLE_DIS=128) one DCDC_PRD cycle before the ADC sampling end point. The DCCS module determines that the DCCS_CLK needs to be adjusted faster or slower based on this trigger signal.

This mode is used for long ADC conversion intervals (greater than 640ns) and does not require a fixed ADC frequency.

Software configuration process:

- Configure DCDC
- Configure the ADC and interconnection manager TRGM (generates a trigger signal and sends it to the DCCS module)
- Configure DCDC_PRD=128 and configure SAMPLE_DIS=128.
- is configured in fixed adc frequency mode (CFG0.MODE=0)
- Enable DCCS (CFG0.ENABLE=1)

39.3 SYNT register list

SYNT is as follows:

SYNT base address: 0xF0464000

Offset	Register Name	Description	Reset Value
0x0000	GCR	Global control register	0x00000000
0x0004	RLD	Counter reload register	0x00000000
0x0008	TIMESTAMP_NEW	timestamp new value register	0x00000000
0x000C	CNT	Counter	0x00000000
0x0010	TIMESTAMP_SAV	timestamp trig save value	0x00000000

Offset	Register Name	Description	Reset Value
0x0014	TIMESTAMP_CUR	timestamp read value	0x00000000
0x0020	CMP[0]	Comparator	0x00000000
0x0024	CMP[1]	Comparator	0x00000000
0x0028	CMP[2]	Comparator	0x00000000
0x002C	CMP[3]	Comparator	0x00000000
0x0030	CMP[4]	Comparator	0x00000000
0x0034	CMP[5]	Comparator	0x00000000
0x0038	CMP[6]	Comparator	0x00000000
0x003C	CMP[7]	Comparator	0x00000000
0x0040	CMP[8]	Comparator	0x00000000
0x0044	CMP[9]	Comparator	0x00000000
0x0048	CMP[10]	Comparator	0x00000000
0x004C	CMP[11]	Comparator	0x00000000
0x0050	CMP[12]	Comparator	0x00000000
0x0054	CMP[13]	Comparator	0x00000000
0x0058	CMP[14]	Comparator	0x00000000
0x005C	CMP[15]	Comparator	0x00000000

Table 178: SYNT Register List

39.4 SYNT register details

SYNT are as follows:

39.4.1 GCR (0x0)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMESTAMP_INC_NEW	TIMESTAMP_DEC_NEW	TIMESTAMP_SET_NEW	TIMESTAMP_RESET	RSVD																						TIMESTAMP_DEBUG_EN	TIMESTAMP_ENABLE	TIMER_ONESHOT	COUNTER_DEBUG_EN	CRST	CEN
WO	WO	WO	WO	N/A																						RW	RW	RW	RW	RW	RW
0	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0

GCR [31:0]

Bits	Name	Description
31	TIMESTAMP_INC_NEW	set to increase the timesamp with new value, auto clr

Bits	Name	Description
30	TIMESTAMP_DEC_NEW	set to decrease the timesamp with new value, auto clr
29	TIMESTAMP_SET_NEW	set the timesamp to new value, auto clr
28	TIMESTAMP_RESET	reset timesamp to 0, auto clr
5	TIMESTAMP_DEBUG_EN	set to enable cpu_debug_mode to stop the timesamp
4	TIMESTAMP_ENABLE	set to enable the timesamp , clr to stop
3	TIMER_ONESHOT	set for one shot mode, timer will run only once. Software need reset timer first to start another one shot
2	COUNTER_DEBUG_EN	set to enable cpu_debug_mode to stop the counter
1	CRST	1- Reset counter
0	CEN	1- Enable counter

GCR Bits

39.4.2 RLD (0x4)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RLD																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RLD [31:0]

Bits	Name	Description
31-0	RLD	counter reload value

RLD Bits

39.4.3 TIMESTAMP_NEW (0x8)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TIMESTAMP_NEW [31:0]

Bits	Name	Description
31-0	VALUE	new value for timesamp , can be used as set/inc/dec

TIMESTAMP_NEW Bits

39.4.4 CNT (0xC)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT																															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CNT [31:0]

Bits	Name	Description
31-0	CNT	counter

CNT Bits

39.4.5 TIMESTAMP_SAV (0x10)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TIMESTAMP_SAV [31:0]

Bits	Name	Description
31-0	VALUE	use the trigger to save timesamp here

TIMESTAMP_SAV Bits

39.4.6 TIMESTAMP_CUR (0x14)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TIMESTAMP_CUR [31:0]

Bits	Name	Description
31-0	VALUE	current timesamp value

TIMESTAMP_CUR Bits

39.4.7 CMP (0x20 + 0x4 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

CMP [31:0]

Bits	Name	Description
31-0	CMP	comparator value, the output will assert when counter count to this value

CMP Bits

40 Quadrature Encoder Interface QEIV2

Based on the input signals, it measures speed and generates position information.

40.1 Feature Summary

This chapter introduces the main features of the Quadrature Encoder Interface QEIV2:

- Supports single-phase, two-phase (quadrature/up-down/direction pulse), and three-phase (120° phase difference) square wave inputs;
- Supports index signal INDEX/Z, which can generate interrupts, reset the line count, change the revolution count
- Supports two homing signals HOME, generating interrupts only
- Supports single-phase and two-phase quadrature sine wave inputs (via ADC bus)
- Supports amplitude/center value and angle correction for sine waves
- Supports up to 32-bit line count
- Supports two preset position-triggered interrupts, configurable for specified revolutions, line counts, direction, and position
- Supports software-triggered latching, event-triggered latching, and two external trigger latches (including revolution, line count, direction, angle, position)
- Supports two speed measurement counters based on pulse count, saving two measurement results, and recording measurement time
- Supports two speed measurement counters based on time duration, saving two measurement results, and recording measurement time
- Supports converting position to binary position, outputting to the position manager via the position bus (including position acquisition time)
- Allows software modification of position, line counter, revolution counter (absolute mode only), supporting incremental and absolute modes
- Supports watchdog, supports stall detection
- All interrupt signals can also be configured to generate DMA requests, trigger latching, or output signals

40.2 Block Diagram

The internal structure of the QEI module is shown in the figure below. The connection relationships with external modules are introduced in the overview chapter of the electric control system.

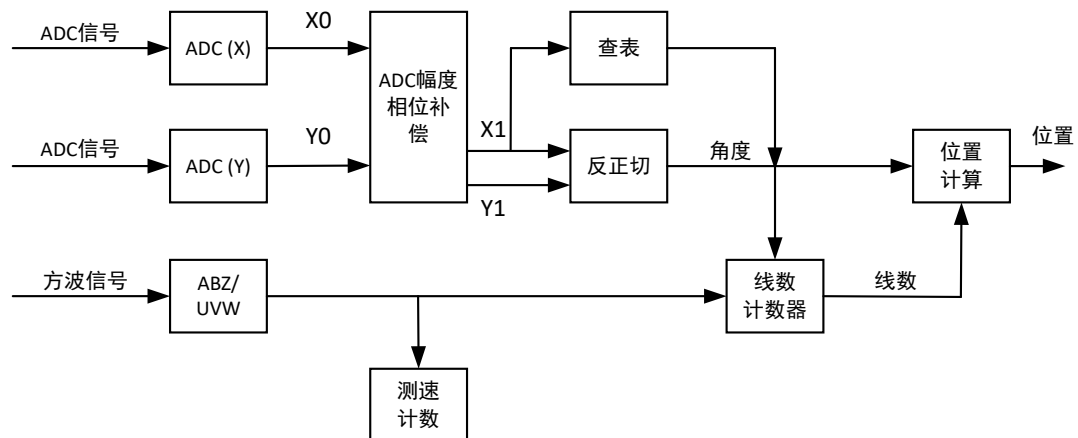


Figure 41: QEI Architecture Diagram

40.3 Pin Description

Pin	Direction	Functional Description
QEI_A	Input	Phase A signal, can also be used as phase U signal for UVW.
QEI_B	Input	Phase B signal, can also be used as phase V signal for UVW.
QEI_Z	Input	Phase Z signal, can also be used as phase W signal for UVW.
QEI_F	Input	Fault signal. When active, it can cut off position information output and generate an interrupt.
QEI_H0	Input	Home0 signal, can generate an interrupt.
QEI_H1	Input	Home1 signal, can generate an interrupt.

Table 179: QEI Pin Description

40.4 Functional Description

40.4.1 QEIV2 Decoding Modes

The Quadrature Encoder Interface QEIV2 module (hereafter referred to as the QEI module) for square wave encoding has three input signals: A, B, and Z. The encoding logic interprets the motor rotor's rotation information based on these three input signals.

The quadrature encoder interface supports single-phase, two-phase (quadrature/up-down/direction pulse), three-phase (120° phase difference) square wave, and sine-cosine wave decoding modes, among others. Users can select the corresponding decoding mode in the [ENCTYP] bit field of the Control Register (CR) according to their needs.

40.4.1.1 Single-Phase Digital Mode

For the single-phase digital mode, only the Phase A signal is required as input, but the user must know the motor rotor's rotation direction in advance.

The user needs to configure the [ENCTYP] bit field in the Control Register (CR) to 4, and configure the [ZCNTCFG] bit field in the Control Register (CR) according to the situation of the Z phase. Additionally, the motor's line count and phase calculation parameters need to be configured in the Phase Configuration Register (PHCFG) and the Phase Calculation Parameter Control Register (PHASE_PARAM).

40.4.1.2 ABZ Mode

In ABZ mode, inputs include Phase A and Phase B signals (AB phases) and optionally Phase A, B, and Z signals (ABZ phases). For the two-phase quadrature signals AB, when Phase A leads Phase B, it indicates the motor is rotating in the forward direction; when Phase A lags Phase B, it indicates the motor is rotating in the reverse direction. When the quadrature encoder interface receives square waves indicating forward rotation, a signal edge will cause the AB phase counter (COUNT[CURRENT][PH] register, offset address 0x0034) to increment by 1, and the AB phase counter (PHASE_CNT) will also increment by 1. If it receives square waves indicating reverse rotation, a signal edge will cause the AB phase counter to decrement by 1, and the AB phase counter (PHASE_CNT) will also decrement by 1. This is illustrated in the figures below. The motor's rotation direction and the input states of the AB phases can be obtained from the [DIR] bit field in the speed register (COUNT[CURRENT][SPD]).

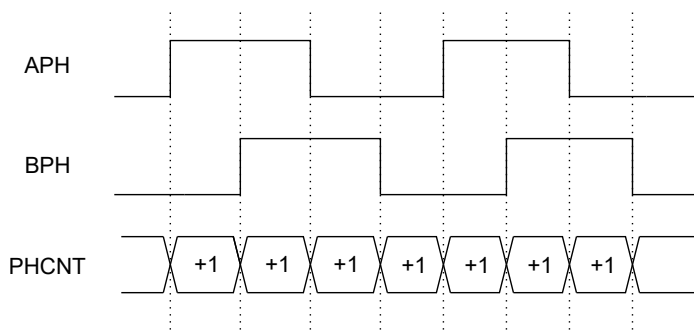


Figure 42: Forward Rotation Phase Counting Diagram

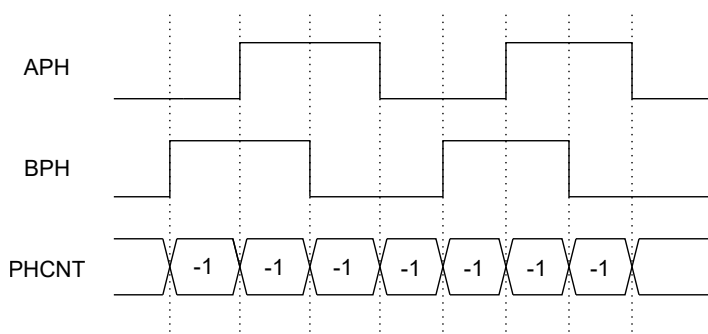


Figure 43: Reverse Rotation Phase Counting Diagram

The user can set the Phase Configuration Register (PHCFG) according to the motor and quadrature encoder; this value represents the total number of PHASE counts for one complete motor revolution.

Whenever the AB phase counter reaches PHMAX while counting forward, the AB phase counter resets to 0.

Whenever the AB phase counter reaches 0 while counting backward, the AB phase counter resets to the user-set motor line count value (i.e., the [PHMAX] value in the Phase Configuration Register PHCFG).

The Z phase and its counter in ABZ mode have the following characteristics:

1. Z Phase Signal Not Used

When the Z phase signal is not used, the Z phase pin should not be configured; only the A and B phase signal pins should be configured. Otherwise, the ZCNT may increase or decrease abnormally. When the Z phase signal is absent, the [ZCNTCFG] bit field in the Control Register (CR) can be configured so that the Z phase count increments by 1 when the Phcnt count reaches Phmax, or decrements by 1 when the phcnt count passes 0.

2. Z Phase Signal Used

The Z phase is often called the index phase. When the AB phases indicate forward motor rotation, upon detecting the rising edge of the Z phase, the Z phase counter increments by 1. When the AB phases indicate reverse motor rotation, upon detecting the falling edge of the Z phase, the Z phase counter decrements by 1. The motor outputs one pulse per revolution on the Z phase, so this signal can be used to locate the zero point of each motor revolution.

The Z phase has the following alignment modes:

- The Z phase signal is always aligned with an edge of Phase A
- The Z phase signal is independent, ignoring alignment with A and B phase edges

1) When the Z phase is always aligned with an edge of Phase A, there are two cases:

- When the motor rotates forward, the Z phase signal detection is always synchronized with the rising edge of Phase A. That is, when the Z phase signal is high and a rising edge of Phase A is detected, the Z phase counter increments by 1.
- When the motor rotates reverse, the Z phase signal detection is always synchronized with the falling edge of Phase A. That is, when the Z phase signal is high and a falling edge of Phase A is detected, the Z phase counter decrements by 1.

The two cases are illustrated below.

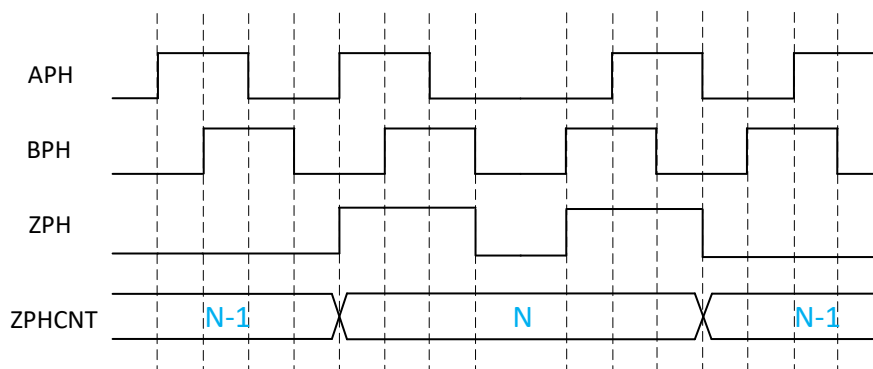


Figure 44: Z Phase Aligned with Phase A Edge Diagram

2) When the Z phase signal is independent, i.e., the Z_ONLY_EN bit in the CR register is enabled:

- When the motor rotates forward, upon detecting a rising edge of the Z phase, the Z phase counter increments by 1.
- When the motor rotates reverse, upon detecting a falling edge of the Z phase, the Z phase counter decrements by 1.

Furthermore, the Z phase signal can be used to calibrate the phase counter. The user needs to set the

value in the Phase Configuration Register (PHCFG), which represents the Z phase calibration value. The Z phase calibration function takes effect only after enabling the [PHCALIZ] bit field in the Control Register (CR). The value in the PHCFG register can be set to 0, indicating the Z phase signal is the zero point of motor rotation. It can also be set to any non-zero value, indicating the Z phase signal has an offset relative to the zero position. Therefore, when the Z phase is present, there are two cases:

- When the motor rotates forward, upon capturing a rising edge of the Z phase, reset $\text{PHCNT} = \text{PHIDX}$.
- When the motor rotates reverse, upon capturing a falling edge of the Z phase, reset $\text{PHCNT} = \text{PHIDX} - 1$. Note: if PHIDX is set to 0, this resets PHCNT to the phase overflow value PHMAX.

40.4.1.3 PD Mode

For PD mode, the Phase A input pin is used for the P-phase pulse signal, and the Phase B input pin is used for the D-phase pulse signal. The D-phase indicates the motor rotation direction: high level indicates forward rotation; otherwise, it indicates reverse rotation. During forward rotation, each edge of the P-phase increments the phase counter by 1. During reverse rotation, each edge of the P-phase decrements the phase counter by 1.

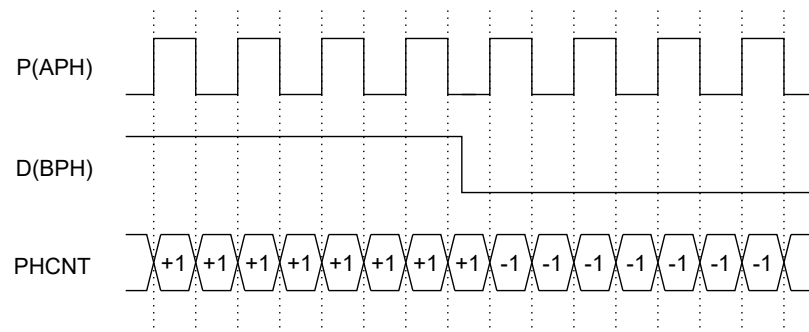


Figure 45: PD Mode Diagram

40.4.1.4 UD Mode

For UD mode, the Phase A input pin is used for the U-phase pulse signal, and the Phase B input pin is used for the D-phase pulse signal. Each edge of the U-phase input, indicating forward motor rotation, increments the phase counter by 1. Each edge of the D-phase input, indicating reverse motor rotation, decrements the phase counter by 1.

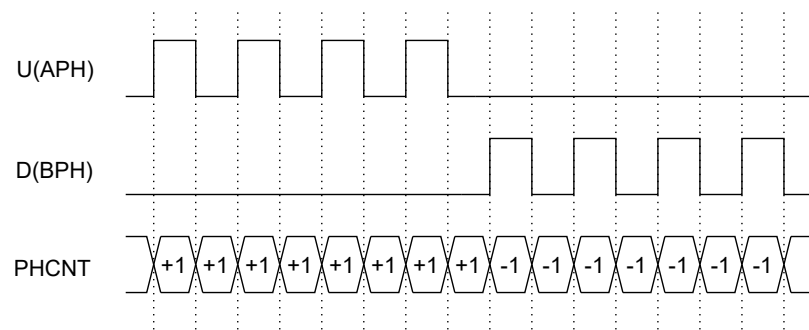


Figure 46: Up-Down (UD) Decoding Mode Diagram

40.4.1.5 UVW Mode

For UVW mode, the Phase A input pin is used for the U-phase pulse signal, the Phase B input pin is used for the V-phase pulse signal, and the Phase Z input pin is used for the W-phase pulse signal.

One cycle of one of these three input signals corresponds to one revolution of the motor rotor. The diagram below shows forward rotation with a 120° phase difference between the three signals (Note: this module requires the three signals to have a 120° phase difference). Based on these three input signals, they can be divided into 6 phases. In this mode, the QEI module's position information output options include outputting precise position information at the signal edge and outputting the position for the next position cycle at the signal edge. The QEI module's position information output option (i.e., the value of the [UVW_POS_OPT0] bit field in the QEI Configuration Register (QEI_CFG)) affects the configuration of the UVW positions (i.e., the UVW Position 0 Configuration Register (UVW_POS_CFG[UVW_POS0_CFG]), UVW Position 1 Configuration Register (UVW_POS_CFG[UVW_POS1_CFG]), UVW Position 2 Configuration Register (UVW_POS_CFG[UVW_POS2_CFG]), UVW Position 3 Configuration Register (UVW_POS_CFG[UVW_POS3_CFG]), UVW Position 4 Configuration Register (UVW_POS_CFG[UVW_POS4_CFG]), UVW Position 5 Configuration Register (UVW_POS_CFG[UVW_POS5_CFG])).

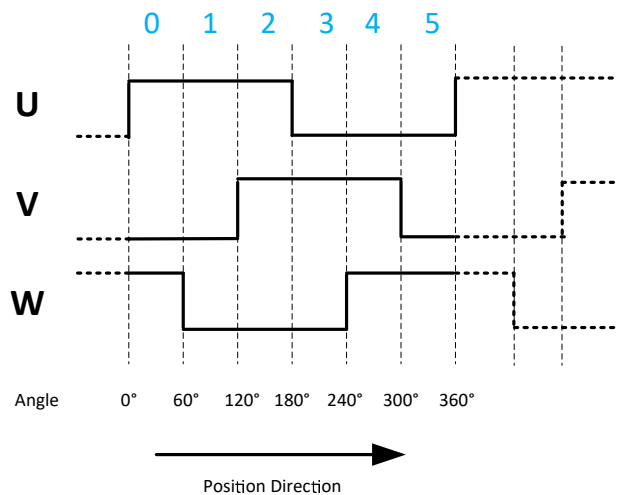


Figure 47: UVW Three-Phase Signal Position Encoding Diagram

Based on the UVW signals in the figure above, when the QEI module's position information output option is to output precise position at the signal edge (i.e., the [UVW_POS_OPT0] bit field in the QEI Configuration Register (QEI_CFG) is 0), the configurations for phases 0 to 5 from the UVW Position 0 Configuration Register to the UVW Position 5 Configuration Register are as follows:

Phase	0	1	2	3	4	5
U Phase Position Select (Config Value)	1	1	2	0	0	2
V Phase Position Select (Config Value)	0	2	1	1	2	0
W Phase Position Select (Config Value)	2	0	0	2	1	1

Table 180: UVW Phase Position Selection 0

Based on the UVW signals in the figure above, when the QEI module's position information output option is to output the next cycle's position at the signal edge (i.e., the [UVW_POS_OPT0] bit field in the QEI Configuration Register (QEI_CFG) is 1), the configurations for phases 0 to 5 from the UVW Position 0 Configuration Register to the UVW Position 5 Configuration Register are as follows:

Phase	0	1	2	3	4
5					
U Phase Position Select (Config Value)	3	3	3	0	0
0					
V Phase Position Select (Config Value)	0	0	3	3	3
0					
W Phase Position Select (Config Value)	3	0	0	0	3
3					

Table 181: UVW Phase Position Selection 1

40.4.1.6 Quadrature Sine-Cosine Wave Mode

The QEI module supports sine-cosine wave input from the ADC. For sine wave input, the signal comes from the chip's built-in ADC. The user can configure the source of the sine and cosine channels in the X-channel ADC Configuration Register 0 (ADCX_CFG0) and Y-channel ADC Configuration Register 0 (ADCY_CFG0), i.e., specify which ADC and which channel the sine and cosine paths come from. Note: if both sine and cosine come from the same ADC, the signal delay between the two paths will cause a larger angle calculation deviation.

Since ADC inputs are unsigned numbers, an offset needs to be configured in the X-channel ADC Configuration Register 2 (ADCX_CFG2) and Y-channel ADC Configuration Register 2 (ADCY_CFG2). The ADC value subtracts the offset, converting the input unsigned number to a signed number.

If the input sine and cosine signals are not perfectly orthogonal or have different amplitudes, parameters can be configured in the X-channel ADC Configuration Register 1 (ADCX_CFG1) or the Y-channel ADC Configuration Register 1 (ADCY_CFG1) for correction and compensation.

If the sine wave signal has a phase shift of B, i.e., the input is $Y0=\sin(A+B)$, $X0=\cos A$, parameters need to be configured to obtain $\sin A$ and $\cos A$. The principle of parameter correction and compensation is illustrated in the figure below. The specific adjustment calculation process is:

$$\begin{aligned}
 \sin A &= \sin((A + B) - B) \\
 &= \sin(A + B) * \cos B - \cos(A + B) * \sin B \\
 &= \sin(A + B) * \cos B - (\cos A * \cos B * \sin B - \sin A * \sin B * \sin B) \\
 \text{Rearranging gives } \sin A * (1 - (\sin B)^2) &= \sin(A + B) * \cos B - \cos A * \cos B * \sin B \\
 \text{Thus } \sin A * (\cos B)^2 &= \sin(A + B) * \cos B - \cos A * \cos B * \sin B \\
 \text{We can get } \sin A &= \sin(A + B) / \cos B + \cos A * (-\tan B)
 \end{aligned}$$

According to the above, x1 and y1 can be processed as:

$$x1 = (x0 - x\_offset) * x\_param0 + (y0 - y\_offset) * y\_param0 \quad (2)$$

$$y1 = (x0 - x\_offset) * x\_param1 + (y0 - y\_offset) * y\_param1 \quad (3)$$

When $x\_param0 = 1$, $y\_param0 = 0$, x1 is the unchanged $\cos(A)$ signal. When $x\_param1 = -\tan(B)$, $y\_param1 = 1/\cos(B)$, y1 is the $\sin(A)$ signal with the deviation angle B eliminated. Multiplying x_param0 and y_param0 by a coefficient can scale the x1 value. Multiplying x_param1 and y_param1 by a coefficient can scale the y1 value.

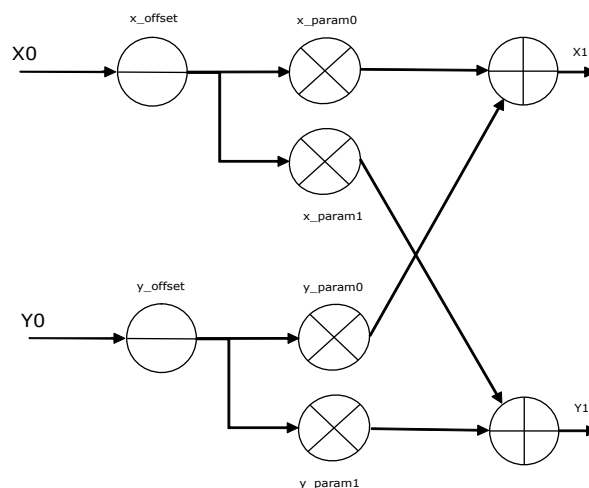


Figure 48: ADC Amplitude/Phase Compensation

Special Note: An ADC amplitude that is too small will result in imprecise calculated position. The closer the maximum positive amplitude is to 0xFFFFFFFF, the more precise the result.

40.4.1.7 Single-Phase Sine Wave Mode

If using the single-phase sine wave mode, the sine wave signal can only be input from the X channel. The input sine wave signal in this mode does not need to distinguish between sine or cosine. The QEI module calculates the position information from the input single-phase sine wave signal. If the ADC amplitude of the input single-phase sine wave is too small, the calculated position result will be imprecise. The closer the maximum positive amplitude of the input single-phase sine wave signal is to 0xFFFFFFFF, the more precise the result.

The software sets the sine wave mode via the CR[ENCTYP] bit (address 0x0) and configures PHCFG (address 0x4) and PHASE_PARAM (address 0x230) according to the actual external input for hardware calculation of position and revolution count.

For example, for a maximum line count of 128 lines, $PHCFG = (128-1)$, and the value for PHASE_PARAM is the result of $2^3 \text{ divided by } 128 (\text{rounded down})$.

40.4.2 Position Information Output

The QEI module converts input signals such as ABZ signals, UVM signals, and sine wave signals into position (position), which can be output to other modules as input signals via the interconnect manager. The

output position is the current revolution position of the rotor. Additionally, the angle within the QEI module is the angle value obtained from the arctangent calculation over one cycle of the sine-cosine signal.

1. For sine wave input signals (single-phase sine wave and quadrature sine-cosine signals):

When sine wave signals are input, they come from the ADC module inside the chip. For quadrature sine-cosine signals, after amplitude and phase adjustment by the QEI module, the angle (angle) is calculated via an arctangent circuit, and then further processed by hardware to obtain the position (position). For a single sine wave, after amplitude adjustment by the QEI module, the angle is obtained from a lookup table, and then hardware processing yields the position.

2. For square wave input signals like ABZ and UVM signals:

The QEI module interprets them into position based on the specific mode and outputs it. The QEI module handles the line counter as follows: the line counter increments by 1 each time the calculated angle crosses 360° (the actual algorithm is: when the previous angle is in the fourth quadrant and the next angle is in the first quadrant, increment by 1), and resets to 0 upon reaching the maximum value. Therefore, based on the current line counter angle value and the maximum line count, the QEI module calculates the current position value:

$$\text{position} = \frac{\text{phase\_cnt} * 2^{32} + \text{angel}}{\text{phase\_total}} \quad (4)$$

Here, the line count (phase_cnt) is the integer part, the angle (angle) is the fractional part, and phase_total is the maximum line count.

For example, for a maximum line count of 128, the software needs to configure phase_param=0x02000000. If the current line counter is 16 and the current angle is 90° (0x40000000), then the current position is $16 * 0x02000000 + 0x40000000/128 = 0x20800000$.

40.4.3 Position Matching

The QEI module supports using position information as a match condition to generate corresponding interrupt events. The matching position information includes revolution count (ZCMP), phase (PHCMP), and speed (SPDCMP). These matching functions can be independently enabled. Please refer to the CR register for details. If the above matching functions are not independently enabled, they are in an AND relationship.

- Revolution match is when the current revolution count equals the preset value in the Z Comparator 2 Register (ZCMP2);
- Phase match is when the current phase equals the preset phase comparison value (AB Comparator 2 Register, PHCMP2);
- Speed match is when the current speed (value of SPDCNT) equals or exceeds the preset speed comparison value (set in the SPDCMP register);

40.4.4 H Phase and F Phase Functions

The H phase input, also called the HOME phase, represents the zero point position on some motors. The quadrature decoder supports setting the SR[HOME2FEN] flag to 1 in the following situations:

- Detecting an H rising edge during forward motor rotation, and CR[HRDIR0] is set to 1
- Detecting an H falling edge during forward motor rotation, and CR[HFDIR0] is set to 1
- Detecting an H rising edge during reverse motor rotation, and CR[HRDIR1] is set to 1
- Detecting an H falling edge during reverse motor rotation, and CR[HFDIR1] is set to 1

The F phase is called the FAULT phase. The user can set the [FAULTPOS] bit in the CR register to 1. When a fault occurs in the motor system, a rising edge input on the F phase causes the QEI module to stop outputting valid position signals. The user can configure the [FAULTE] bit in the IRQEN register to 1, enabling an interrupt to be generated upon detecting a rising edge on the F phase.

40.4.5 Simple Speed Measurement

The QEI module provides two speed measurement methods for square wave signals: one based on pulse count and one based on duration (clock cycles). Each has two counters that can operate independently with different parameters.

40.4.5.1 Based on Pulse Count

Select the required edges from the input square wave signals to generate pulses (can select rising or falling edges, or both, from A, B, Z signals, but not combinations like A rising and B falling).

Two different pulse counts can be configured (the diagram shows only one set of circuits). When a rising or falling edge of the input signal is detected, the pulse counter starts, incrementing by one per edge, and the period counter also starts, incrementing by one per clock cycle.

When the pulse counter equals the configured pulse count, a trigger signal is generated. The current period counter value is stored in Period Snapshot 0, and the previous Period Snapshot 0 is stored in Period Snapshot 1. The period counter and pulse counter are cleared and restart.

The software can read the two most recent speed measurement results from the two snapshot registers.

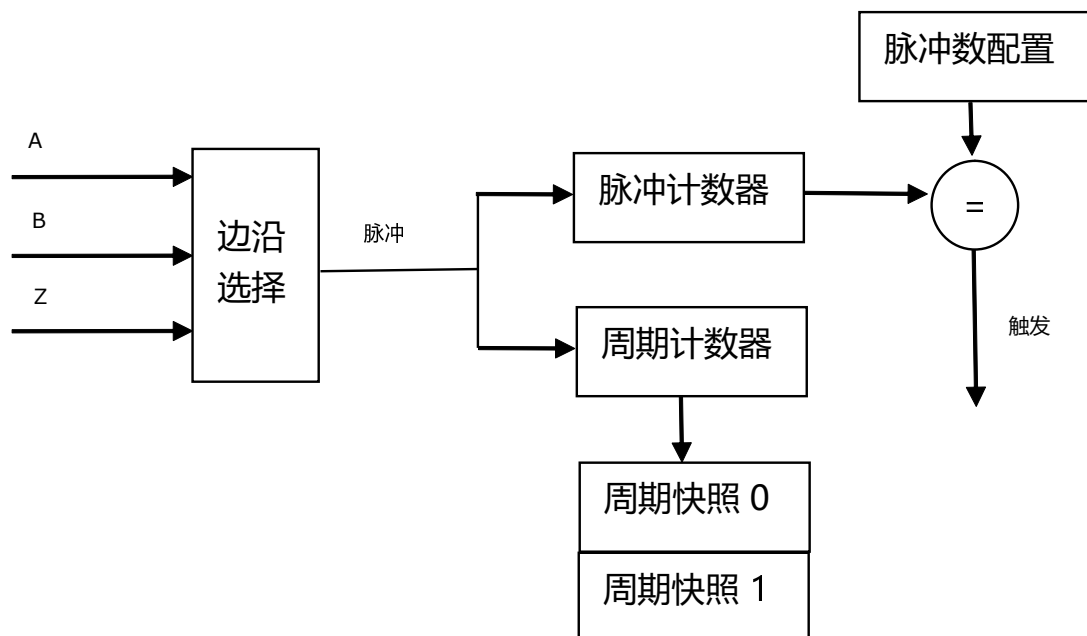


Figure 49: Pulse-based Speed Measurement

40.4.5.2 Based on Duration

When a pulse signal is detected, the pulse counter and period counter start. Each time the pulse counter increments, the current period counter value is stored in the period number.

When the period counter equals the configured period number, a trigger signal is generated. The current pulse counter value and period number are stored in Snapshot 0, and the previous Snapshot 0 is stored in Snapshot 1. The period counter and pulse counter are cleared and restart.

The software can read the two most recent speed measurement results from the two snapshot registers.

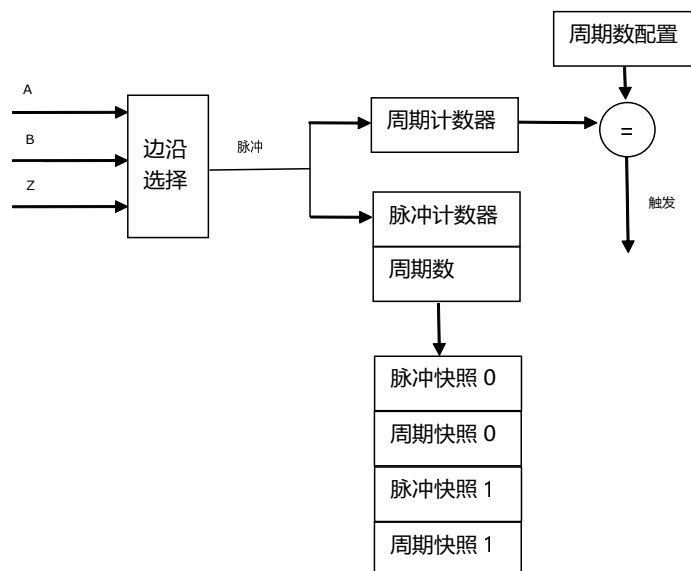


Figure 50: Duration-based Speed Measurement

40.4.6 Watchdog Counter

The watchdog counter is a timer that runs on the quadrature encoder interface clock, incrementing each clock cycle. The watchdog counter is cleared upon detecting any edge of the Phase A or Phase B input signals. The user can freely set the WDGCFG register to configure the timeout period. The purpose of the watchdog counter is to provide an alarm if the motor stalls for longer than a certain time.

The user sets the WDGCFG[WDGEN] bit to 1 to enable the watchdog counter. The timeout length is set via the WDGCFG[WDGTO] bit. When the watchdog count reaches the timeout, the SR[WDGF] flag is set.

When the user clears the SR[WDGF] flag by writing 1 to it, the watchdog counter is also cleared.

40.4.7 Counters Reading and Snapshot Function

The quadrature decoder interface has a dedicated read interface for its counters, making it convenient for users to read sufficient motor position and speed information at once. Users can read real-time information of various counters from the counter register group:

- COUNT[CURRENT][PH] register, the phase counter, and the current motor rotation direction and A/B phase input status.
- COUNT[CURRENT][Z] register, the Z phase counter.
- COUNT[CURRENT][SPD] register, the speed measurement counter, and the current motor rotation direction and A/B phase input status.

- COUNT[CURRENT][TMR], the timer.

Note: Position is the angle within one revolution, 0-360° corresponds to 0x00000000 to 0xFFFFFFFF. Angle is the arctangent angle within one signal cycle of a SinCos encoder. The CR[RD_SEL] bit field can be set to map the values to Position and Angle. When CR[RD_SEL] is 0, the SPD and TMR in the COUNT register are mapped to speed and timer values respectively. When CR[RD_SEL] is 1, the SPD and TMR in the COUNT register are mapped to position value (position) and angle value (angle) respectively.

The quadrature decoder supports saving the values of various counters to the following READ register group when a read event occurs. This ensures that the various counter values read by the user, and the position and speed information they reflect, are strictly from the same moment.

- COUNT[READ][PH] register, saves the phase counter information, and the current motor rotation direction and A/B phase input status.
- COUNT[READ][Z] register, saves the Z phase counter information.
- COUNT[READ][SPD] register, saves the speed measurement counter information, and the current motor rotation direction and A/B phase input status.
- COUNT[READ][TMR], saves the timer's timing information.

Read events can be:

- Software-triggered: the user sets the CR[READ] bit to 1 via software.
- Hardware-triggered: when the following flags are set to 1, and the corresponding hardware read bit (READEN) is also set to 1: (1) Watchdog counter timeout, SR[WDGF] flag set to 1; (2) HOME flag SR[HOMEF] set to 1; (3) ZPH flag SR[ZPHF] set to 1; (4) A position match event occurs, flag SR[POSCOMP] set to 1.

The user can combine hardware-triggered read events with quadrature decoder interrupts. For example, the user can enable interrupts for the Z phase signal via the IRQEN register, while also enabling hardware reading via the READEN register. When a flag triggers an interrupt, it simultaneously triggers a read event, saving the various counter groups to the READ register group. This way, the user can directly obtain the real-time information of various counters at the moment of the Z phase input edge in the interrupt service routine, avoiding data distortion caused by interrupt latency.

The snapshot trigger input SNAP1 comes from the interconnect manager. For details, see the interconnect manager chapter. The counter snapshot function is hardware-triggered. When the CR[SNAPEN] bit is set to 1, the quadrature decoder interface monitors the snapshot trigger input SNAP1. Upon detecting a rising edge on this input, it saves the values of various counters to the following snapshot (snap0/snap1) register groups. Like the READ register group, the various counter values in the snapshot register groups, and the position and speed information they reflect, are strictly from the same moment.

- COUNT[SNAP0][PH]/COUNT[SNAP1][PH] register, saves the phase counter information, and the current motor rotation direction and A/B phase input status.
- COUNT[SNAP0][Z]/COUNT[SNAP1][Z] register, saves the Z phase counter information.
- COUNT[SNAP0][SPD]/COUNT[SNAP1][SPD] register, and the current motor rotation direction and A/B phase input status.
- COUNT[SNAP0][TMR]/COUNT[SNAP1][TMR], saves the timer's timing information.

Note: When a rising edge is captured on the snapshot trigger input, the decoder hardware always transfers

the value of the snap0 register group to the snap1 register group, and saves the current counter values to the snap0 register group. Therefore, whenever a snapshot is triggered, the software can read the counter values at the time of this trigger from the snap0 register group, and read the counter values saved to the snap0 register group at the previous snapshot trigger from the snap1 register group. Counter values saved from earlier snapshot triggers are lost.

The time interval between two snapshots can be obtained by subtracting the timer counter values (TMRSNAP0-TMRSNAP1). For the first snapshot trigger, the time interval since the last update of the phase counter PHCNT can be read from the speed counter snapshot SPDSNAP1. For the second snapshot trigger, the time interval since the last update of the phase counter PHCNT can be read from the speed counter snapshot SPDSNAP0.

The distance the motor traveled between two snapshots can be obtained by comparing the phase counter snapshots PHSNAP0, PHSNAP1, and the Z phase counter snapshots ZPHSNAP0, ZPHSNAP1.

From this, the time interval, speed, and other information of motor operation can be calculated.

40.4.8 Counter Reset

The quadrature decoder supports resetting all counters via the following method. After the user software sets the CR[RSTCNT] bit to 1 to reset, the following counters are reset.

- PHCNT, reset to PHIDX
- ZPHCNT, reset to 0
- SPDCNT, reset to 0
- TMRCNT, reset to 0

40.4.9 Counter Pause

The PAUSE input signal comes from the interconnect manager. For details, see the interconnect manager chapter. When the CR register's PAUSESPD is enabled, the SPDCNT count pauses while the input signal is valid. When the CR register's PAUSEPH is enabled, the PHCNT count pauses while the input signal is valid. When the CR register's PAUSEZ is enabled, the ZCNT count pauses while the input signal is valid.

40.4.10 PAD Filter Function of the QEI Module

Same as the digital filter function of the interconnect manager. For details, see the description in the digital filter subsection of the interconnect manager module.

40.4.11 Interrupts, DMA, and Trigger Output

The interrupts supported by the QEI module include watchdog timeout, missing Z phase, Z phase flag generation, HOME phase flag generation, direction change flag generation, etc. Users can enable the required interrupt events in the Interrupt Request Enable Register (IRQEN) according to their needs. For details, see the Interrupt Request Enable Register.

The DMA requests supported by the QEI module include watchdog timeout, missing Z phase, Z phase flag generation, HOME phase flag generation, direction change flag generation, etc. Users can enable the required DMA requests in the DMA Request Enable Register (DMAEN) according to their needs. For details, see the DMA Request Enable Register.

The QEI module supports emitting a trigger output signal TRGO, one pulse in length, in the following sit-

uations: watchdog timeout, missing Z phase, Z phase flag generation, HOME phase flag generation, direction change flag generation, etc. Users can enable the required interrupt events in the Trigger Output Enable Register (TRGOEN) according to their needs. For details, see the Trigger Output Enable Register.

40.5 QEI Register Configuration Guide

I. ABZ Mode, PD Mode, or UD Mode

1. Configure PHZCNT in the Control Register (CR), configure ZCNTCFG according to the Z phase situation (**Special note: If the Z phase signal is not used, do not configure the Z phase pin**), and select the required decoding mode in the [ENCTYP] bits;
2. As needed, configure the motor's maximum line count in the Phase Configuration Register (PHCFG);
3. As needed, configure the reset value for PHZCNT in the Phase Index Register (PHIDX);
4. As needed, configure the phase calculation parameters in the register at offset address 0x230 (PHASE_PARAM);
5. As needed, configure the QEI Configuration Register (QEI_CFG): configure the [NEGEDGE_EN], [POSIDGE_EN] bit fields to enable signal rising and falling edges; set the [SIGA_EN], [SIGB_EN], and [SIGZ_EN] bit fields to 1 to enable the ABZ three-phase signals;
6. As needed, configure the position output threshold in the Position Change Threshold Register (POS_THRESHOLD);
7. As needed, configure the Interrupt Request Enable (IRQEN) or Trigger Output Enable Register (TRGOEN) to enable the required interrupt function or the trig_out event;

II. UVW Mode

1. As needed, configure the [ZCNTCFG] bit field in the Control Register (CR) for the Z phase counting method, and select the UVW decoding mode in the [ENCTYP] bits (i.e., configure the CR[ENCTYP] bit field to value 3);
2. Configure the value of the Phase Index Register (PHIDX) to 5;
3. As needed, configure the QEI Configuration Register (QEI_CFG): configure the [NEGEDGE_EN], [POSIDGE_EN] bit fields to enable signal rising and falling edges; set the [SIGA_EN], [SIGB_EN], and [SIGZ_EN] bit fields to 1 to enable the ABZ three-phase signals;
4. According to the setting of the [UVW_POS_OPT0] bit field in the QEI Configuration Register (QEI_CFG), and the six-phase levels of the UVW three-phase signal, configure the values of the UVW Position 0 Configuration Register (UVW_POS_CFG[UVW_POS0_CFG]), UVW Position 1 Configuration Register (UVW_POS_CFG[UVW_POS1_CFG]), UVW Position 2 Configuration Register (UVW_POS_CFG[UVW_POS2_CFG]), UVW Position 3 Configuration Register (UVW_POS_CFG[UVW_POS3_CFG]), UVW Position 4 Configuration Register (UVW_POS_CFG[UVW_POS4_CFG]), and UVW Position 5 Configuration Register (UVW_POS_CFG[UVW_POS5_CFG]);
5. As needed, configure the output positions for the six UVW phases: UVW Position 0 Angle Register (UVW_POS[UVW_POS0]), UVW Position 1 Angle Register (UVW_POS[UVW_POS1]), UVW Position 2 Angle Register (UVW_POS[UVW_POS2]), UVW Position 3 Angle Register (UVW_POS[UVW_POS3]), UVW Position 4 Angle Register (UVW_POS[UVW_POS4]), UVW Position 5 Angle Register (UVW_POS[UVW_POS5]);
6. As needed, configure the Interrupt Request Enable (IRQEN) or Trigger Output Enable Register (TRGOEN) to enable the required interrupt function or the trig_out event;

III. Quadrature Sine-Cosine Wave Mode



1. Set the decoding mode in the [ENCTYP] bit field of the Control Register (CR) to quadrature sine-cosine wave mode (i.e., configure CR[ENCTYP] bit field to value 6);
2. As needed, configure to enable revolution counting method, and configure the value of the [ZCNTCFG] bit field in the Control Register (CR) according to the Z phase situation;
3. According to the source of the sine-cosine waves, configure the ADC source for the X channel, the value of the X-channel ADC Configuration Register (ADCX_CFG0); simultaneously, configure the ADC source for the Y channel, the value of the Y-channel ADC Configuration Register (ADCY_CFG0);
4. According to the motor's line count, configure the line count (PHCNT_MAX), i.e., set the value of the Phase Configuration Register (PHCFG); simultaneously, configure the Phase Calculation Parameter Register (PHASE_PARAM) based on the motor line count;
5. As needed, configure the Interrupt Request Enable (IRQEN) or Trigger Output Enable Register (TRGOEN) to enable the required interrupt function or the trig_out event;

IV. Single-Phase Sine Wave Mode

1. Set the decoding mode in the [ENCTYP] bit field of the Control Register (CR) to single-phase sine wave mode (i.e., configure CR[ENCTYP] bit field to value 5);
2. Configure the value of the [ZCNTCFG] bit field in the Control Register (CR) according to the Z phase situation;
3. As needed, configure the ADC source for the X channel in the X-channel ADC Configuration Register 0 (ADCX_CFG0);
4. As needed, configure the X parameters in the X-channel ADC Configuration Register 1 (ADCX_CFG1);
5. As needed, configure the offset value for the ADC value in the X-channel ADC Configuration Register 2 (ADCX_CFG2);
6. Set the Phase Configuration Register (PHCFG) according to the motor line count, and simultaneously configure the value of the Phase Calculation Parameter Register (PHASE_PARAM);
7. As needed, configure the Interrupt Request Enable (IRQEN) or Trigger Output Enable Register (TRGOEN) to enable the required interrupt function or the trig_out event;

40.6 QEIV2 Register List

The register list for QEIV2 is as follows:

QEIO base address: 0xF0400000

QEI1 base address: 0xF0404000

Offset	Register Name	Description	Reset Value
0x0000	CR	Control register	0x00000000
0x0004	PHCFG	Phase configure register	0xFFFFFFFF
0x0008	WDGCFG	Watchdog configure register	0x00000000
0x000C	PHIDX	Phase index register	0x00000000
0x0010	TRGOEN	Tigger output enable register	0x00000000

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3
QEIV2

40 QUADRATURE ENCODER INTERFACE

Offset	Register Name	Description	Reset Value
0x0014	READEN	Read event enable register	0x00000000
0x0018	ZCMP	Z comparator	0x80000000
0x001C	PHCMP	Phase comparator	0x80000000
0x0020	SPDCMP	Speed comparator	0x80000000
0x0024	DMAEN	DMA request enable register	0x00000000
0x0028	SR	Status register	0x00000000
0x002C	IRQEN	Interrupt request register	0x00000000
0x0030	COUNT[CURRENT][Z]	Z counter	0x00000000
0x0034	COUNT[CURRENT][PH]	Phase counter	0x00000000
0x0038	COUNT[CURRENT][SPD]	Speed counter	0x00000000
0x003C	COUNT[CURRENT][TMR]	Timer counter	0x00000000
0x0040	COUNT[READ][Z]	Z counter	0x00000000
0x0044	COUNT[READ][PH]	Phase counter	0x00000000
0x0048	COUNT[READ][SPD]	Speed counter	0x00000000
0x004C	COUNT[READ][TMR]	Timer counter	0x00000000
0x0050	COUNT[SNAP0][Z]	Z snap register	0x00000000
0x0054	COUNT[SNAP0][PH]	Phase snap register	0x00000000
0x0058	COUNT[SNAP0][SPD]	Speed snap register	0x00000000
0x005C	COUNT[SNAP0][TMR]	Timer snap register	0x00000000
0x0060	COUNT[SNAP1][Z]	Z snap register 1	0x00000000
0x0064	COUNT[SNAP1][PH]	Phase snap register 1	0x00000000
0x0068	COUNT[SNAP1][SPD]	Speed snap register 1	0x00000000
0x006C	COUNT[SNAP1][TMR]	Timer snap register 1	0x00000000
0x0080	ZCMP2	Z comparator	0x80000000
0x0084	PHCMP2	Phase comparator	0x80000000
0x0088	SPDCMP2	Speed comparator	0x80000000
0x008C	MATCH_CFG		0x00000000
0x0090	FILT_CFG[FILT_CFG_A]	A signal filter config	0x00001000
0x0094	FILT_CFG[FILT_CFG_B]	B signal filter config	0x00001000
0x0098	FILT_CFG[FILT_CFG_Z]	Z signal filter config	0x00001000
0x009C	FILT_CFG[FILT_CFG_H]	H signal filter config	0x00001000
0x00A0	FILT_CFG[FILT_CFG_H2]	H2 signal filter config	0x00001000
0x00A4	FILT_CFG[FILT_CFG_F]	F signal filter config	0x00001000
0x0100	QEI_CFG	qei config register	0x00000000
0x0110	PULSE0_NUM	pulse0_num	0x00000000
0x0114	PULSE1_NUM	pulse1_num	0x00000000
0x0118	CYCLE0_CNT	cycle0_cnt	0x00000000
0x011C	CYCLE0PULSE_CNT	cycle0pulse_cnt	0x00000000
0x0120	CYCLE1_CNT	cycle1_cnt	0x00000000
0x0124	CYCLE1PULSE_CNT	cycle1pulse_cnt	0x00000000

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3
QEIV2

40 QUADRATURE ENCODER INTERFACE

Offset	Register Name	Description	Reset Value
0x0128	CYCLE0_SNAP0	cycle0_snap0	0x00000000
0x012C	CYCLE0_SNAP1	cycle0_snap1	0x00000000
0x0130	CYCLE1_SNAP0	cycle1_snap0	0x00000000
0x0134	CYCLE1_SNAP1	cycle1_snap1	0x00000000
0x0140	CYCLE0_NUM	cycle0_num	0x00000000
0x0144	CYCLE1_NUM	cycle1_num	0x00000000
0x0148	PULSE0_CNT	pulse0_cnt	0x00000000
0x014C	PULSE0CYCLE_CNT	pulse0cycle_cnt	0x00000000
0x0150	PULSE1_CNT	pulse1_cnt	0x00000000
0x0154	PULSE1CYCLE_CNT	pulse1cycle_cnt	0x00000000
0x0158	PULSE0_SNAP0	pulse0_snap0	0x00000000
0x015C	PULSE0CYCLE_SNAP0	pulse0cycle_snap0	0x00000000
0x0160	PULSE0_SNAP1	pulse0_snap1	0x00000000
0x0164	PULSE0CYCLE_SNAP1	pulse0cycle_snap1	0x00000000
0x0168	PULSE1_SNAP0	pulse1_snap0	0x00000000
0x016C	PULSE1CYCLE_SNAP0	pulse1cycle_snap0	0x00000000
0x0170	PULSE1_SNAP1	pulse1_snap1	0x00000000
0x0174	PULSE1CYCLE_SNAP1	pulse1cycle_snap1	0x00000000
0x01E0	TIMESTAMP	timestamp	0x00000000
0x01F0	ADC_THRESHOLD	adc_threshold	0x0000FFFF
0x0200	ADCX_CFG0	adcx_cfg0	0x00000000
0x0204	ADCX_CFG1	adcx_cfg1	0x00004000
0x0208	ADCX_CFG2	adcx_cfg2	0x80000000
0x020C	ADCX_VAL_SW	adcx_val_sw	0x00000000
0x0210	ADCY_CFG0	adcy_cfg0	0x00000000
0x0214	ADCY_CFG1	adcy_cfg1	0x40000000
0x0218	ADCY_CFG2	adcy_cfg2	0x80000000
0x021C	ADCY_VAL_SW	adcy_val_sw	0x00000000
0x0220	CAL_CFG	cal_cfg	0x00000100
0x0224	ADC_INJECT_CTRL	adc_inject_ctrl	0x00000000
0x0230	PHASE_PARAM	phase_param	0xFFFFFFFF
0x0234	POS_ADJ	pos_adj	0x00000000
0x0238	POS_THRESHOLD	pos_threshold	0x80000000
0x0240	UVW_POS[UVW_POS0]	uvw_pos0	0x00000000
0x0244	UVW_POS[UVW_POS1]	uvw_pos1	0x00000000
0x0248	UVW_POS[UVW_POS2]	uvw_pos2	0x00000000
0x024C	UVW_POS[UVW_POS3]	uvw_pos3	0x00000000
0x0250	UVW_POS[UVW_POS4]	uvw_pos4	0x00000000

Offset	Register Name	Description	Reset Value
0x0254	UVW_POS[UVW_POS5]	uvw_pos5	0x00000000
0x0258	UVW_POS_CFG[UVW_POS0_CFG]	uvw_pos0_cfg	0x00000000
0x025C	UVW_POS_CFG[UVW_POS1_CFG]	uvw_pos1_cfg	0x00000000
0x0260	UVW_POS_CFG[UVW_POS2_CFG]	uvw_pos2_cfg	0x00000000
0x0264	UVW_POS_CFG[UVW_POS3_CFG]	uvw_pos3_cfg	0x00000000
0x0268	UVW_POS_CFG[UVW_POS4_CFG]	uvw_pos4_cfg	0x00000000
0x026C	UVW_POS_CFG[UVW_POS5_CFG]	uvw_pos5_cfg	0x00000000
0x0280	PHASE_CNT	phase_cnt	0x00000000
0x0284	PHASE_UPDATE	phase_update	0x00000000
0x0288	POSITION	position	0x00000000
0x028C	POSITION_UPDATE	position_update	0x00000000
0x0290	ANGLE	angle	0x00000000
0x0294	POS_TIMEOUT	pos_timeout	0x7FFFFFFF
0x02A0	CALC_STATE	calc_state	0x00000000
0x02C0	TOGI_CFG0	togi_cfg0	0x00000000
0x02C4	TOGI_CFG1	togi_cfg1	0x00000000
0x02D0	SINP_ACC	sinp_acc	0x00000000
0x02D4	SINN_ACC	sinn_acc	0x00000000
0x02D8	COSP_ACC	cosp_acc	0x00000000
0x02DC	COSN_ACC	cosn_acc	0x00000000

Table 182: QEIV2 Register List

40.7 QEIV2 Register Details

The detailed description of QEIV2 registers is as follows:

40.7.1 CR (0x0)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
READ									ZCNTCFG	PHCALIZ	Z_ONLY_EN	H2FDIR0	H2FDIR1	H2RDIR0	H2RDIR1	PAUSEPOS	PAUSEPD	PAUSEPH	PAUSEZ	HFDIR0	HFDIR1	HRDIR0	HRDIR1	RSVD	FAULTPOS	SNAPEN	RSTCNT	RD_SEL			ENCTYP
WO									RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	N/A	RW	RW	RW	RW	RW	RW	RW
0	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	0	0	0	0	0	0	0

CR [31:0]

Bits	Name	Description
31	READ	1- load phcnt, zcnt, spdcnt and tmrcnt into their read registers. Hardware auto-clear; read as 0

Bits	Name	Description
22	ZCNTCFG	1- zcnt will increment when phcnt upcount to phmax, decrement when phcnt downcount to 0 0- zcnt will increment or decrement when Z input assert
21	PHCALIZ	1- phcnt will set to phidx when Z input assert(for abz digital signals)
20	Z_ONLY_EN	1- phcnt will set to phidx when Z input assert(only using z phase, ignore ab phase signals)
19	H2FDIR0	
18	H2FDIR1	
17	H2RDIR0	
16	H2RDIR1	
15	PAUSEPOS	1- pause position output valid when PAUSE assert
14	PAUSESPD	1- pause spdcnt when PAUSE assert
13	PAUSEPH	1- pause phcnt when PAUSE assert
12	PAUSEZ	1- pause zcnt when PAUSE assert
11	HFDIR0	1- HOMEF will set at H rising edge when dir == 1 (negative rotation direction)
10	HFDIR1	1- HOMEF will set at H rising edge when dir == 0 (positive rotation direction)
9	HRDIR0	1- HOMEF will set at H falling edge when dir == 1 (negative rotation direction)
8	HRDIR1	1- HOMEF will set at H falling edge when dir == 1 (positive rotation direction)
6	FAULTPOS	
5	SNAPEN	1- load phcnt, zcnt, spdcnt and tmrcnt into their snap registers when snapi input assert
4	RSTCNT	1- reset zcnt, spdcnt and tmrcnt to 0. reset phcnt to phidx
3	RD_SEL	define the width/counter value(affect width_match, width_match2, width_cur, timer_cur, width_read, timer_read, width_snap0,width_snap1, timer_snap0, timer_snap1) 0 : same as hpm1000/500/500s; 1: use width for position; use timer for angle
2-0	ENCTYP	000-abz; 001-pd; 010-ud; 011-UVW(hal) 100-single A; 101-single sin; 110: sin&cos

CR Bits

40.7.2 PHCFG (0x4)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PHMAX																															
RW																															
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	

PHCFG [31:0]

Bits	Name	Description
31-0	PHMAX	maximum phcnt number, phcnt will rollover to 0 when it upcount to phmax

PHCFG Bits

40.7.3 WDGCFG (0x8)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDGEN	WDOG_CFG			WDGTO																											
RW	RW			RW																											
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

WDGCFG [31:0]

Bits	Name	Description
31	WDGEN	1- enable wdog counter
30-28	WDOG_CFG	define as stop if phase_cnt change is less than it if 0, then each change of phase_cnt will clear wdog counter; if 2, then phase_cnt change larger than 2 will clear wdog counter
27-0	WDGTO	watch dog timeout value

WDGCFG Bits

40.7.4 PHIDX (0xC)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PHIDX																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

PHIDX [31:0]

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

40 QUADRATURE ENCODER INTERFACE

QEIV2

Bits	Name	Description
31-0	PHIDX	phcnt reset value, phcnt will reset to phidx when phcaliz set to 1

PHIDX Bits

40.7.5 TRGOEN (0x10)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDGFEN	HOMEFEN	POSCMPFEN	ZPHFEN	ZMISSFEN	WIDHTMFEN	POS2CMPFEN	DIRCHGFEN	CYCLE0FEN	CYCLE1FEN	PULSE0FEN	PULSE1FEN	HOME2FEN	FAULTFEN	RSVD																	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	N/A																	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

TRGOEN [31:0]

Bits	Name	Description
31	WDGFEN	1- enable trigger output when wdg flag set
30	HOMEFEN	1- enable trigger output when homef flag set
29	POSCMPFEN	1- enable trigger output when poscmpf flag set
28	ZPHFEN	1- enable trigger output when zphf flag set
27	ZMISSFEN	
26	WIDTHTMFEN	
25	POS2CMPFEN	
24	DIRCHGFEN	
23	CYCLE0FEN	
22	CYCLE1FEN	
21	PULSE0FEN	
20	PULSE1FEN	
19	HOME2FEN	
18	FAULTFEN	

TRGOEN Bits

40.7.6 READEN (0x14)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDGFEN	HOMEFEN	POSCMPFEN	ZPHFEN	ZMISSFEN	WIDTHTMFEN	POS2CMPFEN	DIRCHGFEN	CYCLE0FEN	CYCLE1FEN	PULSE0FEN	PULSE1FEN	HOME2FEN	FAULTFEN	RSVD																	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	N/A																	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

READEN [31:0]

Bits	Name	Description
31	WDGFEN	1- load counters to their read registers when wdg flag set
30	HOMEFEN	1- load counters to their read registers when homef flag set
29	POSCMPFEN	1- load counters to their read registers when poscmpf flag set
28	ZPHFEN	1- load counters to their read registers when zphf flag set
27	ZMISSFEN	
26	WIDTHTMFEN	
25	POS2CMPFEN	
24	DIRCHGFEN	
23	CYCLE0FEN	
22	CYCLE1FEN	
21	PULSE0FEN	
20	PULSE1FEN	
19	HOME2FEN	
18	FAULTFEN	

READEN Bits

40.7.7 ZCMP (0x18)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ZCMP																															
RW																															
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ZCMP [31:0]

Bits	Name	Description
31-0	ZCMP	zcnt postion compare value

ZCMP Bits

40.7.8 PHCMP (0x1C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PHCMP																															
RW																															
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

PHCMP [31:0]

Bits	Name	Description
31-0	PHCMP	phcnt position compare value

PHCMP Bits

40.7.9 SPDCMP (0x20)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																SPDCMP															
																RW															
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPDCMP [31:0]

Bits	Name	Description
31-0	SPDCMP	spdcnt position compare value

SPDCMP Bits

40.7.10 DMAEN (0x24)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDGFEN	HOMEFEN	POSCMPFEN	ZPHFEN	ZMISSFEN	WIDTHMFEN	POS2CMPFEN	DIRCHGFEN	CYCLE0FEN	CYCLE1FEN	PULSE0FEN	PULSE1FEN	HOME2FEN	FAULTFEN	RSVD																	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	N/A																	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

DMAEN [31:0]

Bits	Name	Description
31	WDGFEN	1- generate dma request when wdg flag set
30	HOMEFEN	1- generate dma request when homef flag set
29	POSCMPFEN	1- generate dma request when poscmpf flag set
28	ZPHFEN	1- generate dma request when zphf flag set
27	ZMISSFEN	
26	WIDTHTMFEN	
25	POS2CMPFEN	
24	DIRCHGFEN	

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

40 QUADRATURE ENCODER INTERFACE

QEIV2

Bits	Name	Description
23	CYCLE0FEN	
22	CYCLE1FEN	
21	PULSE0FEN	
20	PULSE1FEN	
19	HOME2FEN	
18	FAULTFEN	

DMAEN Bits

40.7.11 SR (0x28)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDGF	HOMEF	POSCMPF	ZPHF	ZMISSF	WIDTHTMF	POS2CMPF	DIRCHGF	CYCLE0F	CYCLE1F	PULSE0F	PULSE1F	HOME2F	FAULTF	RSVD																	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	N/A																	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

SR [31:0]

Bits	Name	Description
31	WDGF	watchdog flag
30	HOMEF	home flag
29	POSCMPF	position compare match flag
28	ZPHF	z input flag
27	ZMISSF	
26	WIDTHTMF	
25	POS2CMPF	
24	DIRCHGF	
23	CYCLE0F	
22	CYCLE1F	
21	PULSE0F	
20	PULSE1F	
19	HOME2F	
18	FAULTF	

SR Bits

40.7.12 IRQEN (0x2C)



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDGIE	HOMEIE	POSCMPIE	ZPHIE	ZMISSE	WIDTHTME	POS2CMPE	DIRCHGE	CYCLE0E	CYCLE1E	PULSE0E	PULSE1E	HOME2E	FAULTE	RSVD																	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	N/A																	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	

IRQEN [31:0]

Bits	Name	Description
31	WDGIE	1- generate interrupt when wdg flag set
30	HOMEIE	1- generate interrupt when homef flag set
29	POSCMPIE	1- generate interrupt when poscmpf flag set
28	ZPHIE	1- generate interrupt when zphf flag set
27	ZMISSE	
26	WIDTHTME	
25	POS2CMPE	
24	DIRCHGE	
23	CYCLE0E	
22	CYCLE1E	
21	PULSE0E	
20	PULSE1E	
19	HOME2E	
18	FAULTE	

IRQEN Bits

40.7.13 COUNT[Z] (0x30 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ZCNT																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

COUNT[Z] [31:0]

Bits	Name	Description
31-0	ZCNT	zcnt value

COUNT[Z] Bits

40.7.14 COUNT[PH] (0x34 + 0x10 * n)

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

40 QUADRATURE ENCODER INTERFACE

QEIV2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	DIR		RSVD		ASTAT	BSTAT		RSVD																							
N/A	RO		N/A		RO	RO		N/A																							
x	0	x	x	x	0	0	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

COUNT[PH] [31:0]

Bits	Name	Description
30	DIR	1- reverse rotation 0- forward rotation
26	ASTAT	1- a input is high 0- a input is low
25	BSTAT	1- b input is high 0- b input is low
20-0	PHCNT	phcnt value

COUNT[PH] Bits

40.7.15 COUNT[SPD] (0x38 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIR	ASTAT	BSTAT	RSVD																												
RO	RO	RW	N/A																												
0	0	0	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

COUNT[SPD] [31:0]

Bits	Name	Description
31	DIR	1- reverse rotation 0- forward rotation
30	ASTAT	1- a input is high 0- a input is low
29	BSTAT	1- b input is high 0- b input is low
27-0	SPDCNT	spdcnt value

COUNT[SPD] Bits

40.7.16 COUNT[TMR] (0x3C + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRCNT																															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

COUNT[TMR] [31:0]

Bits	Name	Description
31-0	TMRCNT	32 bit free run timer

COUNT[TMR] Bits

40.7.17 ZCMP2 (0x80)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ZCMP2																															
RW																															
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ZCMP2 [31:0]

Bits	Name	Description
31-0	ZCMP2	

ZCMP2 Bits

40.7.18 PHCMP2 (0x84)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PHCMP2																															
RW																															
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PHCMP2 [31:0]

Bits	Name	Description
31-0	PHCMP2	

PHCMP2 Bits

40.7.19 SPDCMP2 (0x88)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPDCMP2																RW															
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPDCMP2 [31:0]

Bits	Name	Description
31-0	SPDCMP2	

SPDCMP2 Bits

40.7.20 MATCH_CFG (0x8C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ZCMPDIS	DIRCMPDIS	DIRCMP	SPDCMPDIS	PHASE_MATCH_DIS	POS_MATCH_DIR	POS_MATCH_OPT	RSVD									ZCMP2DIS	DIRCMP2DIS	DIRCMP2	SPDCMP2DIS	PHASE_MATCH_DIS2	POS_MATCH2_DIR	POS_MATCH2_OPT	RSVD								
RW	RW	RW	RW	RW	RW	RW	N/A									RW	RW	RW	RW	RW	RW	RW	N/A								
0	0	0	0	0	0	0	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x	x

MATCH_CFG [31:0]

Bits	Name	Description
31	ZCMPDIS	1- position compare not include zcnt
30	DIRCMPDIS	1- position compare not include rotation direction
29	DIRCMP	0- position compare need positive rotation 1- position compare need negative rotation
28	SPDCMPDIS	
27	PHASE_MATCH_DIS	
26	POS_MATCH_DIR	
25	POS_MATCH_OPT	
15	ZCMP2DIS	
14	DIRCMP2DIS	
13	DIRCMP2	
12	SPDCMP2DIS	
11	PHASE_MATCH_DIS2	

Bits	Name	Description
10	POS_MATCH2_DIR	
9	POS_MATCH2_OFFSET	

MATCH_CFG Bits

40.7.21 FILT_CFG (0x90 + 0x4 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD															OUTINV	MODE			SYNCEN	FILTLEN											
N/A															RW	RW			RW	RW											
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

FILT_CFG [31:0]

Bits	Name	Description
16	OUTINV	1- Filter will invert the output 0- Filter will not invert the output
15-13	MODE	This bitfields defines the filter mode 000-bypass; 100-rapid change mode; 101-delay filter mode; 110-stable low mode; 111-stable high mode
12	SYNCEN	set to enable sychronization input signal with TRGM clock
11-0	FILTLEN	This bitfields defines the filter counter length.

FILT_CFG Bits

40.7.22 QEI_CFG (0x100)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
SW_PULSE0_RESTART	SW_PULSE1_RESTART	SW_CYCLE0_RESTART	SW_CYCLE1_RESTART	RSVD								PULSE0_ONESHOT	PULSE1_ONESHOT	CYCLE0_ONESHOT	CYCLE1_ONESHOT	RSVD			SPEED_DIR_CHG_EN	TRIG_PULSE0_EN	TRIG_PULSE1_EN	TRIG_CYCLE0_EN	TRIG_CYCLE1_EN	RSVD			UVW_POS_OPT0	NEGEDGE_EN	POSIDGE_EN	SIGZ_EN	SIGB_EN	SIGA_EN	
RW	RW	RW	RW	N/A								RW	RW	RW	RW	N/A			RW	RW	RW	RW	RW	N/A			RW	RW	RW	RW	RW	RW	RW
0	0	0	0	x	x	x	x	x	x	x	x	0	0	0	0	x	x	x	0	0	0	0	0	x	x	0	0	0	0	0	0		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

QEI_CFG [31:0]

Bits	Name	Description
31	SW_PULSE0_RE START	set to restart cycle counter for configed pulse_num. HW auto clear
30	SW_PULSE1_RE START	
29	SW_CYCLE0_RE START	set to restart pulse counter for configed cycle_num. HW auto clear
28	SW_CYCLE1_RE START	
19	PULSE0_ONESH OT	set to use oneshot mode for configed pulse_num
18	PULSE1_ONESH OT	
17	CYCLE0_ONESH OT	set to use oneshot mode for configed cycle_num
16	CYCLE1_ONESH OT	
12	SPEED_DIR_CHG _EN	clear counter if detect direction change
11	TRIG_PULSE0_E N	set to enable trigger start cycle counter for configed pulse_num(from the selected edge)
10	TRIG_PULSE1_E N	
9	TRIG_CYCLE0_E N	set to enable trigger start pulse counter for configed cycle_num
8	TRIG_CYCLE1_E N	
5	UVW_POS_OPT0	set to output next area position for QEO use; clr to output exact point position for MMC use
4	NEGEDGE_EN	bit4: negedge enable bit3: posedge enable bit2: W in hal enable bit1: signal b(or V in hal) enable bit0: signal a(or U in hal) enable such as: 01001: use posedge A 11010: use both edge of signal B 11111: use both edge of all HAL siganls
3	POSIDGE_EN	

Bits	Name	Description
2	SIGZ_EN	
1	SIGB_EN	
0	SIGA_EN	

QEI_CFG Bits

40.7.23 PULSE0_NUM (0x110)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PULSE0_NUM																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PULSE0_NUM [31:0]

Bits	Name	Description
31-0	PULSE0_NUM	for speed detection, will count the cycle number for configed pulse_num

PULSE0_NUM Bits

40.7.24 PULSE1_NUM (0x114)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PULSE1_NUM																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PULSE1_NUM [31:0]

Bits	Name	Description
31-0	PULSE1_NUM	

PULSE1_NUM Bits

40.7.25 CYCLE0_CNT (0x118)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CYCLE0_CNT																															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CYCLE0_CNT [31:0]

Bits	Name	Description
31-0	CYCLE0_CNT	

CYCLE0_CNT Bits

40.7.26 CYCLE0PULSE_CNT (0x11C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CYCLE0PULSE_CNT																															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CYCLE0PULSE_CNT [31:0]

Bits	Name	Description
31-0	CYCLE0PULSE_CNT	

CYCLE0PULSE_CNT Bits

40.7.27 CYCLE1_CNT (0x120)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CYCLE1_CNT																															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CYCLE1_CNT [31:0]

Bits	Name	Description
31-0	CYCLE1_CNT	

CYCLE1_CNT Bits

40.7.28 CYCLE1PULSE_CNT (0x124)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																CYCLE1PULSE_CNT															
																RO															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CYCLE1PULSE_CNT [31:0]

Bits	Name	Description
31-0	CYCLE1PULSE_CNT	

CYCLE1PULSE_CNT Bits

40.7.29 CYCLE0_SNAP0 (0x128)

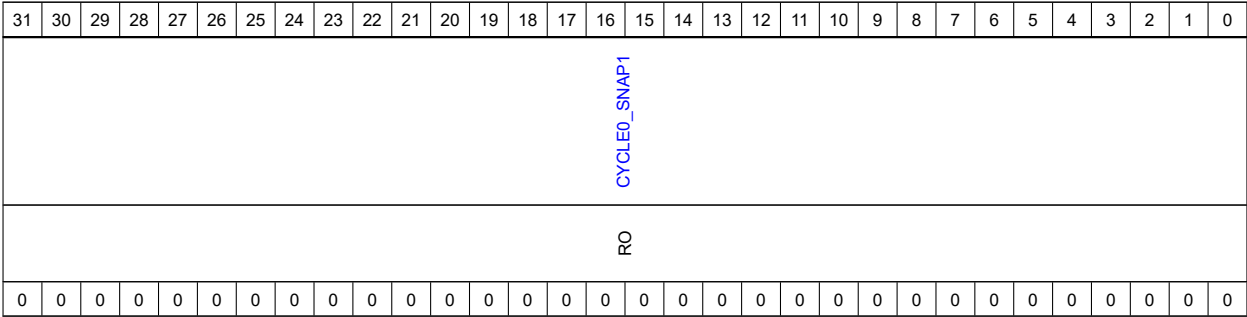
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																CYCLE0_SNAP0															
																RO															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CYCLE0_SNAP0 [31:0]

Bits	Name	Description
31-0	CYCLE0_SNAP0	

CYCLE0_SNAP0 Bits

40.7.30 CYCLE0_SNAP1 (0x12C)

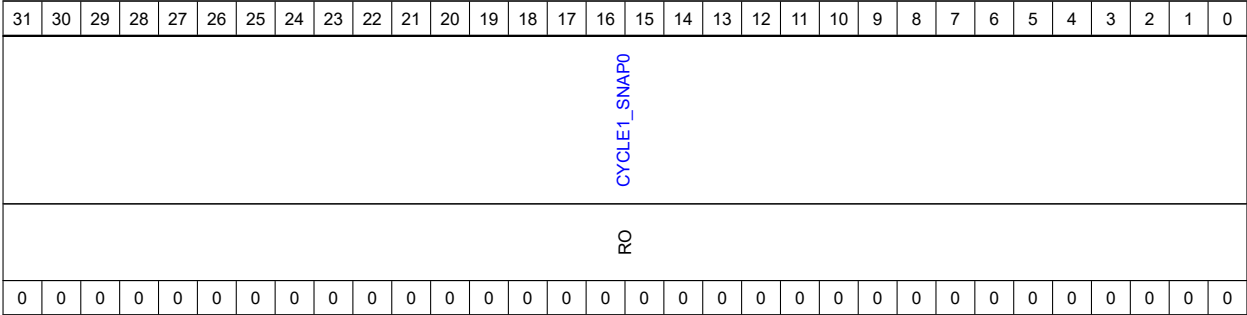


CYCLE0_SNAP1 [31:0]

Bits	Name	Description
31-0	CYCLE0_SNAP1	

CYCLE0_SNAP1 Bits

40.7.31 CYCLE1_SNAP0 (0x130)

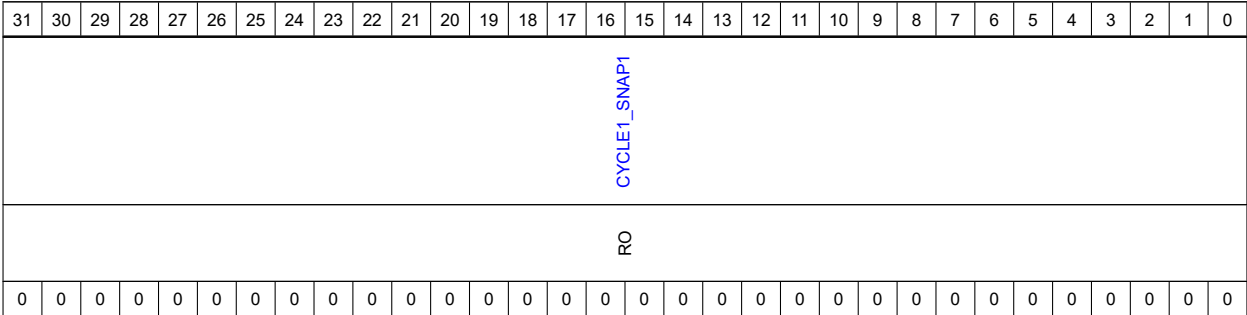


CYCLE1_SNAP0 [31:0]

Bits	Name	Description
31-0	CYCLE1_SNAP0	

CYCLE1_SNAP0 Bits

40.7.32 CYCLE1_SNAP1 (0x134)



CYCLE1_SNAP1 [31:0]

Bits	Name	Description
31-0	CYCLE1_SNAP1	

CYCLE1_SNAP1 Bits

40.7.33 CYCLE0_NUM (0x140)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CYCLE0_NUM																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CYCLE0_NUM [31:0]

Bits	Name	Description
31-0	CYCLE0_NUM	

CYCLE0_NUM Bits

40.7.34 CYCLE1_NUM (0x144)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CYCLE1_NUM																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CYCLE1_NUM [31:0]

Bits	Name	Description
31-0	CYCLE1_NUM	

CYCLE1_NUM Bits

40.7.35 PULSE0_CNT (0x148)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PULSE0_CNT																															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PULSE0_CNT [31:0]

Bits	Name	Description
31-0	PULSE0_CNT	

PULSE0_CNT Bits

40.7.36 PULSE0CYCLE_CNT (0x14C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PULSE0CYCLE_CNT																															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PULSE0CYCLE_CNT [31:0]

Bits	Name	Description
31-0	PULSE0CYCLE_CNT	

PULSE0CYCLE_CNT Bits

40.7.37 PULSE1_CNT (0x150)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PULSE1_CNT																															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PULSE1_CNT [31:0]

Bits	Name	Description
31-0	PULSE1_CNT	

PULSE1_CNT Bits

40.7.38 PULSE1CYCLE_CNT (0x154)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PULSE1CYCLE_CNT																															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PULSE1CYCLE_CNT [31:0]

Bits	Name	Description
31-0	PULSE1CYCLE_CNT	

PULSE1CYCLE_CNT Bits

40.7.39 PULSE0_SNAP0 (0x158)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PULSE0_SNAP0																															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PULSE0_SNAP0 [31:0]

Bits	Name	Description
31-0	PULSE0_SNAP0	

PULSE0_SNAP0 Bits

40.7.40 PULSE0CYCLE_SNAP0 (0x15C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PULSE0CYCLE_SNAP0																															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PULSE0CYCLE_SNAP0 [31:0]

Bits	Name	Description
31-0	PULSE0CYCLE_SNAP0	

PULSE0CYCLE_SNAP0 Bits

40.7.41 PULSE0_SNAP1 (0x160)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PULSE0_SNAP1																															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PULSE0_SNAP1 [31:0]

Bits	Name	Description
31-0	PULSE0_SNAP1	

PULSE0_SNAP1 Bits

40.7.42 PULSE0CYCLE_SNAP1 (0x164)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PULSE0CYCLE_SNAP1																															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PULSE0CYCLE_SNAP1 [31:0]

Bits	Name	Description
31-0	PULSE0CYCLE_S NAP1	

PULSE0CYCLE_SNAP1 Bits

40.7.43 PULSE1_SNAP0 (0x168)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PULSE1_SNAP0																															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

PULSE1_SNAP0 [31:0]

Bits	Name	Description
31-0	PULSE1_SNAP0	

PULSE1_SNAP0 Bits

40.7.44 PULSE1CYCLE_SNAP0 (0x16C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PULSE1CYCLE_SNAP0																															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

PULSE1CYCLE_SNAP0 [31:0]

Bits	Name	Description
31-0	PULSE1CYCLE_S NAP0	

PULSE1CYCLE_SNAP0 Bits

40.7.45 PULSE1_SNAP1 (0x170)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																PULSE1_SNAP1															
																RO															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PULSE1_SNAP1 [31:0]

Bits	Name	Description
31-0	PULSE1_SNAP1	

PULSE1_SNAP1 Bits

40.7.46 PULSE1CYCLE_SNAP1 (0x174)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																PULSE1CYCLE_SNAP1															
																RO															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PULSE1CYCLE_SNAP1 [31:0]

Bits	Name	Description
31-0	PULSE1CYCLE_SNAP1	

PULSE1CYCLE_SNAP1 Bits

40.7.47 TIMESTAMP (0x1E0)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																TIMESTAMP															
																RO															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TIMESTAMP [31:0]

Bits	Name	Description
31-0	TIMESTAMP	for SIN/COS mode, it saves the timestamp of the beginning of first ADC sample time; for ABZ mode, it saves the timestamp of edge of input signals

TIMESTAMP Bits

40.7.48 ADC_THRESHOLD (0x1F0)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LOW_LIMIT																HIGH_LIMIT															
RW																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

ADC_THRESHOLD [31:0]

Bits	Name	Description
31-16	LOW_LIMIT	for SINCOS mode, if (max+min/2) of the two adc result, is small than limit, then this value is treated as invalid, no position output. this is used to avoid wrong adc result(such as 0 or same sin cos value)
15-0	HIGH_LIMIT	high limit for SINCOS mode adc result

ADC_THRESHOLD Bits

40.7.49 ADCX_CFG0 (0x200)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																							X_ADCSEL	X_ADC_ENABLE	RSVD	X_CHAN					
N/A																							RW	RW	N/A	RW					
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	x	x	0	0	0	0	0

ADCX_CFG0 [31:0]

Bits	Name	Description
8	X_ADCSEL	
7	X_ADC_ENABLE	
4-0	X_CHAN	

Bits	Name	Description

ADCX_CFG0 Bits

40.7.50 ADCX_CFG1 (0x204)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X_PARAM1																X_PARAM0															
RW																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ADCX_CFG1 [31:0]

Bits	Name	Description
31-16	X_PARAM1	
15-0	X_PARAM0	

ADCX_CFG1 Bits

40.7.51 ADCX_CFG2 (0x208)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X_OFFSET																															
RW																															
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ADCX_CFG2 [31:0]

Bits	Name	Description
31-0	X_OFFSET	

ADCX_CFG2 Bits

40.7.52 ADCX_VAL_SW (0x20C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCX_VAL_SW																															
RW																															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ADCX_VAL_SW [31:0]

Bits	Name	Description
31-0	ADCX_VAL_SW	adc X channel value when using adc inject feature. 32bit signed value NOTE: x_offset is not used when adc_inject_en is set, x_param0 and x_param1 are still used, user can leave them as default if do not want to use them

ADCX_VAL_SW Bits

40.7.53 ADCY_CFG0 (0x210)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																							Y_ADCSEL	Y_ADC_ENABLE	RSVD		Y_CHAN				
N/A																							RW	RW	N/A		RW				
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	x	x	0	0	0	0	0

ADCY_CFG0 [31:0]

Bits	Name	Description
8	Y_ADCSEL	
7	Y_ADC_ENABLE	
4-0	Y_CHAN	

ADCY_CFG0 Bits

40.7.54 ADCY_CFG1 (0x214)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Y_PARAM1																Y_PARAM0															
RW																RW															
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ADCY_CFG1 [31:0]

Bits	Name	Description
31-16	Y_PARAM1	
15-0	Y_PARAM0	

ADCY_CFG1 Bits

40.7.55 ADCY_CFG2 (0x218)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Y_OFFSET																RW															
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ADCY_CFG2 [31:0]

Bits	Name	Description
31-0	Y_OFFSET	

ADCY_CFG2 Bits

40.7.56 ADCY_VAL_SW (0x21C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCY_VAL_SW																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ADCY_VAL_SW [31:0]

Bits	Name	Description
31-0	ADCY_VAL_SW	

ADCY_VAL_SW Bits

40.7.57 CAL_CFG (0x220)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								XY_DELAY																							

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
N/A								RW																							
x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	

CAL_CFG [31:0]

Bits	Name	Description
23-0	XY_DELAY	valid x/y delay, larger than this delay will be treated as invalid data. Default 1.25us@200MHz; max 80ms;

CAL_CFG Bits

40.7.58 ADC_INJECT_CTRL (0x224)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADC_INJECT_EN	RSD																													ADCY_INJ_VALID	ADCX_INJ_VALID
RW	N/A																													WO	WO
0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	

ADC_INJECT_CTRL [31:0]

Bits	Name	Description
31	ADC_INJECT_EN	set to use adcx_val and adcy_val from software
1	ADCY_INJ_VALID	set for adc X channel valid
0	ADCX_INJ_VALID	set for adc Y channel valid

ADC_INJECT_CTRL Bits

40.7.59 PHASE_PARAM (0x230)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
PHASE_PARAM																																	
RW																																	
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			

PHASE_PARAM [31:0]

Bits	Name	Description
31-0	PHASE_PARAM	

PHASE_PARAM Bits

40.7.60 POS_ADJ (0x234)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																POS_ADJ															
																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

POS_ADJ [31:0]

Bits	Name	Description
31-0	POS_ADJ	

POS_ADJ Bits

40.7.61 POS_THRESHOLD (0x238)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																POS_THRESHOLD															
																RW															
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

POS_THRESHOLD [31:0]

Bits	Name	Description
31-0	POS_THRESHOLD	

POS_THRESHOLD Bits

40.7.62 UVW_POS (0x240 + 0x4 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																UVW_POS0															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

UVW_POS [31:0]

Bits	Name	Description
31-0	UVW_POS0	

UVW_POS Bits

40.7.63 UVW_POS_CFG (0x258 + 0x4 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								POS_EN	U_POS_SEL		V_POS_SEL		W_POS_SEL		
N/A																								RW	RW		RW		RW		
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	

UVW_POS_CFG [31:0]

Bits	Name	Description
6	POS_EN	
5-4	U_POS_SEL	
3-2	V_POS_SEL	
1-0	W_POS_SEL	

UVW_POS_CFG Bits

40.7.64 PHASE_CNT (0x280)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PHASE_CNT																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PHASE_CNT [31:0]

Bits	Name	Description
31-0	PHASE_CNT	

PHASE_CNT Bits

40.7.65 PHASE_UPDATE (0x284)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INC	DEC	VALUE																													
WO	WO	WO																													
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

PHASE_UPDATE [31:0]

Bits	Name	Description
31	INC	set to add value to phase_cnt
30	DEC	set to minus value from phase_cnt(set inc and dec same time willl act inc)
29-0	VALUE	value to be added or minus from phase_cnt. only valid when inc or dec is set in one 32bit write operation

PHASE_UPDATE Bits

40.7.66 POSITION (0x288)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POSITION																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

POSITION [31:0]

Bits	Name	Description
31-0	POSITION	

POSITION Bits

40.7.67 POSITION_UPDATE (0x28C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INC	DEC	VALUE																													
WO	WO	WO																													
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

POSITION_UPDATE [31:0]

Bits	Name	Description
31	INC	set to add value to position
30	DEC	set to minus value from position(set inc and dec same time will act inc)
29-0	VALUE	value to be added or minus from position. only valid when inc or dec is set in one 32bit write operation

POSITION_UPDATE Bits

40.7.68 ANGLE (0x290)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ANGLE																															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ANGLE [31:0]

Bits	Name	Description
31-0	ANGLE	

ANGLE Bits

40.7.69 POS_TIMEOUT (0x294)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE	TIMEOUT																														
RW	RW																														
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	

POS_TIMEOUT [31:0]

Bits	Name	Description
31	ENABLE	enable position timeout feature, if timeout, send valid again
30-0	TIMEOUT	postion timeout value

POS_TIMEOUT Bits

40.7.70 CALC_STATE (0x2A0)

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

40 QUADRATURE ENCODER INTERFACE

QEIV2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD				STATE				RSVD																							
N/A				RO				N/A																							
x	x	x	x	0	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	

CALC_STATE [31:0]

Bits	Name	Description
27-24	STATE	0: calculate finish other: in calculating

CALC_STATE Bits

40.7.71 TOGI_CFG0 (0x2C0)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIN_TOGI	RSVD																														
RW	N/A																														
0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	

TOGI_CFG0 [31:0]

Bits	Name	Description
31	SIN_TOGI	set to use TOGI architecture for single SIN input

TOGI_CFG0 Bits

40.7.72 TOGI_CFG1 (0x2C4)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD				W_PARAM														RW													
N/A																															
x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TOGI_CFG1 [31:0]

Bits	Name	Description
27-0	W_PARAM	w_param is angle speed, which is $2 \times \text{PI} \times \text{signal\_hz} / \text{adc\_sample\_rate} \times 2^{28}$. for example, 50Hz signal with 1MHz ADC sample rate, angle speed is $2 \times 3.14159 \times 50 / 1000000 \times 0x10000000$, which is 84331(0x1496B, default value) if 50Hz with 2MHz sample rate, should config to 0xA4B6

TOGI_CFG1 Bits

40.7.73 SINP_ACC (0x2D0)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																SINP_ACC															
																RO															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SINP_ACC [31:0]

Bits	Name	Description
31-0	SINP_ACC	positive SIN value acc in one period, updated when calculated SIN from pos to neg. each SIN value is 16bit unsinged value(0x8000 for SIN30(0.5))

SINP_ACC Bits

40.7.74 SINN_ACC (0x2D4)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																SINN_ACC															
																RO															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SINN_ACC [31:0]

Bits	Name	Description
31-0	SINN_ACC	negative SIN value acc in one period, updated when calculated SIN from neg to pos each SIN value is 16bit unsinged value(0x8000 for SIN330(-0.5))

Bits	Name	Description

SINN_ACC Bits

40.7.75 COSP_ACC (0x2D8)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																COSP_ACC															
																RO															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

COSP_ACC [31:0]

Bits	Name	Description
31-0	COSP_ACC	positive COS value acc in one period, updated when calculated COS from pos to neg

COSP_ACC Bits

40.7.76 COSN_ACC (0x2DC)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																COSN_ACC															
																RO															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

COSN_ACC [31:0]

Bits	Name	Description
31-0	COSN_ACC	negative COS value acc in one period, updated when calculated COS from neg to pos

COSN_ACC Bits

41 Quadrature Encoder Output QEO

This chapter introduces the main functions and features of the Quadrature Encoder Output QEO.

41.1 Feature Summary

This chapter introduces the main features of the Quadrature Encoder Output QEO:

- Acquires position from the system position manager
- Supports position compensation
- Supports line number conversion
- Supports single-phase, two-phase (quadrature, up/down, direction pulse), and three-phase digital pulse output
- Supports INDEX/Z signal output
- Configurable maximum and minimum pulse widths
- Supports analog signal output from DAC
- Supports PWM output for three-phase brushless DC motor and two-phase stepper motor control
- Supports direct output for three-phase brushless DC and two-phase stepper motors
- Supports sine wave, half-wave, saddle wave, sawtooth wave
- Supports signal amplification/offset adjustment
- Specified duty cycle for out-of-range signals

41.2 Block Diagram

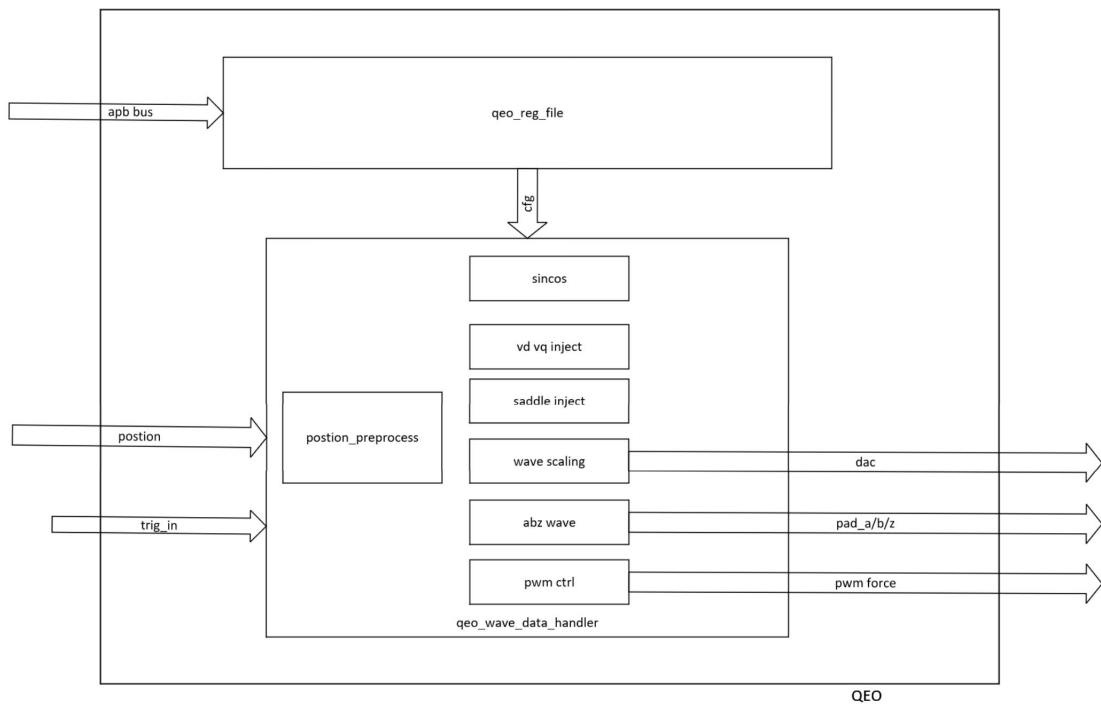


Figure 51: QEO Block Diagram

41.3 Functional Description

41.3.1 Position Preprocessing

QEO has two types of position sources:

- 1) System position manager;
- 2) Software-injected position via writing to the POSTION_SOFTWARE register.

The effective number of bits for the input position to QEO is 32 bits. The transition of this number from 0x00000000 to 0xFFFFFFFF corresponds to a position transition of 1 line, an angle of 2π radians, or 360° .

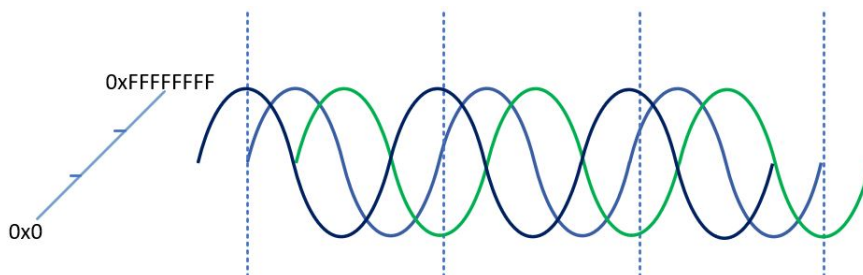
Each valid position input to QEO undergoes preprocessing, which consists of two steps:

- 1) Line number conversion;
- 2) Position compensation.

Line number conversion means that the upstream position input transitioning from 0x00000000 to 0xFFFFFFFF originally corresponds to 1 line. After passing through the line number conversion module, the upstream position input transitioning from 0x00000000 to 0xFFFFFFFF corresponds to n lines, where n is the value of the WAVE/ABZ/PWM[RESOLUTION] register.

Position compensation refers to applying a phase offset to the position after line number conversion.

The sine wave, ABZ, and PWM force outputs each have independent configuration registers for line number conversion and position compensation.



The figure above uses sine wave output as an example, with line conversion set to 3, and the three channels
Figure 52: Example of Sine Wave Output with Line Conversion Set to 3 and 120° Phase Shift

configured with a 120° phase difference. When the input transitions from 0x0 to 0xFFFFFFFF, the output sine wave will output three periods.

41.3.2 Sine Wave Output

Sine wave output supports 4 modes:

- 1) Cosine wave;
- 2) Half-wave (rectified sine);
- 3) Saddle wave;
- 4) Sawtooth wave.

Each output mode is equipped with 3 output channels. The phase offset for each channel can be independently configured via WAVE[PHASH_SHIFT][WAVE n].

41.3.2.1 VD/VQ Control for Sine Waves

The 3 output channels for sine waves can each be controlled by software configuring the WAVE[VD_VQ_INJECT] register to set the corresponding VD/VQ values for each sine wave.

$$V = \cos(\text{angle}) * VD - \sin(\text{angle}) * VQ \quad (5)$$

V represents the output value after inverse Park transformation, and angle is the radian value corresponding to the preprocessed input position of QEO.

41.3.2.2 Saddle Wave Mode in Sine Wave Output

When selecting saddle wave output, there are two saddle wave modes. Software can choose between them by configuring WAVE[MODE].SADDLE_TYPE:

- 1) Standard saddle wave. This can only be output correctly when the phase difference between the three output channels is 120°.
- 2) Saddle wave formed by superimposing triangular waves. This mode has no requirement on the phase difference between the three output channels.

41.3.2.3 Sine Wave Scaling and Limiting

Each channel of the sine wave output supports independent scaling and limiting. The scaling factor is controlled by the WAVE[AMPLITUDE] register. The scaling factor can be configured as $n * (1/4096)$, where n can be any integer from 0 to 65535, taken from bits [15:0] of the WAVE[AMPLITUDE] register.

The scaled output can be center-adjusted by WAVE[MID_POINT][WAVE_n]. Portions of the output exceeding the sine wave output range can be limited by configuring WAVE[MODE][WAVE_n_ABOVE_MAX_LIMIT] and WAVE[MODE][WAVE_n_BELOW_MIN_LIMIT]. Range-limiting is also supported within the output range. For details, refer to WAVE[LIMIT][WAVE_n][MAX] and WAVE[LIMIT][WAVE_n][MIN].

41.3.2.4 Sine Wave Dead Zone Compensation

After limiting, each channel of the sine wave output can also undergo dead zone compensation. This can be done by configuring WAVE[DEADZONE_SHIFT][WAVE_n].

41.3.3 ABZ Output

ABZ is a digital pulse output. It supports:

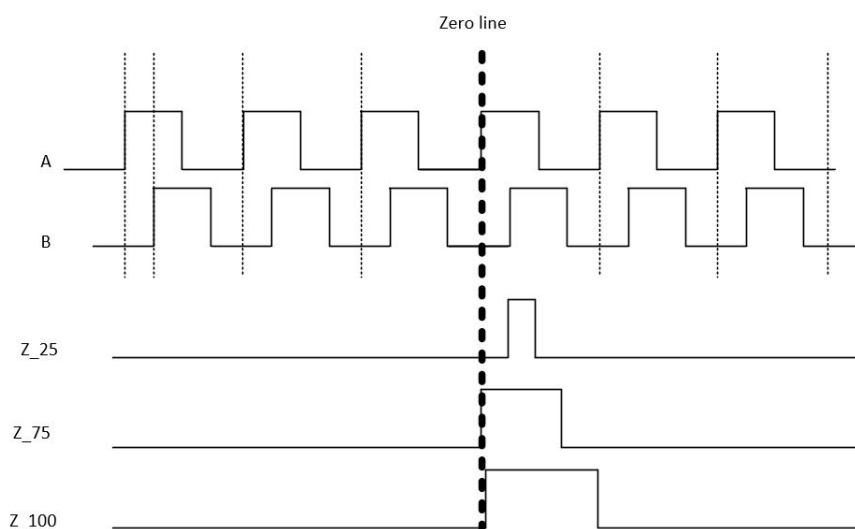
- 1) A/B two-phase quadrature output with its own internal position;
- 2) High-speed Pulse & Direction output;
- 3) Up & Down output;
- 4) A/B two-phase quadrature output without its own internal position.

41.3.3.1 A/B Two-Phase Quadrature Output with Internal Position

The ABZ output module of QEO has a built-in ABZ internal position manager. After a system reset, the internal position is zero. When QEO receives a new input position, the internal position will track the input position with the period set by ABZ[LINE_WIDTH] as the minimum tracking period. Once the internal position matches the input position, tracking stops. The minimum period corresponds to 1/4 of the A, B output line's minimum period.

Software can configure ABZ[POSITION_SYNC] to use the next valid position received by QEO as the position for the internal position manager. This function can be used to set the initial ABZ starting position at startup, preventing erroneous ABZ tracking.

When ABZ is selected as A/B two-phase quadrature output with internal position, the outputs of A, B, and Z are determined by the internal position value.



In the figure above, Z_25, Z_75, and Z_100 are three output modes for the Z phase, controlled by the ABZ[MODE].Z_TYPE register.
Figure 53: ABZ Output Waveforms (Z_25, Z_75, Z_100 are three Z-phase output modes)

register.

41.3.3.2 High-Speed Pulse & Direction Output

The high-speed pulse output mode also uses the ABZ internal position.

The A phase outputs the clock pulse. When the internal position is tracking the input position in the forward direction, the B phase maintains a high-level output; when the internal position is tracking in the reverse direction, the B phase maintains a low-level output.

41.3.3.3 Up & Down Output

The up & down output mode also uses the ABZ internal position.

When the internal position is tracking the input position in the forward direction, the A phase outputs the clock pulse; when tracking in the reverse direction, the B phase outputs the clock pulse.

41.3.3.4 A/B Two-Phase Quadrature Output Without Internal Position

In this mode, the A/B/Z output is determined by the real-time input position to QEO. The minimum period and watchdog configurations are invalid for this mode.

41.3.3.5 Watchdog for Internal Position Manager

To prevent situations where the system cannot determine if it is in normal mode when no position updates are received for a long time, software can configure ABZ[MODE].EN_WDOG and ABZ[WDOG_WIDTH] to enable the watchdog function. Once enabled, if the QEO input position does not change within the set period, the internal position will automatically step forward by 1/4 line, then step backward by 1/4 line, or vice versa.

41.3.4 PWM FORCE Output

QEO provides 8 sets of PWM FORCE control signals. Each set of PWM FORCE corresponds to a 2-bit control signal. The PWM FORCE control signals are determined by looking up the PWM commutation table based on the input position.

The input position comes from the QEO position preprocessing module. Software can configure the number of commutations via PWM[MODE].PHASE_NUM, with a maximum of 12. Then, the commutation table is filled into PWM[PHASE_TABLE][POSEDGEN] or PWM[PHASE_TABLE][NEGEDGEN], where the number of n depends on the number of commutations configured in PWM[MODE].PHASE_NUM. Whether to use PWM[PHASE_TABLE][POSEDGEN] or PWM[PHASE_TABLE][NEGEDGEN] is determined by the PWM[MODE].REVISE_UP_DN register.

This module also provides a safety mode PWM_FORCE lookup table. This lookup table is determined by the PWM[MODE].PWMn_SAFETY register. Two methods to enter safety mode are supported:

- 1) Via TRIG_MUX configuration of QEO.TRIGGER_IN[1]. When the source is high, safety mode is entered.
- 2) By configuring PWM[MODE].PWM_ENTER_SAFETY_MODE to enter safety mode.

41.4 QEO Register List

The QEO register list is as follows:

QEO0 base address: 0xF0410000

QEO1 base address: 0xF0414000

Offset	Register Name	Description	Reset Value
0x0000	WAVE[MODE]	analog waves mode	0x00000000
0x0004	WAVE[RESOLUTION]	resolution of wave0/1/2	0x00000000
0x0008	WAVE[PHASE_SHIFT][WAVE0]	wave0 phase shifter	0x00000000
0x000C	WAVE[PHASE_SHIFT][WAVE1]	wave1 phase shifter	0x00000000
0x0010	WAVE[PHASE_SHIFT][WAVE2]	wave2 phase shifter	0x00000000
0x0014	WAVE[VD_INJECT]	wave vd inject value	0x00000000
0x0020	WAVE[VQ_INJECT]	wave vq inject value	0x00000000
0x002C	WAVE[VD_VQ_LOAD]	load wave0/1/2 vd vq value	0x00000000
0x0030	WAVE[AMPLITUDE][WAVE0]	wave0 amplitude	0x00000000
0x0034	WAVE[AMPLITUDE][WAVE1]	wave1 amplitude	0x00000000
0x0038	WAVE[AMPLITUDE][WAVE2]	wave2 amplitude	0x00000000

Offset	Register Name	Description	Reset Value
0x003C	WAVE[MID_POINT][WAVE0]	wave0 output middle point offset	0x00000000
0x0040	WAVE[MID_POINT][WAVE1]	wave1 output middle point offset	0x00000000
0x0044	WAVE[MID_POINT][WAVE2]	wave2 output middle point offset	0x00000000
0x0048	WAVE[LIMIT0][WAVE0][MIN_LEVEL0]	wave0 low area limit value	0x00000000
0x004C	WAVE[LIMIT0][WAVE0][MAX_LEVEL0]	wave0 high area limit value	0xFFFFFFFF
0x0050	WAVE[LIMIT0][WAVE1][MIN_LEVEL0]	wave1 low area limit value	0x00000000
0x0054	WAVE[LIMIT0][WAVE1][MAX_LEVEL0]	wave1 high area limit value	0xFFFFFFFF
0x0058	WAVE[LIMIT0][WAVE2][MIN_LEVEL0]	wave2 low area limit value	0x00000000
0x005C	WAVE[LIMIT0][WAVE2][MAX_LEVEL0]	wave2 high area limit value	0xFFFFFFFF
0x0060	WAVE[LIMIT1][WAVE0][MIN_LEVEL1]	wave0 low area limit value level1	0x00000000
0x0064	WAVE[LIMIT1][WAVE0][MAX_LEVEL1]	wave0 high area limit value level1	0xFFFFFFFF
0x0068	WAVE[LIMIT1][WAVE1][MIN_LEVEL1]	wave1 low area limit value level1	0x00000000
0x006C	WAVE[LIMIT1][WAVE1][MAX_LEVEL1]	wave1 high area limit value level1	0xFFFFFFFF
0x0070	WAVE[LIMIT1][WAVE2][MIN_LEVEL1]	wave2 low area limit value level1	0x00000000
0x0074	WAVE[LIMIT1][WAVE2][MAX_LEVEL1]	wave2 high area limit value level1	0xFFFFFFFF
0x0078	WAVE[DEADZONE_SHIFT][WAVE0]	deadzone_shifter_wave0	0x00000000
0x007C	WAVE[DEADZONE_SHIFT][WAVE1]	deadzone_shifter_wave1	0x00000000
0x0080	WAVE[DEADZONE_SHIFT][WAVE2]	deadzone_shifter_wave2	0x00000000
0x0084	WAVE[PWM_CYCLE]	pwm_cycle	0x00000000
0x0088	WAVE[FILTER]	wave input position filter	0x00000000
0x0100	ABZ[MODE]	wave_a/b/z output mode	0x00000000
0x0104	ABZ[RESOLUTION]	resolution of wave_a/b/z	0x00000000
0x0108	ABZ[PHASE_SHIFT][A]	wave_a phase shifter	0x00000000
0x010C	ABZ[PHASE_SHIFT][B]	wave_b phase shifter	0x00000000
0x0110	ABZ[PHASE_SHIFT][Z]	wave_z phase shifter	0x00000000
0x0114	ABZ[LINE_WIDTH]	Two-phase orthogonality wave 1/4 period	0x00000000
0x0118	ABZ[WDOG_WIDTH]	wdog width of qeo	0x00000000
0x011C	ABZ[POSTION_SYNC]	sync abz owned postion	0x00000000
0x0120	ABZ[OVERALL_OFFSET]	abz overall position offset	0x00000000
0x0124	ABZ[Z_START]	zero phase start line num	0x00000000
0x0128	ABZ[Z_END]	zero phase end line num	0x00000000

Offset	Register Name	Description	Reset Value
0x012C	ABZ[Z_OFFSET]	zero phase start and end 1/4 line num	0x00000000
0x0130	ABZ[Z_PULSE_WIDTH]	zero pulse width	0x00000000
0x0134	ABZ[FILTER]	abz input position filter	0x00000000
0x0140	PWM[MODE]	pwm mode	0x00000000
0x0144	PWM[RESOLUTION]	resolution of pwm	0x00000000
0x0148	PWM[PHASE_SHIFT][A]	pwm_a phase shifter	0x00000000
0x014C	PWM[PHASE_SHIFT][B]	pwm_b phase shifter	0x00000000
0x0150	PWM[PHASE_SHIFT][C]	pwm_c phase shifter	0x00000000
0x0154	PWM[PHASE_SHIFT][D]	pwm_d phase shifter	0x00000000
0x0158	PWM[PHASE_TABLE][POSEDGE0]	pwm_phase_table 0	0x00000000
0x015C	PWM[PHASE_TABLE][POSEDGE1]	pwm_phase_table 1	0x00000000
0x0160	PWM[PHASE_TABLE][POSEDGE2]	pwm_phase_table 2	0x00000000
0x0164	PWM[PHASE_TABLE][POSEDGE3]	pwm_phase_table 3	0x00000000
0x0168	PWM[PHASE_TABLE][POSEDGE4]	pwm_phase_table 4	0x00000000
0x016C	PWM[PHASE_TABLE][POSEDGE5]	pwm_phase_table 5	0x00000000
0x0170	PWM[PHASE_TABLE][POSEDGE6]	pwm_phase_table 6	0x00000000
0x0174	PWM[PHASE_TABLE][POSEDGE7]	pwm_phase_table 7	0x00000000
0x0178	PWM[PHASE_TABLE][POSEDGE8]	pwm_phase_table 8	0x00000000
0x017C	PWM[PHASE_TABLE][POSEDGE9]	pwm_phase_table 9	0x00000000
0x0180	PWM[PHASE_TABLE][POSEDGE10]	pwm_phase_table 10	0x00000000
0x0184	PWM[PHASE_TABLE][POSEDGE11]	pwm_phase_table 11	0x00000000
0x0188	PWM[PHASE_TABLE][NEGEDGE0]	pwm_phase_table 0	0x00000000
0x018C	PWM[PHASE_TABLE][NEGEDGE1]	pwm_phase_table 1	0x00000000
0x0190	PWM[PHASE_TABLE][NEGEDGE2]	pwm_phase_table 2	0x00000000
0x0194	PWM[PHASE_TABLE][NEGEDGE3]	pwm_phase_table 3	0x00000000
0x0198	PWM[PHASE_TABLE][NEGEDGE4]	pwm_phase_table 4	0x00000000
0x019C	PWM[PHASE_TABLE][NEGEDGE5]	pwm_phase_table 5	0x00000000
0x01A0	PWM[PHASE_TABLE][NEGEDGE6]	pwm_phase_table 6	0x00000000
0x01A4	PWM[PHASE_TABLE][NEGEDGE7]	pwm_phase_table 7	0x00000000
0x01A8	PWM[PHASE_TABLE][NEGEDGE8]	pwm_phase_table 8	0x00000000
0x01AC	PWM[PHASE_TABLE][NEGEDGE9]	pwm_phase_table 9	0x00000000
0x01B0	PWM[PHASE_TABLE][NEGEDGE10]	pwm_phase_table 10	0x00000000
0x01B4	PWM[PHASE_TABLE][NEGEDGE11]	pwm_phase_table 11	0x00000000
0x01B8	PWM[FILTER]	pwm input position filter	0x00000000
0x01F8	POSTION_SOFTWARE	software inject postion	0x00000000
0x01FC	POSTION_SEL	select software inject postion	0x00000000
0x0200	STATUS	qeo status	0x00000000
0x0204	DEBUG0	qeo debug 0	0x00000000
0x0208	DEBUG1	qeo debug 1	0x00000000

Offset	Register Name	Description	Reset Value
0x020C	DEBUG2	qeo debug 2	0x00000000
0x0210	DEBUG3	qeo debug 3	0x00000000
0x0214	DEBUG4	qeo debug 4	0x00000000
0x0218	DEBUG5	qeo debug 5	0x00000000

Table 183: QEOV2 Register List

41.5 QEO Register Details

The detailed description of the QEO registers is as follows:

41.5.1 WAVE[MODE] (0x0)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WAVE2_ABOVE_MAX_LIMIT	WAVE2_HIGH_AREA1_LIMIT	WAVE2_HIGH_AREA0_LIMIT	WAVE2_LOW_AREA1_LIMIT	WAVE2_LOW_AREA0_LIMIT	WAVE2_BELOW_MIN_LIMIT	WAVE1_ABOVE_MAX_LIMIT	WAVE1_HIGH_AREA1_LIMIT	WAVE1_HIGH_AREA0_LIMIT	WAVE1_LOW_AREA1_LIMIT	WAVE1_LOW_AREA0_LIMIT	WAVE1_BELOW_MIN_LIMIT	WAVE0_ABOVE_MAX_LIMIT	WAVE0_HIGH_AREA1_LIMIT	WAVE0_HIGH_AREA0_LIMIT	WAVE0_LOW_AREA1_LIMIT	WAVE0_LOW_AREA0_LIMIT	WAVE0_BELOW_MIN_LIMIT	SADDLE_TYPE	ENABLE_DQ_VALID	ENABLE_POS_VALID	EN_WAVE_VD_VQ_INJECT	RSVD	VD_VQ_SEL	WAVES_OUTPUT_TYPE							
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	N/A	RW	RW	RW	RW
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	0	0	0

WAVE[MODE] [31:0]

Bits	Name	Description
31-30	WAVE2_ABOVE_MAX_LIMIT	wave2 above max limit mode. 0: output all bits are 1. 1: output 0x0. 2: output as level_max_limit2.level0_max_limit
29	WAVE2_HIGH_AR_EA1_LIMIT	wave2 high area1 limit mode. 0: output all bits are 1. 1: output as level_max_limit2.level0_max_limit
28	WAVE2_HIGH_AR_EA0_LIMIT	wave2 high area0 limit mode. 0: output all bits are 1. 1: output as level_max_limit2.level0_max_limit
27	WAVE2_LOW_AR_EA1_LIMIT	wave2 low area1 limit mode. 0: output 0. 1: output as level_min_limit2.level1_min_limit

Bits	Name	Description
26	WAVE2_LOW_AR EA0_LIMIT	wave2 low area0 limit mode. 0: output 0. 1: output as level_min_limit2.level1_min_limit
25-24	WAVE2_BELOW_ MIN_LIMIT	wave2 below min limit mode. 0: output 0. 1: output all bits are 1. 2: output as level_min_limit2.level1_min_limit
23-22	WAVE1_ABOVE_ MAX_LIMIT	wave1 above max limit mode. 0: output all bits are 1. 1: output 0x0. 2: output as level_max_limit1.level0_max_limit
21	WAVE1_HIGH_AR EA1_LIMIT	wave1 high area1 limit mode. 0: output all bits are 1. 1: output as level_max_limit1.level0_max_limit
20	WAVE1_HIGH_AR EA0_LIMIT	wave1 high area0 limit mode. 0: output all bits are 1. 1: output as level_max_limit1.level0_max_limit
19	WAVE1_LOW_AR EA1_LIMIT	wave1 low area1 limit mode. 0: output 0. 1: output as level_min_limit1.level1_min_limit
18	WAVE1_LOW_AR EA0_LIMIT	wave1 low area0 limit mode. 0: output 0. 1: output as level_min_limit1.level1_min_limit
17-16	WAVE1_BELOW_ MIN_LIMIT	wave1 below min limit mode. 0: output 0. 1: output all bits are 1. 2: output as level_min_limit1.level1_min_limit
15-14	WAVE0_ABOVE_ MAX_LIMIT	wave0 above max limit mode. 0: output all bits are 1. 1: output 0x0. 2: output as level_max_limit0.level0_max_limit
13	WAVE0_HIGH_AR EA1_LIMIT	wave0 high area1 limit mode. 0: output all bits are 1. 1: output as level_max_limit0.level0_max_limit
12	WAVE0_HIGH_AR EA0_LIMIT	wave0 high area0 limit mode. 0: output all bits are 1. 1: output as level_max_limit0.level0_max_limit
11	WAVE0_LOW_AR EA1_LIMIT	wave0 low area1 limit mode. 0: output 0. 1: output as level_min_limit0.level1_min_limit

Bits	Name	Description
10	WAVE0_LOW_AR EA0_LIMIT	wave0 low area0 limit mode. 0: output 0. 1: output as level_min_limit0.level1_min_limit
9-8	WAVE0_BELOW_ MIN_LIMIT	wave0 below min limit mode. 0: output 0. 1: output all bits are 1. 2: output as level_min_limit0.level1_min_limit
7	SADDLE_TYPE	saddle type select; 0: standard saddle. 1: triple-cos saddle.
6	ENABLE_DQ_VAL ID	enable vd or vq valid to trigger analog wave calculation 0: disable. 1: enable.
5	ENABLE_POS_VA LID	enable position valid to trigger analog wave calculation 0: disable. 1: enable.
4	EN_WAVE_VD_V Q_INJECT	wave VdVq inject enable. 0: disable VdVq inject. 1: enable VdVq inject.
2	VD_VQ_SEL	vd_vq sel ctrl: 0: from CLC. 1: from software.
1-0	WAVES_OUTPUT _TYPE	wave0/1/2 output mode. 0: cosine wave. 1: saddle wave. 2: abs cosine wave. 3: saw wave

WAVE[MODE] Bits

41.5.2 WAVE[RESOLUTION] (0x4)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LINES																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

WAVE[RESOLUTION] [31:0]

Bits	Name	Description
31-0	LINES	wave0/1/2 resolution

Bits	Name	Description

WAVE[RESOLUTION] Bits

41.5.3 WAVE[PHASE_SHIFT] (0x8 + 0x4 * m)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VAL																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

WAVE[PHASE_SHIFT] [31:0]

Bits	Name	Description
31-0	VAL	wave0 phase shifter value, default is 0x0. write other value will shift phase early as (cfg_value/2^32) period

WAVE[PHASE_SHIFT] Bits

41.5.4 WAVE[VD_INJECT] (0x14)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VD_VAL																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

WAVE[VD_INJECT] [31:0]

Bits	Name	Description
31-0	VD_VAL	Vd inject value

WAVE[VD_INJECT] Bits

41.5.5 WAVE[VQ_INJECT] (0x20)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VQ_VAL																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

WAVE[VQ_INJECT] [31:0]

Bits	Name	Description
31-0	VQ_VAL	Vq inject value

WAVE[VQ_INJECT] Bits

41.5.6 WAVE[VD_VQ_LOAD] (0x2C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																LOAD															
N/A																WO															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0

WAVE[VD_VQ_LOAD] [31:0]

Bits	Name	Description
0	LOAD	load wave0/1/2 vd vq value. always read 0 0: vd vq keep previous value. 1: load wave0/1/2 vd vq value at sametime.

WAVE[VD_VQ_LOAD] Bits

41.5.7 WAVE[AMPLITUDE] (0x30 + 0x4 * m)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD															EN_SCAL	AMP_VAL															
N/A															RW	RW															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

WAVE[AMPLITUDE] [31:0]

Bits	Name	Description
16	EN_SCAL	enable wave amplitude scaling. 0: disable; 1: enable
15-0	AMP_VAL	amplitude scaling value. bit15-12 are integer part value. bit11-0 are fraction value.

WAVE[AMPLITUDE] Bits

41.5.8 WAVE[MID_POINT] (0x3C + 0x4 * m)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
VAL																																

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

WAVE[MID_POINT] [31:0]

Bits	Name	Description
31-0	VAL	wave0 output middle point, use this value as 32 bit signed value. bit 31 is signed bit. bit30-27 is integer part value. bit26-0 is fraction value.

WAVE[MID_POINT] Bits

41.5.9 WAVE[LIMIT0][MIN_LEVEL0] (0x48 + 0x8 * m)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LIMIT_LEVEL0																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

WAVE[LIMIT0][MIN_LEVEL0] [31:0]

Bits	Name	Description
31-0	LIMIT_LEVEL0	low area limit level0

WAVE[LIMIT0][MIN_LEVEL0] Bits

41.5.10 WAVE[LIMIT0][MAX_LEVEL0] (0x4C + 0x8 * m)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LIMIT_LEVEL0																															
RW																															
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

WAVE[LIMIT0][MAX_LEVEL0] [31:0]

Bits	Name	Description
31-0	LIMIT_LEVEL0	high area limit level0

Bits	Name	Description

WAVE[LIMIT0][MAX_LEVEL0] Bits

41.5.11 WAVE[LIMIT1][MIN_LEVEL1] (0x60 + 0x8 * m)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LIMIT_LEVEL1																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

WAVE[LIMIT1][MIN_LEVEL1] [31:0]

Bits	Name	Description
31-0	LIMIT_LEVEL1	low area limit level1

WAVE[LIMIT1][MIN_LEVEL1] Bits

41.5.12 WAVE[LIMIT1][MAX_LEVEL1] (0x64 + 0x8 * m)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LIMIT_LEVEL1																															
RW																															
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

WAVE[LIMIT1][MAX_LEVEL1] [31:0]

Bits	Name	Description
31-0	LIMIT_LEVEL1	high area limit level1

WAVE[LIMIT1][MAX_LEVEL1] Bits

41.5.13 WAVE[DEADZONE_SHIFT] (0x78 + 0x4 * m)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VAL																															
RW																															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

WAVE[DEADZONE_SHIFT] [31:0]

Bits	Name	Description
31-0	VAL	wave0 deadzone shifter value

WAVE[DEADZONE_SHIFT] Bits

41.5.14 WAVE[PWM_CYCLE] (0x84)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VAL																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

WAVE[PWM_CYCLE] [31:0]

Bits	Name	Description
31-0	VAL	pwm_cycle

WAVE[PWM_CYCLE] Bits

41.5.15 WAVE[FILTER] (0x88)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VAL																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

WAVE[FILTER] [31:0]

Bits	Name	Description
31-0	VAL	wave_filter

WAVE[FILTER] Bits

41.5.16 ABZ[MODE] (0x100)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABZ_OUTPUT_ENABLE			REVERSE_EDGE_TYPE	POSITION_SYNC_MODE			EN_WDOG				Z_POLARITY				B_POLARITY				A_POLARITY				Z_TYPE			B_TYPE					A_TYPE
RW	N/A		RW	RW	N/A		RW		N/A		RW	N/A			RW	N/A			RW	N/A		RW	N/A			RW	N/A				RW
0	x	x	0	0	x	x	0	x	x	x	0	x	x	x	0	x	x	x	0	x	x	0	0	x	x	0	0	x	x	0	0

ABZ[MODE] [31:0]

Bits	Name	Description
31	ABZ_OUTPUT_ENABLE	abz output enable: 0: abz output disable, all keep 0 1: abz output enable.
28	REVERSE_EDGE_TYPE	pulse reverse wave, reverse edge point: 0: between pulse's posedge and negedge, min period dedicated by the num line_width 1: edge change point flow pulse's negedge.
27	POSITION_SYNC_MODE	position sync mode: 0: only sync integer line part into qeo own position. 1: sync integer and fraction part into qeo own position.
24	EN_WDOG	enable abz wdog: 0: disable abz wdog. 1: enable abz wdog.
20	Z_POLARITY	wave_z polarity. 0: normal output. 1: invert normal output
16	B_POLARITY	wave_b polarity. 0: normal output. 1: invert normal output
12	A_POLARITY	wave_a polarity. 0: normal output. 1: invert normal output
9-8	Z_TYPE	wave_z type: 0: zero pulse type, start and end line number decided by z_start, z_end and z_offset. 1: zero pulse type, z output start to high when position= z_start, and mantain numbers of 1/4 line cfg in z_pulse_width register 2: reserved 3: wave_z output as tree-phase wave same as wave_a/wave_b

Bits	Name	Description
5-4	B_TYPE	wave_b type: 0: Two-phase orthogonality wave_b. 1: reverse wave of pulse/reverse type. 2: down wave of up/down type. 3: Three-phase orthogonality wave_b.
1-0	A_TYPE	wave_a type: 0: Two-phase orthogonality wave_a. 1: pulse wave of pulse/reverse type. 2: up wave of up/down type. 3: Three-phase orthogonality wave_a.

ABZ[MODE] Bits

41.5.17 ABZ[RESOLUTION] (0x104)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LINES																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ABZ[RESOLUTION] [31:0]

Bits	Name	Description
31-0	LINES	wave_a/b/z resolution

ABZ[RESOLUTION] Bits

41.5.18 ABZ[PHASE_SHIFT] (0x108 + 0x4 * m)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VAL																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ABZ[PHASE_SHIFT] [31:0]

Bits	Name	Description
31-0	VAL	wave_a phase shifter value, default is 0x0. write other value will shift phase early as (cfg_value/2^32) period.

ABZ[PHASE_SHIFT] Bits

41.5.19 ABZ[LINE_WIDTH] (0x114)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LINE																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ABZ[LINE_WIDTH] [31:0]

Bits	Name	Description
31-0	LINE	the num of system clk by 1/4 period when using as Two-phase orthogonality.

ABZ[LINE_WIDTH] Bits

41.5.20 ABZ[WDOG_WIDTH] (0x118)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WIDTH																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ABZ[WDOG_WIDTH] [31:0]

Bits	Name	Description
31-0	WIDTH	wave will step 1/4 line to reminder user QEO still in controlled if QEO has no any toggle after the num of wdog_width sys clk.

ABZ[WDOG_WIDTH] Bits

41.5.21 ABZ[POSTION_SYNC] (0x11C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																															POSITION
N/A																															WO
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	

ABZ[POSTION_SYNC] [31:0]

Bits	Name	Description
0	POSTION	load next valid postion into abz owned postion. always read 0 0: sync abz owned postion with next valid postion. 1: not sync.

ABZ[POSTION_SYNC] Bits

41.5.22 ABZ[OVERALL_OFFSET] (0x120)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VAL																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ABZ[OVERALL_OFFSET] [31:0]

Bits	Name	Description
31-0	VAL	abz position overall offset, it affects abz position before resolution convert

ABZ[OVERALL_OFFSET] Bits

41.5.23 ABZ[Z_START] (0x124)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Z_START																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ABZ[Z_START] [31:0]

Bits	Name	Description
31-0	Z_START	number of Z start line

ABZ[Z_START] Bits

41.5.24 ABZ[Z_END] (0x128)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Z_END																															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ABZ[Z_END] [31:0]

Bits	Name	Description
31-0	Z_END	number of Z end line

ABZ[Z_END] Bits

41.5.25 ABZ[Z_OFFSET] (0x12C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																						Z_END_OFFSET		RSVD		Z_START_OFFSET		RSVD			
N/A																						RW		N/A		RW		N/A			
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	x	x	0	0	x	x	x	x

ABZ[Z_OFFSET] [31:0]

Bits	Name	Description
9-8	Z_END_OFFSET	number of Z end 1/4 line
5-4	Z_START_OFFSET	number of Z start 1/4 line

ABZ[Z_OFFSET] Bits

41.5.26 ABZ[Z_PULSE_WIDTH] (0x130)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VAL																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ABZ[Z_PULSE_WIDTH] [31:0]

Bits	Name	Description
31-0	VAL	number of z_pulse_width

ABZ[Z_PULSE_WIDTH] Bits

41.5.27 ABZ[FILTER] (0x134)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VAL																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ABZ[FILTER] [31:0]

Bits	Name	Description
31-0	VAL	abz_filter

ABZ[FILTER] Bits

41.5.28 PWM[MODE] (0x140)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																										
PWM7_SAFETY				PWM6_SAFETY				PWM5_SAFETY				PWM4_SAFETY				PWM3_SAFETY				PWM2_SAFETY				PWM1_SAFETY				PWM0_SAFETY				ENABLE_PWM				RSVD						PWM_ENTER_SAFETY_MODE		PWM_SAFETY_BYPASS		RSVD				REVISE_UP_DN		PHASE_NUM					
RW				RW				RW				RW				RW				RW				RW				RW		RW		N/A										RW		RW		N/A				RW		RW					
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x	x	x	x	0	0	0	x	x	x	0	0	0	0	0	0																								

PWM[MODE] [31:0]

Bits	Name	Description
31-30	PWM7_SAFETY	PWM safety mode phase table
29-28	PWM6_SAFETY	PWM safety mode phase table
27-26	PWM5_SAFETY	PWM safety mode phase table
25-24	PWM4_SAFETY	PWM safety mode phase table
23-22	PWM3_SAFETY	PWM safety mode phase table
21-20	PWM2_SAFETY	PWM safety mode phase table
19-18	PWM1_SAFETY	PWM safety mode phase table
17-16	PWM0_SAFETY	PWM safety mode phase table
15	ENABLE_PWM	enable PWM force output 0: disable 1: enable

Bits	Name	Description
9	PWM_ENTER_SAFETY_MODE	PWM enter safety mode 0: not enter 1: enter
8	PWM_SAFETY_BYPASS	PWM safety mode bypass 0: not bypass 1: bypass
4	REVISE_UP_DN	exchange PWM pairs' output 0: not exchange. 1: exchange.
3-0	PHASE_NUM	pwm force phase number.

PWM[MODE] Bits

41.5.29 PWM[RESOLUTION] (0x144)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LINES																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PWM[RESOLUTION] [31:0]

Bits	Name	Description
31-0	LINES	pwm resolution

PWM[RESOLUTION] Bits

41.5.30 PWM[PHASE_SHIFT] (0x148 + 0x4 * m)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VAL																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PWM[PHASE_SHIFT] [31:0]

Bits	Name	Description
31-0	VAL	pwm_a phase shifter value, default is 0x0. write other value will shift phase early as (cfg_value/2^32) period

PWM[PHASE_SHIFT] Bits

41.5.31 PWM[PHASE_TABLE] (0x158 + 0x4 * m)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																PWM7	PWM6	PWM5	PWM4	PWM3	PWM2	PWM1	PWM0								
N/A																RW	RW	RW	RW	RW	RW	RW	RW								
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

PWM[PHASE_TABLE] [31:0]

Bits	Name	Description
15-14	PWM7	pwm phase table value
13-12	PWM6	pwm phase table value
11-10	PWM5	pwm phase table value
9-8	PWM4	pwm phase table value
7-6	PWM3	pwm phase table value
5-4	PWM2	pwm phase table value
3-2	PWM1	pwm phase table value
1-0	PWM0	pwm phase table value

PWM[PHASE_TABLE] Bits

41.5.32 PWM[FILTER] (0x1B8)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VAL																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PWM[FILTER] [31:0]

Bits	Name	Description
31-0	VAL	pwm_filter

PWM[FILTER] Bits

41.5.33 POSTION_SOFTWARE (0x1F8)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																POSTION_SOFTWARE															
																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

POSTION_SOFTWARE [31:0]

Bits	Name	Description
31-0	POSTION_SOFT WAVE	software inject postion

POSTION_SOFTWARE Bits

41.5.34 POSTION_SEL (0x1FC)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																RSVD															
																N/A															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0

POSTION_SEL [31:0]

Bits	Name	Description
0	POSTION_SEL	enable software inject postion. 0: disable. 1: enable.

POSTION_SEL Bits

41.5.35 STATUS (0x200)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																RSVD															
																N/A															
PWM_FOURCE																PWM_SAFETY															
RO																RO															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0

STATUS [31:0]

Bits	Name	Description
31-16	PWM_FOURCE	qeo_pwm_force observe
0	PWM_SAFETY	pwm_fault status

STATUS Bits

41.5.36 DEBUG0 (0x204)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																VALUE_DAC0															
																RO															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DEBUG0 [31:0]

Bits	Name	Description
31-0	VALUE_DAC0	wave0

DEBUG0 Bits

41.5.37 DEBUG1 (0x208)

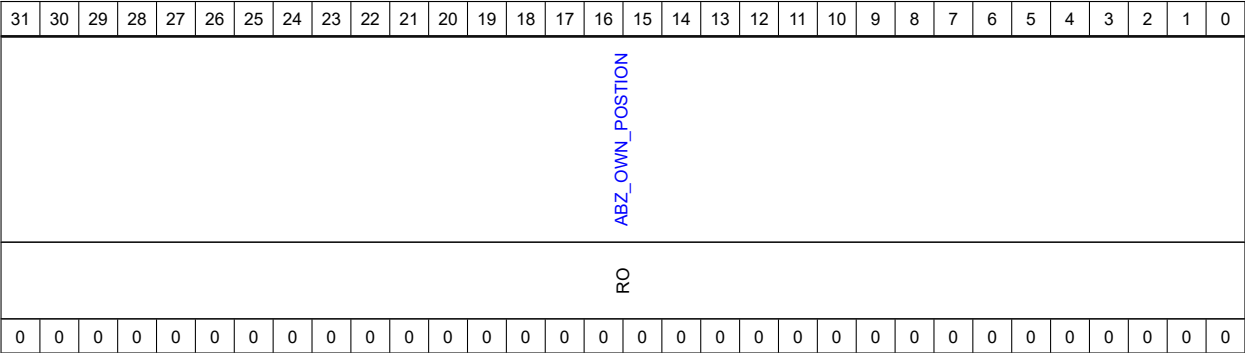
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD			QEO_FINISH	RSVD			PAD_Z	RSVD			PAD_B	RSVD			PAD_A	RSVD															
N/A			RO	N/A			RO	N/A			RO	N/A			RO	N/A															
x	x	x	0	x	x	x	0	x	x	x	0	x	x	x	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

DEBUG1 [31:0]

Bits	Name	Description
28	QEO_FINISH	qeo finish observe
24	PAD_Z	pad_z observe
20	PAD_B	pad_b observe
16	PAD_A	pad_a observe

DEBUG1 Bits

41.5.38 DEBUG2 (0x20C)

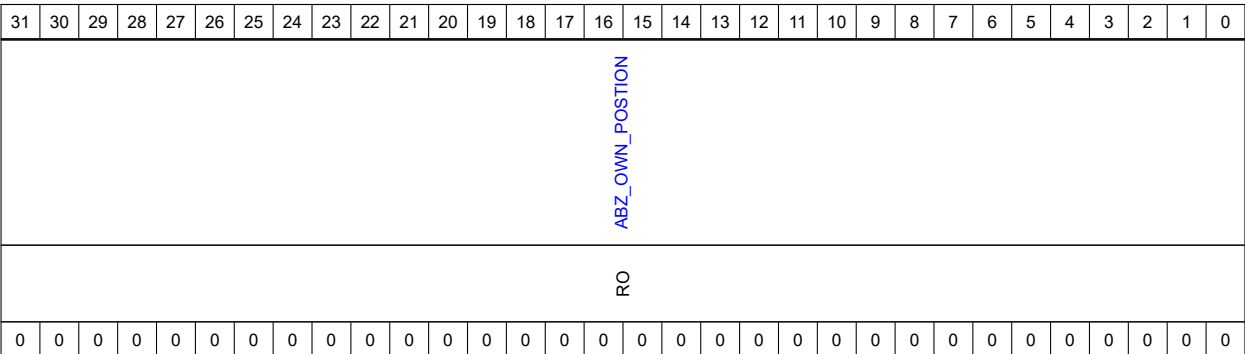


DEBUG2 [31:0]

Bits	Name	Description
31-0	ABZ_OWN_POSITION	abz_own_postion observe

DEBUG2 Bits

41.5.39 DEBUG3 (0x210)

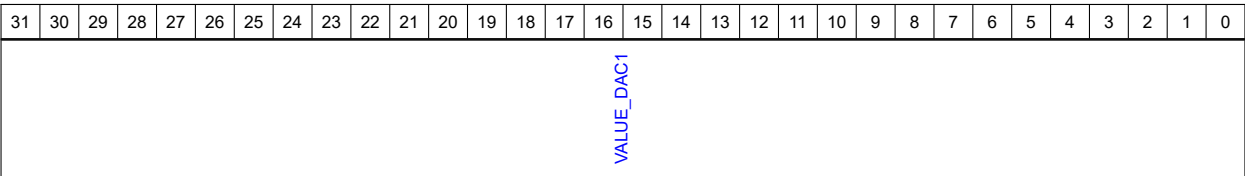


DEBUG3 [31:0]

Bits	Name	Description
31-0	ABZ_OWN_POSITION	abz_own_postion observe

DEBUG3 Bits

41.5.40 DEBUG4 (0x214)



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DEBUG4 [31:0]

Bits	Name	Description
31-0	VALUE_DAC1	wave1

DEBUG4 Bits

41.5.41 DEBUG5 (0x218)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE_DAC2																															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DEBUG5 [31:0]

Bits	Name	Description
31-0	VALUE_DAC2	wave2

DEBUG5 Bits

42 Programmable Logic Block (PLB)

This chapter introduces the main functions and features of the Programmable Logic Block (PLB).

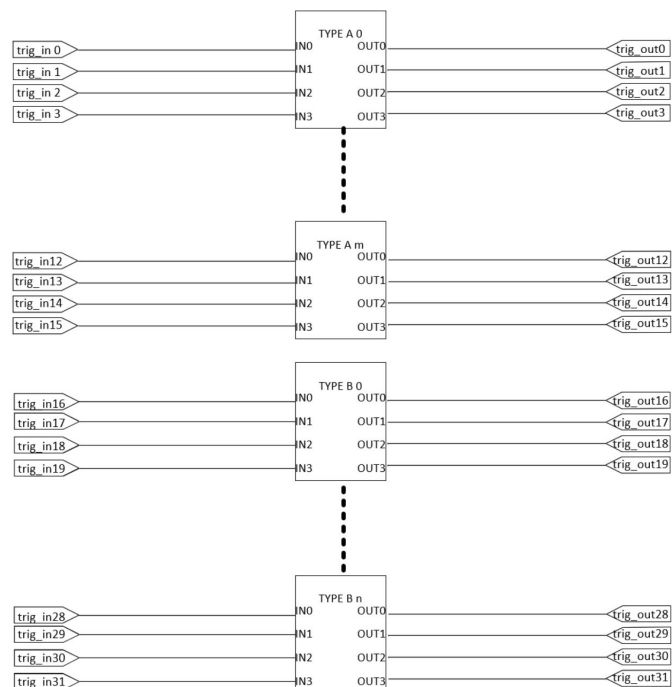
42.1 Feature Summary

This chapter introduces the main features of the Programmable Logic Block (PLB):

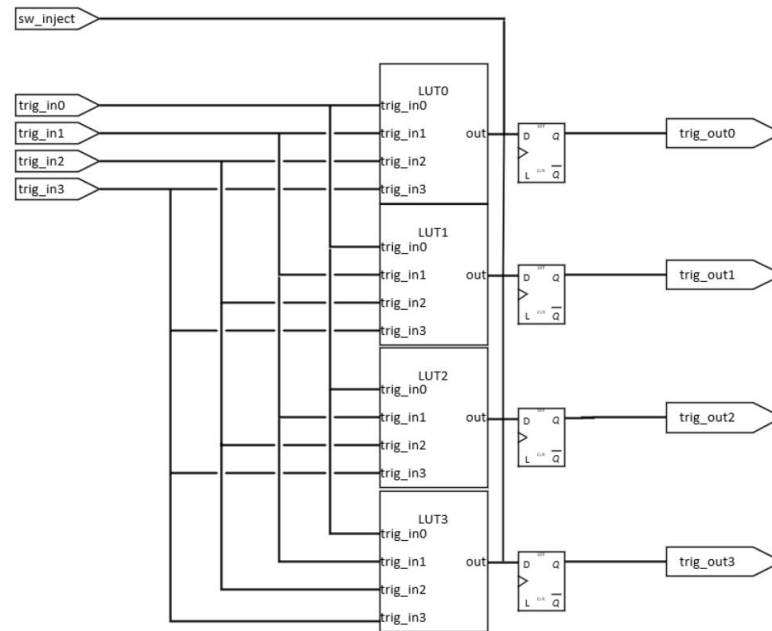
- Two programmable types
- TYPE_A can implement combinational logic + sequential logic programming
- TYPE_B can implement complex operations such as counting, shifting, and filtering

42.2 Block Diagram

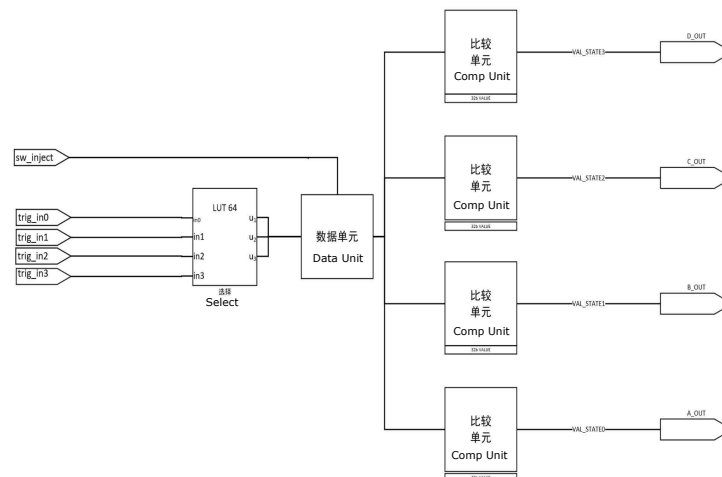
PLB inputs and outputs are connected in TRIG_MUX. The overall structure is shown in the figure below:



The TYPE A structure is shown in the figure below:



The TYPE B structure is shown in the figure below:



42.3 Functional Description

42.3.1 TYPE A Functional Description

The PLB has 8 built-in TYPE A channels. Each TYPE A channel has 4 trig_in inputs and 4 trig_out outputs. Each trig_out output corresponds to a lookup table unit.

These 4 lookup tables all use 4-bit trig_in as the index. The content of the lookup tables can be considered as the truth table for the 4-bit trig_in. These 4 lookup tables use the same trig_in. Different TYPE A modules use trig_in in sequential increments. For example, TYPE A 0 uses trig_in[3:0], and TYPE A 1 uses trig_in[7:4].

The output of the lookup table is output to trig_out via a register.

Software can configure TYPE_A[SW_INJECT] to directly inject a value onto trig_out. This injected value is held for one system clock cycle, after which the trig_out value continues to be determined by trig_in and the lookup table.

42.3.2 TYPE B Functional Description

The PLB has 8 built-in TYPE B channels. Each TYPE B channel has 4 trig_in inputs and 4 trig_out outputs.

Each TYPE B channel has a 64-bit lookup table. This lookup table is formed by concatenating TYPE_B[LUT][0] and TYPE_B[LUT][1] into a 64-bit double-word. lut0 occupies bits 31:0, and lut1 occupies bits 63:32. Then the 64-bit word is viewed as 16 slices of 4 bits each. slice0 corresponds to bits 3:0, slice1 corresponds to bits 7:4, and so on, with slice15 corresponding to bits 63:60. Each TYPE B channel is allocated 4 trig_in bits. These 4 trig_in bits serve as the index to select a slice. Thus, each channel generates a corresponding 4-bit operation selection, which determines the operation of the TYPE B data unit.

0: The data unit maintains its value from the previous cycle.

1: The data unit will take the value of the cmp[0] register of this channel as its next cycle value.

2: The data unit will take the value of the cmp[1] register of this channel as its next cycle value.

3: The data unit will take the value of the cmp[2] register of this channel as its next cycle value.

4: When MODE.OPT_SEL=0, the next cycle value of the data unit is the current value + 1; when MODE.OPT_SEL=1, the next cycle value of the data unit is the current value shifted left by 1 bit, with 0 filled in the LSB.

5: When MODE.OPT_SEL=0, the next cycle value of the data unit is the current value + 2; when MODE.OPT_SEL=1, the next cycle value of the data unit is the current value shifted left by 1 bit, with 1 filled in the LSB.

6: When MODE.OPT_SEL=0, the next cycle value of the data unit is the current value - 1; when MODE.OPT_SEL=1, the next cycle value of the data unit is the current value shifted right by 1 bit, with 0 filled in the MSB.

7: When MODE.OPT_SEL=0, the next cycle value of the data unit is the current value - 2; when MODE.OPT_SEL=1, the next cycle value of the data unit is the current value shifted right by 1 bit, with 1 filled in the MSB.

Other values: The data unit value remains unchanged.

Each TYPE B channel contains 4 comparison units. The output of each comparison unit corresponds to its respective trig_out.

The operation mode of the comparison unit is determined by TYPE_B[MODE].OUTn_SEL.

0: Output 0

1: Output 1

2: Whether the data unit data is greater than the value of TYPE_B[cmp]. If greater, output 1; otherwise, output 0.

3: Whether the data unit data is less than the value of TYPE_B[cmp]. If less, output 1; otherwise, output 0.

4: Whether the data unit data is equal to the value of TYPE_B[cmp]. If equal, output 1; otherwise, output 0.

5: Whether the data unit data is not equal to the value of TYPE_B[cmp]. If not equal, output 1; otherwise, output 0.

6: Whether the data unit data is greater than or equal to the value of TYPE_B[cmp]. If greater than or equal, output 1; if less, output 0.

7: Whether the data unit data is less than or equal to the value of TYPE_B[cmp]. If less than or equal, output 1; if greater, output 0.

8: Output 0

9: Output 1

10: Perform an AND operation on the bit positions of the data unit data where the corresponding bit in TYPE_B[*CMP*] is 1.

11: Perform an OR operation on the bit positions of the data unit data where the corresponding bit in TYPE_B[*CMP*] is 1.

12: Perform an XOR operation on the bit positions of the data unit data where the corresponding bit in TYPE_B[*CMP*] is 1.

13: Perform a NAND operation on the bit positions of the data unit data where the corresponding bit in TYPE_B[*CMP*] is 1.

14: Perform a NOR operation on the bit positions of the data unit data where the corresponding bit in TYPE_B[*CMP*] is 1.

15: Perform an XNOR operation on the bit positions of the data unit data where the corresponding bit in TYPE_B[*CMP*] is 1.

The comparison units in TYPE B have the function of latching the data unit value. It is triggered by the 4-bit trig_in through the LUT_CMP lookup table content. When a latch event occurs, the current value of the data unit is latched into the comparison unit and can be read via the bus from TYPE_B[*CMP*][*n*].

42.4 PLB Register List

The PLB register list is as follows:

PLB base address: 0xF0460000

Offset	Register Name	Description	Reset Value
0x0000	TYPE_A[0][LOOKUP_TABLE][0]	TYPE A CHN0 lookup_table 0	0x00000000
0x0004	TYPE_A[0][LOOKUP_TABLE][1]	TYPE A CHN0 lookup_table 1	0x00000000
0x0008	TYPE_A[0][LOOKUP_TABLE][2]	TYPE A CHN0 lookup_table 2	0x00000000
0x000C	TYPE_A[0][LOOKUP_TABLE][3]	TYPE A CHN0 lookup_table 3	0x00000000
0x0010	TYPE_A[0][SW_INJECT]	TYPE A CHN0 software inject	0x00000000
0x0020	TYPE_A[1][LOOKUP_TABLE][0]	TYPE A CHN1 lookup_table 0	0x00000000
0x0024	TYPE_A[1][LOOKUP_TABLE][1]	TYPE A CHN1 lookup_table 1	0x00000000
0x0028	TYPE_A[1][LOOKUP_TABLE][2]	TYPE A CHN1 lookup_table 2	0x00000000
0x002C	TYPE_A[1][LOOKUP_TABLE][3]	TYPE A CHN1 lookup_table 3	0x00000000
0x0030	TYPE_A[1][SW_INJECT]	TYPE A CHN1 software inject	0x00000000
0x0040	TYPE_A[2][LOOKUP_TABLE][0]	TYPE A CHN2 lookup_table 0	0x00000000
0x0044	TYPE_A[2][LOOKUP_TABLE][1]	TYPE A CHN2 lookup_table 1	0x00000000
0x0048	TYPE_A[2][LOOKUP_TABLE][2]	TYPE A CHN2 lookup_table 2	0x00000000
0x004C	TYPE_A[2][LOOKUP_TABLE][3]	TYPE A CHN2 lookup_table 3	0x00000000
0x0050	TYPE_A[2][SW_INJECT]	TYPE A CHN2 software inject	0x00000000
0x0060	TYPE_A[3][LOOKUP_TABLE][0]	TYPE A CHN3 lookup_table 0	0x00000000
0x0064	TYPE_A[3][LOOKUP_TABLE][1]	TYPE A CHN3 lookup_table 1	0x00000000
0x0068	TYPE_A[3][LOOKUP_TABLE][2]	TYPE A CHN3 lookup_table 2	0x00000000

Offset	Register Name	Description	Reset Value
0x006C	TYPE_A[3][LOOKUP_TABLE][3]	TYPE A CHN3 lookup_table 3	0x00000000
0x0070	TYPE_A[3][SW_INJECT]	TYPE A CHN3 software inject	0x00000000
0x0080	TYPE_A[4][LOOKUP_TABLE][0]	TYPE A CHN4 lookup_table 0	0x00000000
0x0084	TYPE_A[4][LOOKUP_TABLE][1]	TYPE A CHN4 lookup_table 1	0x00000000
0x0088	TYPE_A[4][LOOKUP_TABLE][2]	TYPE A CHN4 lookup_table 2	0x00000000
0x008C	TYPE_A[4][LOOKUP_TABLE][3]	TYPE A CHN4 lookup_table 3	0x00000000
0x0090	TYPE_A[4][SW_INJECT]	TYPE A CHN4 software inject	0x00000000
0x00A0	TYPE_A[5][LOOKUP_TABLE][0]	TYPE A CHN5 lookup_table 0	0x00000000
0x00A4	TYPE_A[5][LOOKUP_TABLE][1]	TYPE A CHN5 lookup_table 1	0x00000000
0x00A8	TYPE_A[5][LOOKUP_TABLE][2]	TYPE A CHN5 lookup_table 2	0x00000000
0x00AC	TYPE_A[5][LOOKUP_TABLE][3]	TYPE A CHN5 lookup_table 3	0x00000000
0x00B0	TYPE_A[5][SW_INJECT]	TYPE A CHN5 software inject	0x00000000
0x00C0	TYPE_A[6][LOOKUP_TABLE][0]	TYPE A CHN6 lookup_table 0	0x00000000
0x00C4	TYPE_A[6][LOOKUP_TABLE][1]	TYPE A CHN6 lookup_table 1	0x00000000
0x00C8	TYPE_A[6][LOOKUP_TABLE][2]	TYPE A CHN6 lookup_table 2	0x00000000
0x00CC	TYPE_A[6][LOOKUP_TABLE][3]	TYPE A CHN6 lookup_table 3	0x00000000
0x00D0	TYPE_A[6][SW_INJECT]	TYPE A CHN6 software inject	0x00000000
0x00E0	TYPE_A[7][LOOKUP_TABLE][0]	TYPE A CHN7 lookup_table 0	0x00000000
0x00E4	TYPE_A[7][LOOKUP_TABLE][1]	TYPE A CHN7 lookup_table 1	0x00000000
0x00E8	TYPE_A[7][LOOKUP_TABLE][2]	TYPE A CHN7 lookup_table 2	0x00000000
0x00EC	TYPE_A[7][LOOKUP_TABLE][3]	TYPE A CHN7 lookup_table 3	0x00000000
0x00F0	TYPE_A[7][SW_INJECT]	TYPE A CHN7 software inject	0x00000000
0x0400	TYPE_B[0][LUT][0]	TYPE B CHN0 lookup table 0	0x00000000
0x0404	TYPE_B[0][LUT][1]	TYPE B CHN0 lookup table 1	0x00000000
0x0408	TYPE_B[0][CMP][0]	TYPE B CHN0 data unit cmp0	0x00000000
0x040C	TYPE_B[0][CMP][1]	TYPE B CHN0 data unit cmp1	0x00000000
0x0410	TYPE_B[0][CMP][2]	TYPE B CHN0 data unit cmp2	0x00000000
0x0414	TYPE_B[0][CMP][3]	TYPE B CHN0 data unit cmp3	0x00000000
0x0418	TYPE_B[0][MODE]	TYPE B CHN0 mode ctrl	0x00000000
0x041C	TYPE_B[0][SW_INJECT]	TYPE B CHN0 software inject	0x00000000
0x0420	TYPE_B[0][LUT_CMP][0]	TYPE B CHN0 cmp0 lut	0x00000000
0x0424	TYPE_B[0][LUT_CMP][1]	TYPE B CHN0 cmp1 lut	0x00000000
0x0428	TYPE_B[0][LUT_CMP][2]	TYPE B CHN0 cmp2 lut	0x00000000
0x042C	TYPE_B[0][LUT_CMP][3]	TYPE B CHN0 cmp3 lut	0x00000000
0x0480	TYPE_B[1][LUT][0]	TYPE B CHN1 lookup table 0	0x00000000
0x0484	TYPE_B[1][LUT][1]	TYPE B CHN1 lookup table 1	0x00000000
0x0488	TYPE_B[1][CMP][0]	TYPE B CHN1 data unit cmp0	0x00000000
0x048C	TYPE_B[1][CMP][1]	TYPE B CHN1 data unit cmp1	0x00000000
0x0490	TYPE_B[1][CMP][2]	TYPE B CHN1 data unit cmp2	0x00000000
0x0494	TYPE_B[1][CMP][3]	TYPE B CHN1 data unit cmp3	0x00000000

Offset	Register Name	Description	Reset Value
0x0498	TYPE_B[1][MODE]	TYPE B CHN1 mode ctrl	0x00000000
0x049C	TYPE_B[1][SW_INJECT]	TYPE B CHN1 software inject	0x00000000
0x04A0	TYPE_B[1][LUT_CMP][0]	TYPE B CHN1 cmp0 lut	0x00000000
0x04A4	TYPE_B[1][LUT_CMP][1]	TYPE B CHN1 cmp1 lut	0x00000000
0x04A8	TYPE_B[1][LUT_CMP][2]	TYPE B CHN1 cmp2 lut	0x00000000
0x04AC	TYPE_B[1][LUT_CMP][3]	TYPE B CHN1 cmp3 lut	0x00000000
0x0500	TYPE_B[2][LUT][0]	TYPE B CHN2 lookup table 0	0x00000000
0x0504	TYPE_B[2][LUT][1]	TYPE B CHN2 lookup table 1	0x00000000
0x0508	TYPE_B[2][CMP][0]	TYPE B CHN2 data unit cmp0	0x00000000
0x050C	TYPE_B[2][CMP][1]	TYPE B CHN2 data unit cmp1	0x00000000
0x0510	TYPE_B[2][CMP][2]	TYPE B CHN2 data unit cmp2	0x00000000
0x0514	TYPE_B[2][CMP][3]	TYPE B CHN2 data unit cmp3	0x00000000
0x0518	TYPE_B[2][MODE]	TYPE B CHN2 mode ctrl	0x00000000
0x051C	TYPE_B[2][SW_INJECT]	TYPE B CHN2 software inject	0x00000000
0x0520	TYPE_B[2][LUT_CMP][0]	TYPE B CHN2 cmp0 lut	0x00000000
0x0524	TYPE_B[2][LUT_CMP][1]	TYPE B CHN2 cmp1 lut	0x00000000
0x0528	TYPE_B[2][LUT_CMP][2]	TYPE B CHN2 cmp2 lut	0x00000000
0x052C	TYPE_B[2][LUT_CMP][3]	TYPE B CHN2 cmp3 lut	0x00000000
0x0580	TYPE_B[3][LUT][0]	TYPE B CHN3 lookup table 0	0x00000000
0x0584	TYPE_B[3][LUT][1]	TYPE B CHN3 lookup table 1	0x00000000
0x0588	TYPE_B[3][CMP][0]	TYPE B CHN3 data unit cmp0	0x00000000
0x058C	TYPE_B[3][CMP][1]	TYPE B CHN3 data unit cmp1	0x00000000
0x0590	TYPE_B[3][CMP][2]	TYPE B CHN3 data unit cmp2	0x00000000
0x0594	TYPE_B[3][CMP][3]	TYPE B CHN3 data unit cmp3	0x00000000
0x0598	TYPE_B[3][MODE]	TYPE B CHN3 mode ctrl	0x00000000
0x059C	TYPE_B[3][SW_INJECT]	TYPE B CHN3 software inject	0x00000000
0x05A0	TYPE_B[3][LUT_CMP][0]	TYPE B CHN3 cmp0 lut	0x00000000
0x05A4	TYPE_B[3][LUT_CMP][1]	TYPE B CHN3 cmp1 lut	0x00000000
0x05A8	TYPE_B[3][LUT_CMP][2]	TYPE B CHN3 cmp2 lut	0x00000000
0x05AC	TYPE_B[3][LUT_CMP][3]	TYPE B CHN3 cmp3 lut	0x00000000
0x0600	TYPE_B[4][LUT][0]	TYPE B CHN4 lookup table 0	0x00000000
0x0604	TYPE_B[4][LUT][1]	TYPE B CHN4 lookup table 1	0x00000000
0x0608	TYPE_B[4][CMP][0]	TYPE B CHN4 data unit cmp0	0x00000000
0x060C	TYPE_B[4][CMP][1]	TYPE B CHN4 data unit cmp1	0x00000000
0x0610	TYPE_B[4][CMP][2]	TYPE B CHN4 data unit cmp2	0x00000000
0x0614	TYPE_B[4][CMP][3]	TYPE B CHN4 data unit cmp3	0x00000000
0x0618	TYPE_B[4][MODE]	TYPE B CHN4 mode ctrl	0x00000000
0x061C	TYPE_B[4][SW_INJECT]	TYPE B CHN4 software inject	0x00000000
0x0620	TYPE_B[4][LUT_CMP][0]	TYPE B CHN4 cmp0 lut	0x00000000
0x0624	TYPE_B[4][LUT_CMP][1]	TYPE B CHN4 cmp1 lut	0x00000000

Offset	Register Name	Description	Reset Value
0x0628	TYPE_B[4][LUT_CMP][2]	TYPE B CHN4 cmp2 lut	0x00000000
0x062C	TYPE_B[4][LUT_CMP][3]	TYPE B CHN4 cmp3 lut	0x00000000
0x0680	TYPE_B[5][LUT][0]	TYPE B CHN5 lookup table 0	0x00000000
0x0684	TYPE_B[5][LUT][1]	TYPE B CHN5 lookup table 1	0x00000000
0x0688	TYPE_B[5][CMP][0]	TYPE B CHN5 data unit cmp0	0x00000000
0x068C	TYPE_B[5][CMP][1]	TYPE B CHN5 data unit cmp1	0x00000000
0x0690	TYPE_B[5][CMP][2]	TYPE B CHN5 data unit cmp2	0x00000000
0x0694	TYPE_B[5][CMP][3]	TYPE B CHN5 data unit cmp3	0x00000000
0x0698	TYPE_B[5][MODE]	TYPE B CHN5 mode ctrl	0x00000000
0x069C	TYPE_B[5][SW_INJECT]	TYPE B CHN5 software inject	0x00000000
0x06A0	TYPE_B[5][LUT_CMP][0]	TYPE B CHN5 cmp0 lut	0x00000000
0x06A4	TYPE_B[5][LUT_CMP][1]	TYPE B CHN5 cmp1 lut	0x00000000
0x06A8	TYPE_B[5][LUT_CMP][2]	TYPE B CHN5 cmp2 lut	0x00000000
0x06AC	TYPE_B[5][LUT_CMP][3]	TYPE B CHN5 cmp3 lut	0x00000000
0x0700	TYPE_B[6][LUT][0]	TYPE B CHN6 lookup table 0	0x00000000
0x0704	TYPE_B[6][LUT][1]	TYPE B CHN6 lookup table 1	0x00000000
0x0708	TYPE_B[6][CMP][0]	TYPE B CHN6 data unit cmp0	0x00000000
0x070C	TYPE_B[6][CMP][1]	TYPE B CHN6 data unit cmp1	0x00000000
0x0710	TYPE_B[6][CMP][2]	TYPE B CHN6 data unit cmp2	0x00000000
0x0714	TYPE_B[6][CMP][3]	TYPE B CHN6 data unit cmp3	0x00000000
0x0718	TYPE_B[6][MODE]	TYPE B CHN6 mode ctrl	0x00000000
0x071C	TYPE_B[6][SW_INJECT]	TYPE B CHN6 software inject	0x00000000
0x0720	TYPE_B[6][LUT_CMP][0]	TYPE B CHN6 cmp0 lut	0x00000000
0x0724	TYPE_B[6][LUT_CMP][1]	TYPE B CHN6 cmp1 lut	0x00000000
0x0728	TYPE_B[6][LUT_CMP][2]	TYPE B CHN6 cmp2 lut	0x00000000
0x072C	TYPE_B[6][LUT_CMP][3]	TYPE B CHN6 cmp3 lut	0x00000000
0x0780	TYPE_B[7][LUT][0]	TYPE B CHN7 lookup table 0	0x00000000
0x0784	TYPE_B[7][LUT][1]	TYPE B CHN7 lookup table 1	0x00000000
0x0788	TYPE_B[7][CMP][0]	TYPE B CHN7 data unit cmp0	0x00000000
0x078C	TYPE_B[7][CMP][1]	TYPE B CHN7 data unit cmp1	0x00000000
0x0790	TYPE_B[7][CMP][2]	TYPE B CHN7 data unit cmp2	0x00000000
0x0794	TYPE_B[7][CMP][3]	TYPE B CHN7 data unit cmp3	0x00000000
0x0798	TYPE_B[7][MODE]	TYPE B CHN7 mode ctrl	0x00000000
0x079C	TYPE_B[7][SW_INJECT]	TYPE B CHN7 software inject	0x00000000
0x07A0	TYPE_B[7][LUT_CMP][0]	TYPE B CHN7 cmp0 lut	0x00000000
0x07A4	TYPE_B[7][LUT_CMP][1]	TYPE B CHN7 cmp1 lut	0x00000000
0x07A8	TYPE_B[7][LUT_CMP][2]	TYPE B CHN7 cmp2 lut	0x00000000
0x07AC	TYPE_B[7][LUT_CMP][3]	TYPE B CHN7 cmp3 lut	0x00000000

Table 184: PLB Register List

42.5 PLB Register Details

The detailed description of the PLB registers is as follows:

42.5.1 TYPE_A[LOOKUP_TABLE] (0x0 + 0x20 * n + 0x4 * m)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																LOOKUP_TABLE															
N/A																RW															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TYPE_A[LOOKUP_TABLE] [31:0]

Bits	Name	Description
15-0	LOOKUP_TABLE	using 4 bit trig_in as lookup index. software can program this register as trig_in's true table.

TYPE_A[LOOKUP_TABLE] Bits

42.5.2 TYPE_A[SW_INJECT] (0x10 + 0x20 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																												SW_INJECT			
N/A																												RW			
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

TYPE_A[SW_INJECT] [31:0]

Bits	Name	Description
3-0	SW_INJECT	software can inject value to TYPEA's output

TYPE_A[SW_INJECT] Bits

42.5.3 TYPE_B[LUT] (0x400 + 0x80 * n + 0x4 * m)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LOOKUP_TABLE																															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TYPE_B[LUT] [31:0]

Bits	Name	Description
31-0	LOOKUP_TABLE	lut0 and lut1 union as 64bit, consider each 4bit as one slice. then, total 16 slice. slice0 as bit3:0, slice1 as bit7:4...etc. using 4bit trig in as index of slice. the operate sel in data unit of type B channle is decided by which slice value choosed by trig_in

TYPE_B[LUT] Bits

42.5.4 TYPE_B[COMP] (0x408 + 0x80 * n + 0x4 * m)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP_VALUE																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TYPE_B[COMP] [31:0]

Bits	Name	Description
31-0	CMP_VALUE	cmp value, using as data unit operation

TYPE_B[COMP] Bits

42.5.5 TYPE_B[MODE] (0x418 + 0x80 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD															OPT_SEL	OUT3_SEL				OUT2_SEL				OUT1_SEL				OUT0_SEL			
N/A															RW	RW				RW				RW				RW			
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

TYPE_B[MODE] [31:0]

Bits	Name	Description
16	OPT_SEL	operation selection in data unit.
15-12	OUT3_SEL	trig out 3 output type in current channel
11-8	OUT2_SEL	trig out 2 output type in current channel

Bits	Name	Description
7-4	OUT1_SEL	trig out 1 output type in current channel
3-0	OUT0_SEL	trig out 0 output type in current channel

TYPE_B[MODE] Bits

42.5.6 TYPE_B[SW_INJECT] (0x41C + 0x80 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TYPE_B[SW_INJECT] [31:0]

Bits	Name	Description
31-0	SOFTWARE_INJECT	data unit value can be changed if program this register

TYPE_B[SW_INJECT] Bits

42.5.7 TYPE_B[LUT_CMP] (0x420 + 0x80 * n + 0x4 * m)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																LUT_CMP															
N/A																RW															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TYPE_B[LUT_CMP] [31:0]

Bits	Name	Description
15-0	LUT_CMP	using 4 bit trig_in as lookup index. software can program this register as cmp load control's true table.

TYPE_B[LUT_CMP] Bits

43 Sigma-Delta Signal Reception Unit SDM

43.1 Overview

The SDM module includes four mutually independent filter stages, capable of simultaneously decoding four channels of current measurement or rotor position data transmitted via delta-sigma ($\Delta\Sigma$) modulation for motor control applications, and providing reference timestamps. It also supports configuring mask signals to ignore unreliable portions of the input signal. This module supports upper/lower threshold comparison and zero-crossing detection. The block diagram of the module is shown in [Figure 54](#).

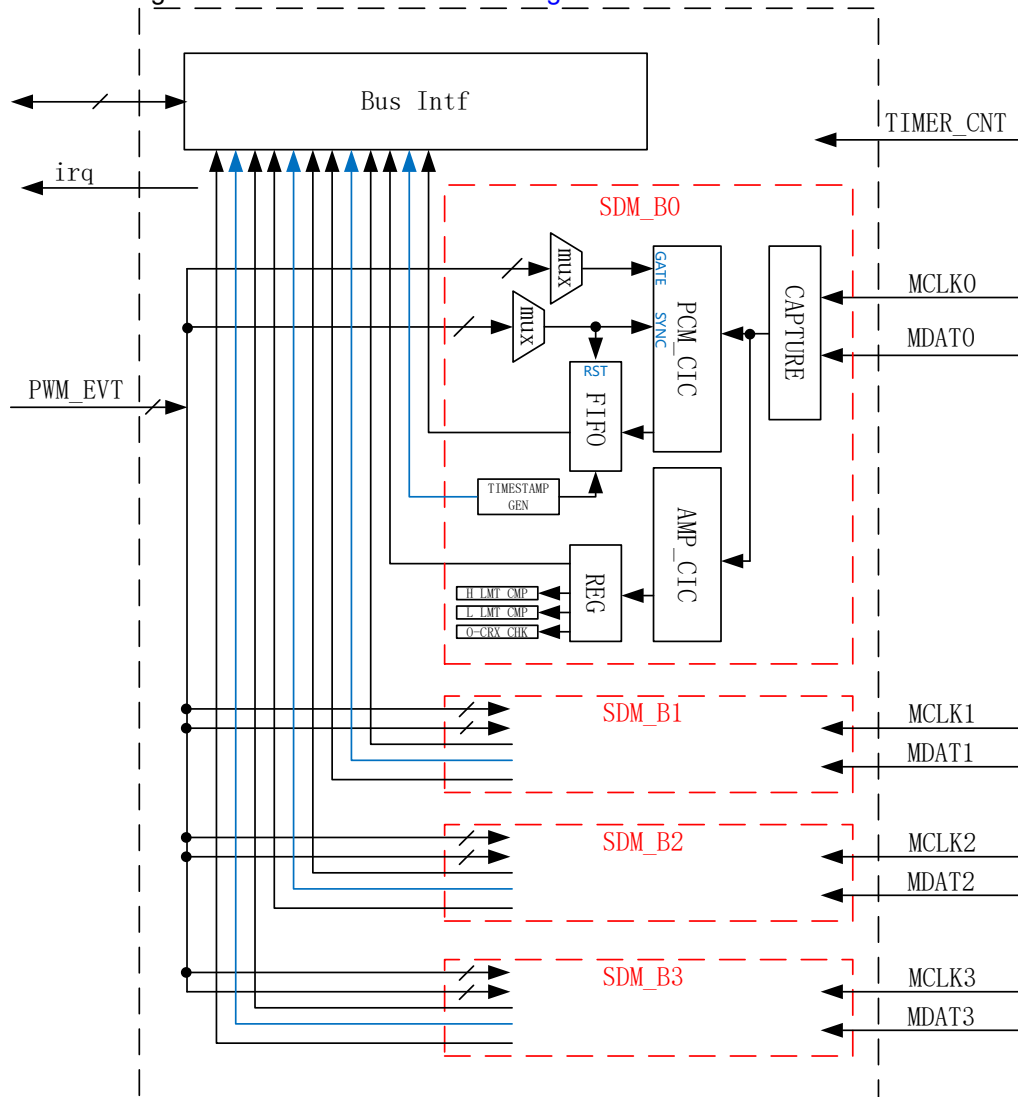


Figure 54: SDM Block Diagram

43.2 Features

The SDM includes the following features:

- Supports up to 8 external signal lines grouped into 4 channels:
 - 4 sigma-delta data input pins (MDATx, where x = 1 to 4 denotes the channel)
 - 4 sigma-delta clock input pins (MCLKx, where x = 1 to 4 denotes the channel)

- Each MCLK/MDAT pair can be independently configured into one of the following modes:
 - Mode 0: Capture on the rising edge of MCLK.
 - Mode 1: Capture on both the rising and falling edges of MCLK.
 - Mode 2: Manchester encoding mode.
 - Mode 3: Capture on the falling edge of MCLK.
 - Mode 4: Capture once every 2 rising edges of MCLK.
 - Mode 5: Capture once every 2 falling edges of MCLK.
- Each channel contains independently configurable second filter (AMP_CIC+comparator) units:
 - Four configurable filter modes (Sinc1/Sinc2/SincFast/Sinc3)
 - Simultaneous monitoring of: high threshold crossing, low threshold crossing, and exceeding a specified threshold events. These monitoring signals can be connected to the system's trigger_mux to subsequently drive corresponding control logic. Among them, high threshold and low threshold crossing events can trigger SDM interrupts.
 - The oversampling rate (OSR) for this filter unit is configurable from 1 to 32.
- Each channel contains independently configurable first filter (PCM_CIC data filter) units:
 - Four configurable filter modes (Sinc1/Sinc2/SincFast/Sinc3)
 - The oversampling rate (OSR) for this filter unit is configurable from 1 to 256.
 - This filter unit can be individually enabled.
 - The 4 filter channels can be synchronized for sampling via CTRL[CH_EN[3:0]] or by selecting the same PWM_EVT.
 - Supports optional timestamp output.
 - Mask signals can be selected via CH[x][SDCTRLE[CIC_GATE_SEL[3:0]]] to ignore unreliable PDM bit symbols. The ignore duration is software configurable.
- Data filter output can be either 16-bit or 32-bit.
- The data filter can be enabled in FIFO mode to reduce interrupt overhead. This FIFO has the following characteristics:
 - Size: 16 sample depth, 32-bit width.
 - FIFO threshold is configurable. An "data ready" interrupt can be generated when the number of data items in the buffer exceeds this threshold.
 - FIFO write can be externally synchronized: Waits for the selected PWM_EVT to arrive (also called SDSYNC synchronization event) before writing data to the FIFO, otherwise ignores previously arrived sample signals.
- A total of 16 PWM_EVT signal lines can be configured as channel-independent SDSYNC synchronization events.

43.3 Detailed Description

The SDM module has 4 filter blocks. These filter blocks are identical in performance and operate independently. Each filter block contains the following functional units:

- Input Control and Capture Unit
- First Filter (PCM_CIC Data Filter) Unit: Also called the Data Path
- Second Filter (AMP_CIC+Comparator) Unit: Contains 3 independent comparators, also called the Amplitude Path

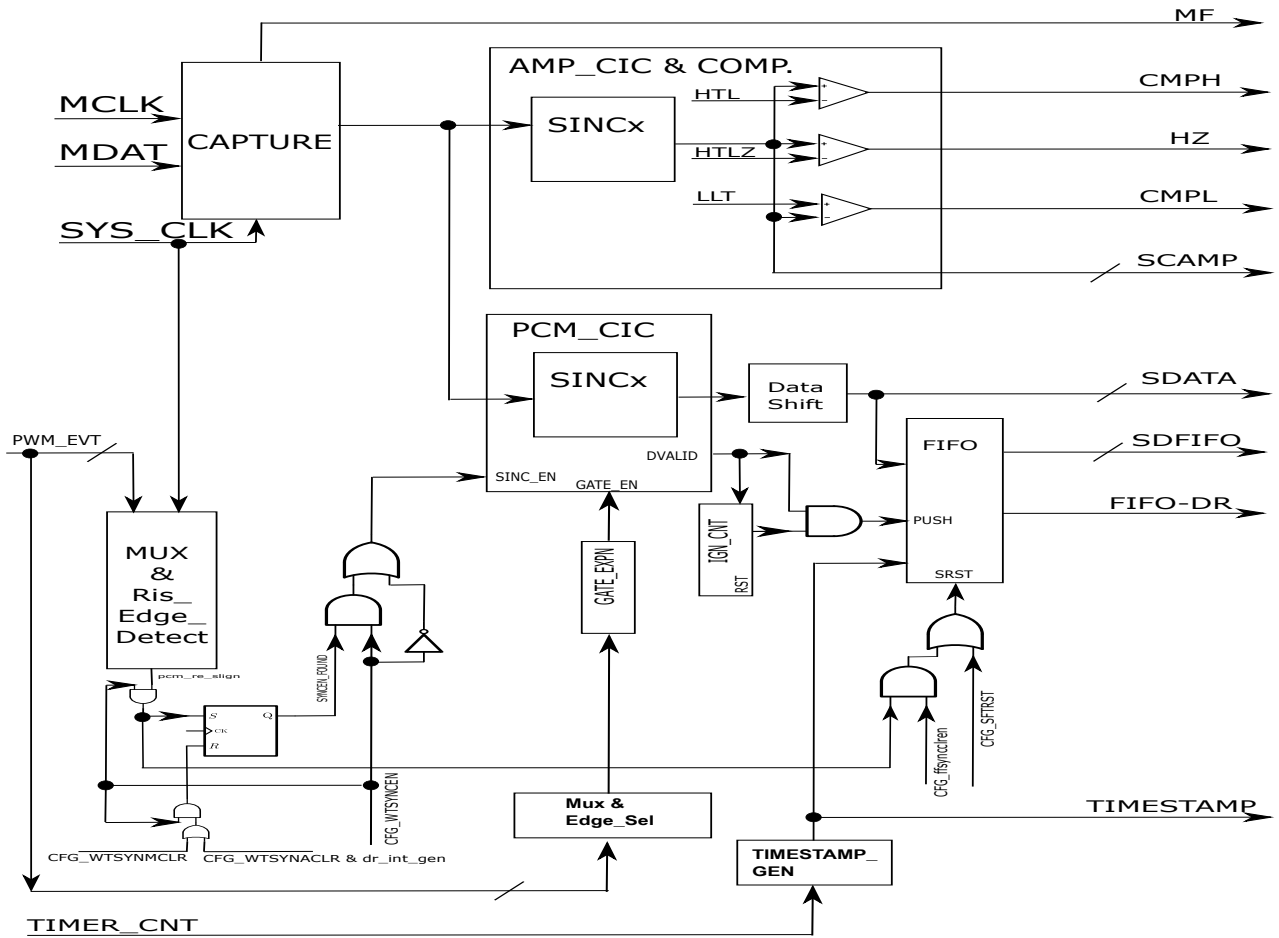


Figure 55: Filter Block Structure Diagram

A more detailed diagram of a filter block is shown in Figure 55. As seen in Figure 55, apart from sharing the input signal capture unit, the first and second filter units within each filter block are actually independent and can be configured separately.

43.3.1 Manchester Mode

In this mode, data and clock are combined into a single signal transmitted via the MDATx pin; this signal contains both clock and data, as shown in Figure 56. In this case, MCLKx is not used.

The decoding circuit in the SDM module first needs to measure the maximum period in the MDATx signal and store it in SDSTn[PERIOD_MCLK[7:0]]. The CPU should read SDSTn[PERIOD_MCLK[7:0]] several times until a reasonable period is found (its reciprocal should approximately equal the signal bit rate), then multiply this period value by 3/4 and write it into the register SDCTRLPn[MANCH_THR[6:0]].

This period count is based on the SDM module's bus frequency. For example, if the SDM module's bus frequency is 200 MHz, the maximum configurable MANCH_THR[6:0] is 0x7F. At this point, the maximum PERIOD_MCLK[7:0] can reach 0xA8 = 168, meaning the minimum acceptable signal bit rate on MDATx is 200 MHz / 168 = 1.19 Mbps.

After enabling Manchester mode reception, the first few samples might be slightly erroneous. It is recommended to ignore the initial captured values.

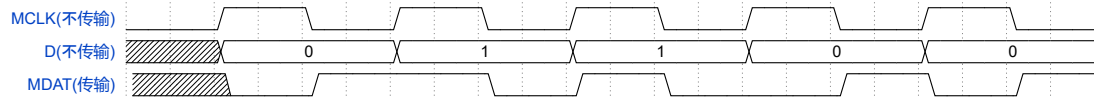


Figure 56: Manchester Mode Transmission Waveform

43.3.2 Sinc Filters

Both the first and second filter units utilize their own private Sinc filter as the core computational element. This $Sinc^N$ filter is essentially a low-pass filter for digital signals, converting a high-sample-rate digital bitstream into a digital signal with larger bit width but lower sample rate (the ratio of sampling rates is called the oversampling ratio, OSR), thereby reconstructing the analog signal at the input of the $\Delta\Sigma$ ADC.

The effective number of bits (ENOB) of the Sinc filter depends on the filter type, OSR, and the sampling rate of the $\Delta\Sigma$ ADC. With other factors fixed, a larger OSR yields higher accuracy but also greater delay. Therefore, a trade-off between sampling speed and accuracy is needed.

The order of the Sinc filter is shown in [Table 185](#).

Filter Type	Order
Sinc1	1
Sinc2	2
Sinc3	3
SincFast	2

Table 185: Sinc Filter Order

43.3.2.1 Sinc Filter Output Data Rate and Delay

The output data rate of the Sinc filter is calculated as follows, in samples per second.

$$\text{Sinc Filter Output Data Rate} = \frac{\Delta\Sigma\text{ADC Sampling Rate}}{\text{OSR}}$$

The Sinc filter delay is defined as the time from its initialization to when it produces the first correct filter output for that filter type; the unit is seconds. The calculation formula for a given filter type is:

$$\text{Sinc Filter Delay} = \frac{\text{Sinc Filter Order}}{\text{Sinc Filter Output Data Rate}}$$

Examples are as follows:

43.3.3 First Filter Unit (Data Path)

The Sinc filter in this unit supports the following types: Sinc1, Sinc2, Sinc3, and SincFast; the oversampling rate (DOSR) is configurable from 1 to 256. By default, this filter unit is disabled. It must be enabled by configuring $\text{CH}[x][\text{SDCTRLP}[\text{EN}]] = 1$.

The Sinc filter output in this unit is a 26-bit two's complement value. Correspondingly, a 0 output from the input capture unit is mapped to -1; a 1 output is mapped to 1. The upper and lower bounds of this 26-bit value vary depending on the Sinc filter type and OSR, as detailed in [Table 186](#).

For calculations regarding output data rate and filter delay, refer to [Subsection 43.3.2](#).

Filter Type = sinc3
 ADC Data Rate = 20 MHz
 OSR = 256
 Sinc Filter Output Data Rate = 20 MHz / 256 = 78.1 K samples / sec
 Sinc Filter Delay = 38.4 μ s

Filter Type = sinc2
 ADC Data Rate = 20 MHz
 OSR = 256
 Sinc Filter Output Data Rate = 20 MHz / 256 = 78.1 K samples / sec
 Sinc Filter Delay = 25.6 μ s

DOSR	Sinc1	Sinc2	Sinc3	SincFast
x	x	x^2	x^3	$2x^2$
4	[-4:4]	[-16:16]	[-64:64]	[-32:32]
8	[-8:8]	[-64:64]	[-512:512]	[-128:128]
16	[-16:16]	[-256:256]	[-4096:4096]	[-512:512]
32	[-32:32]	[-1024:1024]	[-32,768:32,768]	[-2048:2048]
64	[-64:64]	[-4096:4096]	[-262,144:262,144]	[-8192:8192]
128	[-128:128]	[-16,384:16,384]	[-2,097,152:2,097,152]	[-32,768:32,768]
256	[-256:256]	[-65,536:65,636]	[-16,777,216:16,777,216]	[-131,072:131,072]

Table 186: Sinc Filter Output Signal Upper and Lower Bounds

43.3.3.1 32-bit or 16-bit Output Control

The filter output can be forced to 32-bit or 16-bit.

32-bit Output:

- When CH[x][SDCTRLP[D32]]=1, the filter output is 32-bit. CH[x][SDCTRLE[CIC_SCL]] is ineffective in this case.

16-bit Output:

- The filter output defaults to 16-bit.
- When CH[x][SDCTRLP[D32]]=0, the filter output is 16-bit. CH[x][SDCTRLE[CIC_SCL]] must be correctly configured in this case to ensure correct output, otherwise saturation/clipping distortion will occur. When saturation/clipping occurs, CH[x][SDST[DSAT_ERR]] will be set to 1.

For example, when the filter is configured as follows,

- Filter Type = Sinc3
- OSR = 128
- CH[x][SDCTRLP[D32]]=0

The filter internally outputs 26 bits, ranging in $[-33,554,432:33,554,432]$, but 16 bits can only represent $[-32,768:32,767]$. Therefore, the 26-bit result needs to be right-shifted by at least 10 bits, i.e., configure $CH[x][SDCTRL[CIC\_S]]$ to ensure the result can be represented in 16 bits.

43.3.3.2 Data FIFO

The data path includes a 16-sample deep, 32-bit wide first-in-first-out buffer (FIFO). This FIFO has a configurable threshold ($CH[x][SDFIFOCTRL[THRSH]]$). An interrupt is generated when the number of samples in the FIFO exceeds this threshold. By default, this FIFO is disabled. It must be enabled by configuring $CH[x][SDCTRLP[DR\_OPT]]$. The lower 5 bits of the sample count in the FIFO are displayed in real-time in $CH[x][SDST[FILL]]$.

Configuring the FIFO to Generate a Data Ready Interrupt After a Certain Number of Samples

- Enable the FIFO ($CH[x][SDCTRLP[DR\_OPT]] = 1$)
- Enable the FIFO data ready interrupt ($CH[x][SDCTRLP[DRIE]] = 1$)
- Configure $CH[x][SDFIFOCTRL[THRSH]]$ to a number between 0 and 15.

When $CH[x][SDST[FILL]] > CH[x][SDFIFOCTRL[THRSH]]$, $CH[x][SDST[FIFO\_DR]] = 1$, and an interrupt is generated. At the end of the interrupt service routine, write 1 to $CH[x][SDST[FIFO\_DR]]$ to clear this status bit.

Synchronized Sampling Implementation

Data path sampling can be configured to start filter calculation only after the selected PWM_EVT (also called the **synchronization event**) arrives. This function is disabled by default. It must be enabled by configuring $CH[x][SDCTRLP[WTSYNEN]] = 1$.

When Synchronized Sampling is Disabled

The filter starts calculation immediately upon enablement; each calculated number is written into the FIFO, until the FIFO is full or an interrupt is triggered when $CH[x][SDST[FILL]] > CH[x][SDFIFOCTRL[THRSH]]$.

When Synchronized Sampling is Enabled

Only when the synchronization event arrives does $CH[x][SDST[WTSYNFLG]]$ become 1; afterwards, filter calculation begins. The results calculated by the filter are then written into the FIFO. Writing continues until the FIFO is full, or until $CH[x][SDCTRLP[WTSYNMCLR]] = 1$, or until $CH[x][SDST[FILL]] > CH[x][SDFIFOCTRL[THRSH]]$ generates an interrupt request (when $CH[x][SDCTRLP[WTSYNACLR]] = 1$). $CH[x][SDST[WTSYNFLG]]$ can be cleared manually or automatically. When $CH[x][SDST[WTSYNFLG]] = 0$, writing new data to the FIFO stops until a new synchronization event arrives.

Manual Clear of WTSYNFLG

When $CH[x][SDCTRLP[WTSYNACLR]] = 0$, configure $CH[x][SDCTRLP[WTSYNMCLR]] = 1$ to clear WTSYNFLG.

Automatic Clear of WTSYNFLG

When $CH[x][SDCTRLP[WTSYNACLR]]=1$, if a data ready interrupt is generated, the hardware automatically clears WTSYNFLG simultaneously with the interrupt generation.

FIFO Content Clear

FIFO content can be cleared in the following ways:

- When the FIFO is disabled, i.e., $CH[x][SDCTRLP[DR\_OPT]]=0$.
- When the data path is disabled, i.e., $CH[x][SDCTRLP[EN]]=0$.
- When $CH[x][SDCTRLP[WTSYNCEN]]=1$ and $CH[x][SDCTRLP[FFSYNCCCLREN]]=1$, the arrival of a synchronization event automatically clears the FIFO content.

43.3.3.3 Synchronization Event

The data path filter can be synchronously driven by the rising edge of a PWM_EVT signal. PWM_EVT can originate from other modules in the system. This event resets the CIC filter's oversampling counter. Configuring $CH[x][SDCTRLP[WTSYNCEN]]=1$ enables this synchronization function. The specific event to use is selected by $CH[x][SDCTRLP[SYNCSEL]]$.

Due to the characteristics of the Sinc filter, the initial few output sample values are likely incorrect under the following circumstances:

- When the Sinc filter is first configured and enabled.
- When the Sinc filter is disabled and then re-enabled.
- When synchronization is enabled and a synchronization event arrives.

The number of these incorrect initial samples is shown in [Table 187](#):

Filter Type	Number of Initially Incorrect Samples
Sinc1	0
Sinc2	1
Sinc3	2
SincFast	2

Table 187: Relationship Between Number of Initially Incorrect Samples and Filter Type

Note: The amplitude path has the same issue. Therefore, to avoid erroneous operation, it is necessary to wait for approximately 5 output samples after enabling the amplitude path before enabling the related interrupts.

43.3.3.4 Mask Control

When the PWM signal of a motor turns on and off, it causes significant current disturbance. ADC samples during this period contain considerable noise. Excluding these noisy portions from the PDM received signal can make the SDM's PCM output data more reliable. This can be achieved through mask control.

The implementation of this function is as follows:

1. Select the signal synchronized with the PWM turn-on/turn-off as the mask trigger signal, configured in CH[x][SDCTRLE[CIC_GATE_SEL[3:0]]].
2. Configure CH[x][SDCTRLE[CIC_GATE_POL]] to specify whether the trigger signal's rising or falling edge generates the trigger edge event.
3. When the trigger edge of the trigger signal arrives, pause the CIC input according to the specification in CH[x][SDCTRLE[CIC_GATE_TYPE]].
4. After pausing the CIC input for the specified number of PDM bits, stop the pause operation and continue CIC filtering on subsequent PDM bits.

43.3.3.5 Timestamp Output

A corresponding timestamp can be output along with the PCM signal. This timestamp marks the approximate time corresponding to the PCM data, making it more suitable for applications with strict data timing requirements.

There are two mechanisms for timestamp generation.

- When CH[x][SDCTRLE[TIMESTAMP_TYPE]]=0: The output timestamp is the moment the PCM data is calculated.
- When CH[x][SDCTRLE[TIMESTAMP_TYPE]]=1: Given the CIC's downsampling rate and filter order, the CIC filter is equivalent to a fixed-length FIR filter. The output timestamp is the calculation moment of the PCM data minus the equivalent delay of this filter.

Due to mask control, when a mask event occurs, CIC calculation is paused. This pause is ignored by the timestamp calculation logic, so the output timestamp has a certain degree of ambiguity.

43.3.4 Second Filter Unit (Amplitude Path)

Many control systems require stopping the PWM when current or voltage exceeds a specified range. The second filter unit is used for monitoring and requires a fast enough response to stop the PWM and prevent system damage.

Note

The second filter unit has no synchronization event control, no mask control, and no timestamp output.

The amplitude path filter is also a Sinc filter, supporting Sinc1, Sinc2, Sinc3, and SincFast types, but its OSR (also called COSR) only supports values from 1 to 32. It is enabled by configuring CH[x][SCCTRL[EN]]. This Sinc filter outputs a 16-bit unsigned value. Correspondingly, a 0 output from the input capture unit is mapped to 0; a 1 output is mapped to 1.

The upper and lower bounds of this 16-bit value vary depending on the Sinc filter type and OSR, as detailed in [Table 188](#).

For calculations regarding output data rate and filter delay, refer to [Subsection 43.3.2](#).

The amplitude value output from the amplitude path Sinc filter can be read directly and in real-time from CH[x][SCAMP], with no FIFO buffering. This output is also connected to multiple comparison branches.

All comparator outputs are routed outside this module to trigger subsequent operations.

COSR	Sinc1	Sinc2	Sinc3	SincFast
x	x	x^2	x^3	$2x^2$
4	[0:4]	[0:16]	[0:64]	[0:32]
8	[0:8]	[0:64]	[0:512]	[0:128]
16	[0:16]	[0:256]	[0:4096]	[0:512]
32	[0:32]	[0:1024]	[0:32,768]	[0:2048]

Table 188: Amplitude Path Sinc Filter Output Signal Upper and Lower Bounds

43.3.4.1 Upper Threshold Comparator

The threshold is configured in the CH[x][SCHTL] register. When the amplitude is greater than or equal to this threshold, CH[x][SCST[CMPH]] is set to 1.

If CH[x][SCCTRL[HL_IE]] is configured to 1, an interrupt can be triggered. Write 1 to CH[x][SCST[CMPH]] to clear it to 0.

43.3.4.2 Zero-Crossing Comparator

The threshold is configured in the CH[x][SCHTLZ] register. When the amplitude is greater than or equal to this threshold, CH[x][SCST[HZ]] is set to 1. This event cannot be configured to trigger an interrupt. Write 1 to CH[x][SCST[HZ]] to clear it to 0.

43.3.4.3 Lower Threshold Comparator

The threshold is configured in the CH[x][SCLLT] register. When the amplitude is less than or equal to this threshold, CH[x][SCST[CMPL]] is set to 1.

If CH[x][SCCTRL[LL_IE]] is configured to 1, an interrupt can be triggered. Write 1 to CH[x][SCST[CMPL]] to clear it to 0.

43.4 SDM Registers

43.4.1 Register Description

The SDM register list is as follows:

SDM0 base address: 0xF0450000

Offset	Register Name	Description	Reset Value
0x0000	CTRL	SDM control register	0x00000000
0x0004	INT_EN	Interrupt enable register.	0x00000000
0x0008	STATUS	Status Registers	0x00000000
0x0010	CH[0][SDFIFOCTRL]	Data FIFO Path Control Register	0x00000000
0x0014	CH[0][SDCTRLP]	Data Path Control Primary Register	0x00000000

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3
SDM

43 SIGMA-DELTA SIGNAL RECEPTION UNIT

Offset	Register Name	Description	Reset Value
0x0018	CH[0][SDCTRLE]	Data Path Control Extra Register	0x00000000
0x001C	CH[0][SDST]	Data Path Status	0x00000000
0x0020	CH[0][SDATA]	Data	0x00000000
0x0024	CH[0][SDFIFO]	FIFO Data	0x00000000
0x0028	CH[0][SCAMP]	instant Amplitude Results	0x00000000
0x002C	CH[0][SCHTL]	Amplitude Threshold for High Limit	0x00000000
0x0030	CH[0][SCHTLZ]	Amplitude Threshold for zero crossing	0x00000000
0x0034	CH[0][SCLLT]	Amplitude Threshold for low limit	0x00000000
0x0038	CH[0][SCCTRL]	Amplitude Path Control	0x00000000
0x003C	CH[0][SCST]	Amplitude Path Status	0x00000000
0x0050	CH[1][SDFIFOCTRL]	Data FIFO Path Control Register	0x00000000
0x0054	CH[1][SDCTRLP]	Data Path Control Primary Register	0x00000000
0x0058	CH[1][SDCTRLE]	Data Path Control Extra Register	0x00000000
0x005C	CH[1][SDST]	Data Path Status	0x00000000
0x0060	CH[1][SDATA]	Data	0x00000000
0x0064	CH[1][SDFIFO]	FIFO Data	0x00000000
0x0068	CH[1][SCAMP]	instant Amplitude Results	0x00000000
0x006C	CH[1][SCHTL]	Amplitude Threshold for High Limit	0x00000000
0x0070	CH[1][SCHTLZ]	Amplitude Threshold for zero crossing	0x00000000
0x0074	CH[1][SCLLT]	Amplitude Threshold for low limit	0x00000000
0x0078	CH[1][SCCTRL]	Amplitude Path Control	0x00000000
0x007C	CH[1][SCST]	Amplitude Path Status	0x00000000
0x0090	CH[2][SDFIFOCTRL]	Data FIFO Path Control Register	0x00000000
0x0094	CH[2][SDCTRLP]	Data Path Control Primary Register	0x00000000
0x0098	CH[2][SDCTRLE]	Data Path Control Extra Register	0x00000000
0x009C	CH[2][SDST]	Data Path Status	0x00000000
0x00A0	CH[2][SDATA]	Data	0x00000000
0x00A4	CH[2][SDFIFO]	FIFO Data	0x00000000

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3
SDM

43 SIGMA-DELTA SIGNAL RECEPTION UNIT

Offset	Register Name	Description	Reset Value
0x00A8	CH[2][SCAMP]	instant Amplitude Results	0x00000000
0x00AC	CH[2][SCHTL]	Amplitude Threshold for High Limit	0x00000000
0x00B0	CH[2][SCHTLZ]	Amplitude Threshold for zero crossing	0x00000000
0x00B4	CH[2][SCLLT]	Amplitude Threshold for low limit	0x00000000
0x00B8	CH[2][SCCTRL]	Amplitude Path Control	0x00000000
0x00BC	CH[2][SCST]	Amplitude Path Status	0x00000000
0x00D0	CH[3][SDFIFOCTRL]	Data FIFO Path Control Register	0x00000000
0x00D4	CH[3][SDCTRLP]	Data Path Control Primary Register	0x00000000
0x00D8	CH[3][SDCTRLE]	Data Path Control Extra Register	0x00000000
0x00DC	CH[3][SDST]	Data Path Status	0x00000000
0x00E0	CH[3][SDATA]	Data	0x00000000
0x00E4	CH[3][SDFIFO]	FIFO Data	0x00000000
0x00E8	CH[3][SCAMP]	instant Amplitude Results	0x00000000
0x00EC	CH[3][SCHTL]	Amplitude Threshold for High Limit	0x00000000
0x00F0	CH[3][SCHTLZ]	Amplitude Threshold for zero crossing	0x00000000
0x00F4	CH[3][SCLLT]	Amplitude Threshold for low limit	0x00000000
0x00F8	CH[3][SCCTRL]	Amplitude Path Control	0x00000000
0x00FC	CH[3][SCST]	Amplitude Path Status	0x00000000

Table 189: SDM Register List

43.4.2 Register Details

The detailed description of SDM registers is as follows:

43.4.3 CTRL (0x0)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
SFTRST	RSVD					CHMD													SYNC_MCLK				SYNC_MDAT				CH_LEN				IE	RSVD
RW	N/A					RW													RW				RW				RW				RW	N/A
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

CTRL [31:0]

Bits	Name	Description
31	SFTRST	software reset the module if asserted to be 1' b1.
25-14	CHMD	Channel Rcv mode Bits[2:0] for Ch0. Bits[5:3] for Ch1 Bits[8:6] for Ch2 Bits[11:9] for Ch3 3'b000: Capture at posedge of MCLK 3'b001: Capture at both posedge and negedge of MCLK 3'b010: Manchester Mode 3'b011: Capture at negedge of MCLK 3'b100: Capture at every other posedge of MCLK 3'b101: Capture at every other negedge of MCLK Others: Undefined
13-10	SYNC_MCLK	Asserted to double sync the mclk input pin before its usage inside the module
9-6	SYNC_MDAT	Asserted to double sync the mdat input pin before its usage inside the module
5-2	CH_EN	Channel Enable
1	IE	Interrupt Enable

CTRL Bits

43.4.4 INT_EN (0x4)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RSVD																								CH3DRY	CH2DRY	CH1DRY	CH0DRY	CH3ERR	CH2ERR	CH1ERR	CH0ERR	
N/A																								RW	RW	RW	RW	RW	RW	RW	RW	RW
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

INT_EN [31:0]

Bits	Name	Description
7	CH3DRY	Ch3 Data Ready interrupt enable.
6	CH2DRY	Ch2 Data Ready interrupt enable
5	CH1DRY	Ch1 Data Ready interrupt enable
4	CH0DRY	Ch0 Data Ready interrupt enable
3	CH3ERR	Ch3 Error interrupt enable.

Bits	Name	Description
2	CH2ERR	Ch2 Error interrupt enable
1	CH1ERR	Ch1 Error interrupt enable
0	CH0ERR	Ch0 Error interrupt enable

INT_EN Bits

43.4.5 STATUS (0x8)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								CH3DRY	CH2DRY	CH1DRY	CH0DRY	CH3ERR	CH2ERR	CH1ERR	CH0ERR
N/A																								RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

STATUS [31:0]

Bits	Name	Description
7	CH3DRY	Ch3 Data Ready. De-assert this bit by reading the data (or data fifo) registers.
6	CH2DRY	Ch2 Data Ready
5	CH1DRY	Ch1 Data Ready
4	CH0DRY	Ch0 Data Ready
3	CH3ERR	Ch3 Error. ORed together by channel related error signals and corresponding error interrupt enable signals. De-assert this bit by write-1-clear the corresponding error status bits in the channel status registers.
2	CH2ERR	Ch2 Error
1	CH1ERR	Ch1 Error
0	CH0ERR	Ch0 Error

STATUS Bits

43.4.6 CH[SDFIFOCTRL] (0x10 + 0x40 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								GATE_SAMPLES								RSVD								THRSH				RSVD			
N/A								RW								N/A								RW				N/A			

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	0	0	0	0	0	x	x	x	x

CH[SDFIFOCTRL] [31:0]

Bits	Name	Description
23-16	GATE_SAMPLES	The number-1-3 of input PDM bit samples to be gated when CIC_GATE_EN=1. Max 255. So the minimum gated samples is 4 samples when GATE_SAMPLES=0.
8-4	THRSH	FIFO threshold (0,...,16) (fillings > threshold, then gen int)

CH[SDFIFOCTRL] Bits

43.4.7 CH[SDCTRLP] (0x14 + 0x40 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MANCH_THR							WDOG_THR							RSVD	DFFOVIE	DSATIE	DRIE	SYNCSEL							FFSYNCLREN	WTSYNACLR	WTSYNMCLR	WTSYNCEN	D32	DR_OPT	EN
RW							RW							N/A	RW	RW	RW	RW							RW	RW	RW	RW	RW	RW	RW
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CH[SDCTRLP] [31:0]

Bits	Name	Description
31-25	MANCH_THR	Manchester Decoding threshold. 3/4 of PERIOD_MCLK[7:0]
24-17	WDOG_THR	Watch dog threshold for channel failure of CLK halting
15	DFFOVIE	Ch Data FIFO overflow interrupt enable
14	DSATIE	Ch CIC Data Saturation Interrupt Enable
13	DRIE	Ch Data Ready Interrupt Enable
12-7	SYNCSEL	Select the PWM SYNC Source
6	FFSYNCLREN	Auto clear FIFO when a new SDSYNC event is found. Only valid when WTSYNCEN=1
5	WTSYNACLR	1: Asserted to Auto clear WTSYNFLG when the SDFFINT is gen 0: WTSYNFLG should be cleared manually by WTSYNMCLR
4	WTSYNMCLR	1: Manually clear WTSYNFLG. Auto-clear.
3	WTSYNCEN	1: Start to store data only after PWM SYNC event 0: Start to store data whenever enabled
2	D32	1:32 bit data 0:16 bit data
1	DR_OPT	1: Use Data FIFO Ready as data ready when fifo fillings are greater than the threshold 0: Use Data Reg Ready as data ready

Bits	Name	Description
0	EN	Data Path Enable

CH[SDCTRLP] Bits

43.4.8 CH[SDCTRLE] (0x18 + 0x40 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CIC_GATE_TYPE	CIC_GATE_POL	CIC_GATE_SEL				CIC_GATE_EN	RSVD		TIMESTAMP_TYPE	DFIFO_S_T	DATA_S_T	RSVD		SGD_ORDER	PWMSYNC	RSVD		CIC_SCL				CIC_DEC_RATIO								IGN_INI_SAMPLES	
RW	RW	RW				RW	N/A		RW	RW	RW	N/A		RW	RW	N/A		RW				RW								RW	
0	0	0	0	0	0	0	x	x	0	0	0	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CH[SDCTRLE] [31:0]

Bits	Name	Description
31	CIC_GATE_TYPE	1: the gate cycle is determined by SDFIFOCTRLn[GATE_SAMPLES]. 0: the gate cycle is determined by the CIC decimation counter, and the minimal gated off PDM bits are determined by SDFIFOCTRLn[GATE_SAMPLES], and at the same time, to keep alignment with normal PCM sampling time.
30	CIC_GATE_POL	1: When mask signal is 1, pause the CIC stage at the rising edge of mask signal. 0: When mask signal is 0, pause the CIC stage at the falling edge of mask signal.
29-26	CIC_GATE_SEL	Select the mask signal for CIC gate signal.
25	CIC_GATE_EN	1: the CIC stage can be paused by the mask input. 0: the CIC stage won't be paused by the mask input.
22	TIMESTAMP_TYPE	1: Use the time (when the data is calculated out) - delta_time_of_filter_span as the timestamp. 0: Use the time when the data is calculated out.
21	DFIFO_S_T	1: the output of SDFIFO is data and timestamp interleaved. First is data. 0: the output of SDFIFO is data only
20	DATA_S_T	"1: the read output of SData is data and timestamp interleaved. First is data. 0: the read output of SData is data only"

Bits	Name	Description
18-17	SGD_ORDR	CIC order 0: SYNC1 1: SYNC2 2: SYNC3 3: FAST_SYNC
16	PWMSYNC	Asserted to double sync the PWM trigger signal
14-11	CIC_SCL	CIC shift control
10-3	CIC_DEC_RATIO	CIC decimation ratio. 0 means div-by-256
2-0	IGN_INI_SAMPLE S	NotZero: Don't store the first samples that are not accurate Zero: Store all samples

CH[SDCTRLE] Bits

43.4.9 CH[SDST] (0x1C + 0x40 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	PERIOD_MCLK								RSVD								SDATA_D0_T1	SDFIFO_D0_T1	RSVD		FIFO_DR	RSVD	DOV_ERR	DSAT_ERR	WTSYNFLG	FILL					
N/A	RO								N/A								RO	RO	N/A		W1C	N/A	W1C	W1C	RO	RO					
0	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x	x	0	0	x	x	0	x	0	0	0	0	0	0	0	0

CH[SDST] [31:0]

Bits	Name	Description
30-23	PERIOD_MCLK	maxim of mclk spacing in cycles, using edges of mclk signal. In manchester coding mode, it is just the period of MCLK. In other modes, it is almost the half period.
13	SDATA_D0_T1	1: current value in SDATA is data 0: current value in SDATA is timestamp
12	SDFIFO_D0_T1	1: first value readout from FIFO is data 0: first value readout from FIFO is timestamp
9	FIFO_DR	FIFO data ready
7	DOV_ERR	Data FIFO Overflow Error. Error flag.
6	DSAT_ERR	CIC out Data saturation err. Error flag.
5	WTSYNFLG	Wait-for-sync event found
4-0	FILL	Data FIFO Fillings

CH[SDST] Bits

43.4.10 CH[SDATA] (0x20 + 0x40 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VAL																															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CH[SDATA] [31:0]

Bits	Name	Description
31-0	VAL	Data

CH[SDATA] Bits

43.4.11 CH[SDFIFO] (0x24 + 0x40 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VAL																															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CH[SDFIFO] [31:0]

Bits	Name	Description
31-0	VAL	FIFO Data

CH[SDFIFO] Bits

43.4.12 CH[SCAMP] (0x28 + 0x40 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																VAL															
N/A																RO															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CH[SCAMP] [31:0]

Bits	Name	Description
15-0	VAL	instant Amplitude Results

CH[SCAMP] Bits

43.4.13 CH[SCHTL] (0x2C + 0x40 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																VAL															
N/A																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CH[SCHTL] [31:0]

Bits	Name	Description
15-0	VAL	Amplitude Threshold for High Limit

CH[SCHTL] Bits

43.4.14 CH[SCHTLZ] (0x30 + 0x40 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																VAL															
N/A																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CH[SCHTLZ] [31:0]

Bits	Name	Description
15-0	VAL	Amplitude Threshold for zero crossing

CH[SCHTLZ] Bits

43.4.15 CH[SCLLT] (0x34 + 0x40 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																VAL															
N/A																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CH[SCLLT] [31:0]

Bits	Name	Description
15-0	VAL	Amplitude Threshold for low limit

CH[SCLLT] Bits

43.4.16 CH[SCCTRL] (0x38 + 0x40 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								HZ_EN	MF_IE	HL_IE	LL_IE	SGD_ORDR		RSVD								CIC_DEC_RATIO					IGN_INI_SAMPLES			EN	
N/A								RW	RW	RW	RW	RW		N/A								RW					RW			RW	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0

CH[SCCTRL] [31:0]

Bits	Name	Description
23	HZ_EN	Zero Crossing Enable
22	MF_IE	Module failure Interrupt enable
21	HL_IE	HLT Interrupt Enable
20	LL_IE	LLT interrupt Enable
19-18	SGD_ORDR	CIC order 0: SYNC1 1: SYNC2 2: SYNC3 3: FAST_SYNC
8-4	CIC_DEC_RATIO	CIC decimation ratio. 0 means div-by-32
3-1	IGN_INI_SAMPLE S	NotZero: Ignore the first samples that are not accurate Zero: Use all samples
0	EN	Amplitude Path Enable

CH[SCCTRL] Bits

43.4.17 CH[SCST] (0x3C + 0x40 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																												HZ	MF	CMPH	CMPH
N/A																												W1C	W1C	W1C	W1C
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

CH[SCST] [31:0]

Bits	Name	Description
3	HZ	Amplitude rising above HZ event found.
2	MF	power modulator Failure found. MCLK not found. Error flag.

Bits	Name	Description
1	CMPH	HLT out of range. Error flag.
0	CMPL	LLT out of range. Error flag.

CH[SCST] Bits

43.5 SDM Configuration and Usage Guide

43.5.1 Interrupt Handling

SDM has 2 types of interrupts. 1) Data Ready Interrupt: In this case, STATUS[CHxDRY]=1. If CH[x][SDCTRLP[DR_OPT]] (data transferred via FIFO), the interrupt service routine must read a certain number of samples from CH[x][SDFIFO] before the interrupt source can be cleared. If CH[x][SDCTRLP[DR_OPT]]=0 (no FIFO transfer), reading CH[x][SDATA] itself will clear the interrupt source, or writing 1 to CH[x][SDST[AF]] can also clear the interrupt source. 2) Error Interrupt: In this case, STATUS[CHxERR]=1. CH[x][SDST[DOV_ERR]] error interrupt: Write 1 to CH[x][SDST[DOV_ERR]] to clear the interrupt source. CH[x][SDST[DSAT_ERR]] error interrupt: Write 1 to CH[x][SDST[DSAT_ERR]] to clear the interrupt source. CH[x][SCST[MF]] error interrupt: Write 1 to CH[x][SCST[MF]] to clear the interrupt source. CH[x][SCST[CMPH]] error interrupt: Write 1 to CH[x][SCST[CMPH]] to clear the interrupt source. CH[x][SCST[CMPL]] error interrupt: Write 1 to CH[x][SCST[CMPL]] to clear the interrupt source. Note: CH[x][SCST[HZ]] zero-crossing detection does not generate an interrupt, it only generates a signal for the trigger_mux.

44 Timer Overview

This chapter introduces the timers available on this product. The timers on this product include general-purpose timers, watchdog timers, and real-time clocks.

44.1 General-Purpose Timers GPTMR, PTMR

This product supports 6 general-purpose timers. Each general-purpose timer supports 4 channels, with each channel featuring a 32-bit counter, a reload register, and an input capture/output compare channel. It supports general-purpose timing, input capture, output compare, PWM generation, as well as generating interrupts and DMA requests.

- 4 timers located in the system power domain are called General-Purpose Timers GPTMR0~3.
- 1 timer located in the CONN subsystem of the system power domain is called the Network Timer NTMR0.
- 1 timer located in the power management domain is called the Power Management Domain Timer PTMR.

On this product, the 4 input capture/output compare channels of PTMR are connected to the power management domain IOs (GPIO port Y). PTMR can maintain operation when the system power domain loses power. PTMR interrupts can wake the system from low-power modes. PTMR does not support generating DMA requests.

44.2 Watchdog Timers WDG, PWDG

This product supports 3 watchdog timers. The watchdog timers can prevent the system from locking up due to software or hardware errors. The watchdog timers support generating an interrupt or a system reset when the software "feeds the dog" times out.

Among them, 2 located in the system power domain are called WDG0~1. The clock for WDGx is CLK_TOP_AHB, and the external clock source for the watchdog counter is CLK_32K.

One is located in the power management domain, called PWDG. PWDG can maintain operation when the system power domain loses power. The clock for PWDG is CLK_24M, and the external clock source for the watchdog counter is CLK_32K. PWDG supports outputting a reset signal from the power management domain IO.

45 Timer TMR

This chapter describes the main functions and features of timer TMR.

45.1 Feature Summary

This chapter describes the main features of timer TMR:

- timer consists of 4 channels
- Single channel is configured with a 32-bit resolution counter and supports up mode
- Counters for each channel support synchronization
- Single channel supports configurable 32-bit overload registers
- channel can be used for both input capture and output comparison
- Supports 2 output comparators on a single channel
- supports capturing rising and/or falling edges, and supports measuring PWM cycles and duty cycles
- supports single mode, burst mode, and loop mode
- supports external signal counting
- supports simple encoder mode
- supports generation of DMA and interrupt requests

timer is shown as [Figure 57](#) .

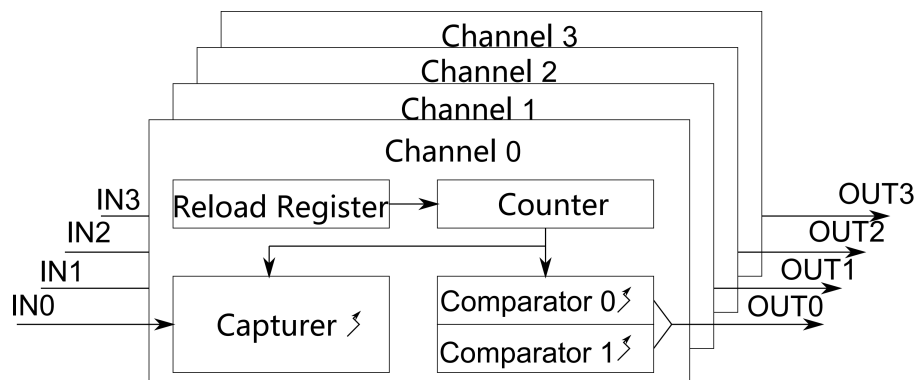


Figure 57: Timer Block Diagram

45.2 Function Description

This chapter describes the functions of each sub-module of the timer.

45.2.1 Timer Time Base

The time benchmark module for each channel of the timer is used to determine the time and cycle required for the timer to run.

contains the following parts:

- 32-bit counter
- 32-bit overloaded register

timer supports counting up.

The user can set the CHxCR [CEN] bit to enable the counter. After enabling, the counter always starts

counting from 0. When the value of the timer reaches the overload value, the overload flag is set to 1, and the counter returns to 0 and starts counting.

The user can reset the counter at any time using the CHxCR [CNTRST] bit. Setting the CHxCR[CNTRST] bit 1 returns the counter to 0. You need to set the CHxCR[CNTRST] bit to 0 again before the counter starts counting again. When the counter is reset using the CHxCR[CNTRST] bit, it also resets all input capture value registers and resets the output of the output comparison to the starting level defined by the CHxCR[COMPINIT] bit.

The counter register is a read-only register, and the user can update the value of the counter CHxCNT to the expected value by using the following steps:

- Writes the expected value to register CHxCNTUPTVAL
- Set the CHxCR [CNTUPT] bit to 1 and load the value of CHxCNTNEW into the counter CHxCNT

Users can also use the timer's synchronized trigger input (SYNCl) to synchronize the timing when the counter starts counting. The timer supports synchronization mechanisms such as software synchronization and hardware synchronization:

- After setting the CHxCR [SWSYNClEN] bit to 1, software synchronization can be triggered by writing the GCR[SWSYNCTx] bit. Each channel has a separate software trigger bit. The user can set the software trigger position of multiple channels to 1 at the same time, so that the timers of these channels can start timing synchronously.
- The hardware synchronization signal comes from outside the timer, namely SYNCl. You can select whether synchronization occurs on the rising edge or the falling edge of SYNCl by setting the CHxCR [SYNClREN] or CHxCR [SYNClFEN] bit to 1.
- In addition, for channels with channel sequence numbers greater than 0, the user can choose to set the CHxCR [SYNClFLW] bit to 1, so that the channel counter will be synchronized with the counter of the channel before it. For example, the counter on channel 1 will be synchronized with the counter on channel 0 at the same time.

If the counter is reset to 0 through a synchronous trigger input, the overload flag bit RLDF will also be set to 1.

45.2.2 output comparison

universal timer supports output comparison.

The configuration process using the channel output comparison function is as follows:

- configures the starting state of the output through the CHxCR [COMPINIT] bit. When this bit is 0, it means that when the counter starts counting, the starting state of the output is logic 0; otherwise, when the counter starts counting, the starting state of the output is logic 1. In addition, when the counter is overloaded (CNT == RLD), the output is also reset to the starting state.
- configures two comparators for the channel, CMP0 and CMP1. The channel output flips whenever the counter reaches any comparator value.

If CMP0 and CMP1 are configured to be equal, the output will not change when the counter reaches the comparator value.

- Turn on the output comparison function by setting the CHxCR [CMPEN] bit to 1
- Enable the counter by setting the CHxCR [CEN] bit to 1

If users need to output edge-aligned PWM, they can set the duty cycle of the PWM through CMP0 and set the CMP1 value equal to the overload register value (CMP1 = RLD).

Users can set $0 < CMP0 < CMP1 < RLD$ if they need to output center-aligned PWM. In this way, the PWM cycle is the RLD value and the duty cycle is (CMP1-CMP0).

Output comparison supports fast output of logic 0 and logic 1, that is, when CMP0 is configured to 32'hFFFFFF, the channel output is always logic 0. When CMP1 is configured to 32'hFFFFFF and CMP0 is not all 1s, the channel output is always logic 1.

Note: The initial values of CMP0 and CMP1 are both 32'hFFFFFF0. Users need to configure two values when using the compare output function.

45.2.3 Input Capture

universal timer supports input capture.

The user can configure the CHxCR[CAPMODE] bit to select different capture modes:

- 3'b000, turn off input capture
- 3'b001, captures the rising edge, saves the current counter value to the CAPPOS register whenever the input changes from logic 0 to logic 1, and sets the flag bit CHxCAPF to 1
- 3'b010, captures the falling edge, saves the current counter value to the CAPNEG register whenever the input changes from logic 1 to logic 0, and sets the flag bit CHxCAPF to 1
- 3'b011, captures rising and falling edges, saves the current counter value to the CAPPOS and CAPNEG registers whenever a flip occurs on the input, and sets the flag bit CHxCAPF to 1
- 3'b100, PWM measurement mode, when the timer will automatically measure the PWM period and duty cycle.

timer saves the count difference between the two rising edges of the input signal (i.e., the PWM period) in the CAPPRD register, and saves the count difference between the rising edge and the falling edge of the signal (i.e., the PWM duty cycle) in the CAPDITY register. The CAPPOS register still holds the counter value at the time of the most recent rising edge of the signal.

In this mode, when the rising edge of the input signal is captured, the flag bit CHxCAPF is set to 1.

In PWM measurement mode, the user can configure CHxCR [MONITOR_EN] to detect whether the input signal period (or rising edge, selected by CHxCR [MONITOR_SEL]) is within the range (greater than CMP0 and less than CMP1). If it exceeds the range, an interrupt (flag bit CHxCAPF) or a DMA request

can be generated

Note that the values of the CAPPOS register, CAPNEG register, CAPPRD register, and CAPDITY register will be refreshed when a new edge is captured, depending on different settings, and the previously captured values will be lost.

Note that in PWM measurement mode, the user should pay attention to the value of the counter CNT. If the counter overload occurs between adjacent signal edge jumps, if the counter value returns to 0, the values of the CAPPRD register and the CAPDITY register will be inaccurate.

45.2.4 Single Mode

The user can configure the CHxCR[OPMODE] bit to enable single-shot mode, where the counter stops after reaching the overloaded value.

45.2.5 burst mode

Users can configure the CHxCR[BURST_MODE] bit to enable burst mode, in which the counter cycles a specified amount and then stops.

users can configure the number of bursts through the BURST_CFG register.

Note that single mode and burst mode cannot be enabled simultaneously.

45.2.6 External counting mode

The user can configure the CHxCR[CNT_MODE] bit to enable external counting mode, in which the counter uses the rising edge of the input capture signal as a counter plus signal.

45.2.7 Simple Encoder Mode

Users can configure GCR[QE101] or GCR[QE123] to use two counter channels as a simple encoder module.

supports AB mode, UD mode and PD mode. Channels 0 and 1 count the number of forward and reverse lines respectively (similar for channels 2/3).

- AB mode, the input signal is a quadrature signal. In this mode, the rising edge of the capture signal of each channel is only valid when the capture input of another channel is 1.
For example, when Channel 0 captures the rising edge of a signal, if the capture input of Channel 1 is 1, the signal enters Channel 0 for counting; if the capture input of Channel 1 is 0, Channel 0 does not count.
- UD mode, the input signal is a positive and negative rotation pulse. This mode does not process the capture input. The rising edge of channel 0 is used as the forward rotation input, and the rising edge of channel 1 is used as the reverse rotation input.
- PD mode. In this mode, channel 0 captures the input as a count signal, and channel 1 captures the input as a direction signal. Channel 0 counts when the direction is 1, and channel 1 counts when the direction is 0.

45.2.8 Interrupts and DMA

timer supports generating the following interrupts:

- CHxRLDF flag position is 1, which means that the counter count reaches the value of the overload register, or the counter overload is triggered by the synchronous trigger input SYNCI
- CHxCMPxF flag position is 1, that is, when the output comparison function is turned on, when the counter reaches the value of CMP0 or CMP1
- When the CHxCAPF flag position is 1, that is, when the input capture function is turned on, when the input channel captures a specified edge transition

The above interrupt request can be turned on or off by setting the corresponding bit in the CHxIRQEN register.

The user can check all flag bits by reading the CHxSR register, and the flag bits can be cleared by writing 1 to the corresponding flag bit in the SR register.

The timer supports generating the following DMA requests:

- CHxRLDF flag bit 1, that is, the counter count reaches the value of the overload register, or the counter overload is triggered by the synchronous trigger input SYNCI

- The CHxCMPxF flag position is 1, that is, when the output comparison function is turned on, the counter reaches the value of CMP0 or CMP1
- When the CHxCAPF flag position is 1, that is, when the input capture function is turned on, when the input channel captures a specified edge transition

By setting the CHxCR[DMASEL] register bit, you can choose to select one of the above requests as a DMA request output.

You can turn on or off DMA requests by setting the CHxCR [DMAEN] register bit.

45.2.9 debug mode support

universal timer supports debugging, that is, when the kernel enters debug mode (Debug), the counter is suspended. This function takes effect when the user sets the CHxCR[DBGPAUSE] bit to 1.

45.3 Timer Register

45.3.1 register description

timer is as follows:

GPTMR0 base address: 0xF0000000

GPTMR1 base address: 0xF0004000

GPTMR2 base address: 0xF0008000

GPTMR3 base address: 0xF000C000

NTMR0 base address: 0xF1410000

PTMR base address: 0xF4120000

Offset	Register Name	Description	Reset Value
0x0000	CHANNEL[CH0][CR]	Control Register	0x00000000
0x0004	CHANNEL[CH0][CMP][CMP0]	Comparator register 0	0xFFFFFFFF0
0x0008	CHANNEL[CH0][CMP][CMP1]	Comparator register 1	0xFFFFFFFF0
0x000C	CHANNEL[CH0][RLD]	Reload register	0xFFFFFFFF
0x0010	CHANNEL[CH0][CNTUPTVAL]	Counter update value register	0x00000000
0x0014	CHANNEL[CH0][BURST_CFG]	burst_cfg	0x00000000
0x0018	CHANNEL[CH0][BURST_COUNT]	burst_count	0x00000000
0x0020	CHANNEL[CH0][CAPPOS]	Capture rising edge register	0x00000000
0x0024	CHANNEL[CH0][CAPNEG]	Capture falling edge register	0x00000000
0x0028	CHANNEL[CH0][CAPPRD]	PWM period measure register	0x00000000
0x002C	CHANNEL[CH0][CAPDTY]	PWM duty cycle measure register	0x00000000
0x0030	CHANNEL[CH0][CNT]	Counter	0x00000000

Offset	Register Name	Description	Reset Value
0x0040	CHANNEL[CH1][CR]	Control Register	0x00000000
0x0044	CHANNEL[CH1][CMP][CMP0]	Comparator register 0	0xFFFFFFFF0
0x0048	CHANNEL[CH1][CMP][CMP1]	Comparator register 1	0xFFFFFFFF0
0x004C	CHANNEL[CH1][RLD]	Reload register	0xFFFFFFFF
0x0050	CHANNEL[CH1][CNTUPTVAL]	Counter update value register	0x00000000
0x0054	CHANNEL[CH1][BURST_CFG]	burst_cfg	0x00000000
0x0058	CHANNEL[CH1][BURST_COUNT]	burst_count	0x00000000
0x0060	CHANNEL[CH1][CAPPOS]	Capture rising edge register	0x00000000
0x0064	CHANNEL[CH1][CAPNEG]	Capture falling edge register	0x00000000
0x0068	CHANNEL[CH1][CAPPRD]	PWM period measure register	0x00000000
0x006C	CHANNEL[CH1][CAPDTY]	PWM duty cycle measure register	0x00000000
0x0070	CHANNEL[CH1][CNT]	Counter	0x00000000
0x0080	CHANNEL[CH2][CR]	Control Register	0x00000000
0x0084	CHANNEL[CH2][CMP][CMP0]	Comparator register 0	0xFFFFFFFF0
0x0088	CHANNEL[CH2][CMP][CMP1]	Comparator register 1	0xFFFFFFFF0
0x008C	CHANNEL[CH2][RLD]	Reload register	0xFFFFFFFF
0x0090	CHANNEL[CH2][CNTUPTVAL]	Counter update value register	0x00000000
0x0094	CHANNEL[CH2][BURST_CFG]	burst_cfg	0x00000000
0x0098	CHANNEL[CH2][BURST_COUNT]	burst_count	0x00000000
0x00A0	CHANNEL[CH2][CAPPOS]	Capture rising edge register	0x00000000
0x00A4	CHANNEL[CH2][CAPNEG]	Capture falling edge register	0x00000000
0x00A8	CHANNEL[CH2][CAPPRD]	PWM period measure register	0x00000000
0x00AC	CHANNEL[CH2][CAPDTY]	PWM duty cycle measure register	0x00000000
0x00B0	CHANNEL[CH2][CNT]	Counter	0x00000000
0x00C0	CHANNEL[CH3][CR]	Control Register	0x00000000
0x00C4	CHANNEL[CH3][CMP][CMP0]	Comparator register 0	0xFFFFFFFF0
0x00C8	CHANNEL[CH3][CMP][CMP1]	Comparator register 1	0xFFFFFFFF0
0x00CC	CHANNEL[CH3][RLD]	Reload register	0xFFFFFFFF
0x00D0	CHANNEL[CH3][CNTUPTVAL]	Counter update value register	0x00000000
0x00D4	CHANNEL[CH3][BURST_CFG]	burst_cfg	0x00000000

Offset	Register Name	Description	Reset Value
0x00D8	CHANNEL[CH3][BURST_COUNT]	burst_count	0x00000000
0x00E0	CHANNEL[CH3][CAPPOS]	Capture rising edge register	0x00000000
0x00E4	CHANNEL[CH3][CAPNEG]	Capture falling edge register	0x00000000
0x00E8	CHANNEL[CH3][CAPPRD]	PWM period measure register	0x00000000
0x00EC	CHANNEL[CH3][CAPDTY]	PWM duty cycle measure register	0x00000000
0x00F0	CHANNEL[CH3][CNT]	Counter	0x00000000
0x0200	SR	Status register	0x00000000
0x0204	IRQEN	Interrupt request enable register	0x00000000
0x0208	GCR	Global control register	0x00000000

Table 190: GPTMR Register List

timer are described in detail as follows:

45.3.2 CHANNEL[CR] (0x0 + 0x40 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
CNTUPT	RSVD											BURST_MODE	CNT_MODE	OPMODE	MONITOR_SEL	MONITOR_LEN	CNTRST	SYNCFW	SYNCFEN	SYNCIREN	CEN	CMPIINIT	CMPIEN	DMASEL	DMAEN	SWSYNCIEN	DBGPAUSE	CAPMODE						
WO	N/A											RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW			
0	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			

CHANNEL[CR] [31:0]

Bits	Name	Description
31	CNTUPT	1- update counter to new value as CNTUPTVAL This bit will be auto cleared after 1 cycle
19	BURST_MODE	set to enable burst mode, timer will reload configured times(burst_cfg), then stop. user need clear CEN and set it to start timer again. NOTE: do not set burst_mode and opmode at same time
18	CNT_MODE	0: internal counting mode, timer increase each gptmr clock cycle. 1: external counting mode, timer increase at each input signal posedge, reload/compare feature can still work but change at input signal posedge.
17	OPMODE	0: round mode 1: one-shot mode, timer will stopped at reload point.user need clear CEN and set it to start timer again. NOTE: reload irq will be always set at one-shot mode at end

Bits	Name	Description
16	MONITOR_SEL	set to monitor input signal high level time(chan_meas_high) clr to monitor input signal period(chan_meas_prd)
15	MONITOR_EN	set to monitor input signal period or high level time. When this bit is set, if detected period less than val_0 or more than val_1, will set related irq_sts only can be used when trig_mode is selected as measure mode(100) the time may not correct after reload, so monitor is disabled after reload point, and enabled again after two continul posedge. if no posedge after reload for more than val_1, will also assert irq_capt
14	CNTRST	1- reset counter
13	SYNCFLW	1- enable this channel to reset counter to reload(RLD) together with its previous channel. This bit is not valid for channel 0.
12	SYNCIFEN	1- SYNCI is valid on its falling edge
11	SYNCIREN	1- SYNCI is valid on its rising edge
10	CEN	1- counter enable
9	CMPINIT	Output compare initial polariarity 1- The channel output initial level is high 0- The channel output initial level is low User should set this bit before set CMPEN to 1.
8	CMPEN	1- Enable the channel output compare function. The output signal can be generated per comparator (CMPx) settings.
7-6	DMASEL	select one of DMA request: 00- CMP0 flag 01- CMP1 flag 10- Input signal toggle captured 11- RLD flag, counter reload;
5	DMAEN	1- enable dma
4	SWSYNCIEN	1- enable software sync. When this bit is set, counter will reset to RLD when swsynct bit is set
3	DBGPAUSE	1- counter will pause if chip is in debug mode
2-0	CAPMODE	This bitfield define the input capture mode 100: width measure mode, timer will calculate the input signal period and duty cycle 011: capture at both rising edge and falling edge 010: capture at falling edge 001: capture at rising edge 000: No capture

CHANNEL[CR] Bits

45.3.3 CHANNEL[**CMP**] (0x4 + 0x40 * n + 0x4 * m)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP																															
RW																															
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0

CHANNEL[**CMP**] [31:0]

Bits	Name	Description
31-0	CMP	compare value 0

CHANNEL[**CMP**] Bits

45.3.4 CHANNEL[**RLD**] (0xC + 0x40 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RLD																															
RW																															
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

CHANNEL[**RLD**] [31:0]

Bits	Name	Description
31-0	RLD	reload value

CHANNEL[**RLD**] Bits

45.3.5 CHANNEL[**CNTUPTVAL**] (0x10 + 0x40 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNTUPTVAL																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CHANNEL[**CNTUPTVAL**] [31:0]

Bits	Name	Description
31-0	CNTUPTVAL	counter will be set to this value when software write cntupt bit in CR

CHANNEL[CNTUPTVAL] Bits

45.3.6 CHANNEL[BURST_CFG] (0x14 + 0x40 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																BURST_CFG															
N/A																RW															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CHANNEL[BURST_CFG] [31:0]

Bits	Name	Description
15-0	BURST_CFG	

CHANNEL[BURST_CFG] Bits

45.3.7 CHANNEL[BURST_COUNT] (0x18 + 0x40 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																BURST_COUNT															
N/A																RO															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CHANNEL[BURST_COUNT] [31:0]

Bits	Name	Description
15-0	BURST_COUNT	

CHANNEL[BURST_COUNT] Bits

45.3.8 CHANNEL[CAPPOS] (0x20 + 0x40 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPPOS																															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CHANNEL[CAPPOS] [31:0]

Bits	Name	Description
31-0	CAPPOS	This register contains the counter value captured at input signal rising edge

CHANNEL[CAPPOS] Bits

45.3.9 CHANNEL[CAPNEG] (0x24 + 0x40 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPNEG																															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CHANNEL[CAPNEG] [31:0]

Bits	Name	Description
31-0	CAPNEG	This register contains the counter value captured at input signal falling edge

CHANNEL[CAPNEG] Bits

45.3.10 CHANNEL[CAPPRD] (0x28 + 0x40 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPPRD																															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CHANNEL[CAPPRD] [31:0]

Bits	Name	Description
31-0	CAPPRD	This register contains the input signal period when channel is configured to input capture measure mode.

CHANNEL[CAPPRD] Bits

45.3.11 CHANNEL[CAPDTY] (0x2C + 0x40 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MEAS_HIGH																RO															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CHANNEL[CAPDTY] [31:0]

Bits	Name	Description
31-0	MEAS_HIGH	This register contains the input signal duty cycle when channel is configured to input capture measure mode.

CHANNEL[CAPDTY] Bits

45.3.12 CHANNEL[CNT] (0x30 + 0x40 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNTER																RO															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CHANNEL[CNT] [31:0]

Bits	Name	Description
31-0	COUNTER	32 bit counter value

CHANNEL[CNT] Bits

45.3.13 SR (0x200)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																CH3CMP1F	CH3CMP0F	CH3CAPF	CH3RLDF	CH2CMP1F	CH2CMP0F	CH2CAPF	CH2RLDF	CH1CMP1F	CH1CMP0F	CH1CAPF	CH1RLDF	CH0CMP1F	CH0CMP0F	CH0CAPF	CH0RLDF
N/A																W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SR [31:0]

Bits	Name	Description
15	CH3CMP1F	channel 3 compare value 1 match flag
14	CH3CMP0F	channel 3 compare value 1 match flag

Bits	Name	Description
13	CH3CAPF	channel 3 capture flag, the flag will be set at the valid capture edge per CAPMODE setting. If the capture channel is set to measure mode, the flag will be set at rising edge.
12	CH3RLDF	channel 3 counter reload flag
11	CH2CMP1F	channel 2 compare value 1 match flag
10	CH2CMP0F	channel 2 compare value 1 match flag
9	CH2CAPF	channel 2 capture flag, the flag will be set at the valid capture edge per CAPMODE setting. If the capture channel is set to measure mode, the flag will be set at rising edge.
8	CH2RLDF	channel 2 counter reload flag
7	CH1CMP1F	channel 1 compare value 1 match flag
6	CH1CMP0F	channel 1 compare value 1 match flag
5	CH1CAPF	channel 1 capture flag, the flag will be set at the valid capture edge per CAPMODE setting. If the capture channel is set to measure mode, the flag will be set at rising edge.
4	CH1RLDF	channel 1 counter reload flag
3	CH0CMP1F	channel 1 compare value 1 match flag
2	CH0CMP0F	channel 1 compare value 1 match flag
1	CH0CAPF	channel 1 capture flag, the flag will be set at the valid capture edge per CAPMODE setting. If the capture channel is set to measure mode, the flag will be set at rising edge.
0	CH0RLDF	channel 1 counter reload flag

SR Bits

45.3.14 IRQEN (0x204)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																CH3CMP1EN	CH3CMP0EN	CH3CAPEN	CH3RLDEN	CH2CMP1EN	CH2CMP0EN	CH2CAPEN	CH2RLDEN	CH1CMP1EN	CH1CMP0EN	CH1CAPEN	CH1RLDEN	CH0CMP1EN	CH0CMP0EN	CH0CAPEN	CH0RLDEN
N/A																RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IRQEN [31:0]

Bits	Name	Description
15	CH3CMP1EN	1- generate interrupt request when ch3cmp1f flag is set
14	CH3CMP0EN	1- generate interrupt request when ch3cmp0f flag is set
13	CH3CAPEN	1- generate interrupt request when ch3capf flag is set
12	CH3RLDEN	1- generate interrupt request when ch3rldf flag is set
11	CH2CMP1EN	1- generate interrupt request when ch2cmp1f flag is set

Bits	Name	Description
10	CH2CMP0EN	1- generate interrupt request when ch2cmp0f flag is set
9	CH2CAPEN	1- generate interrupt request when ch2capf flag is set
8	CH2RLDEN	1- generate interrupt request when ch2rldf flag is set
7	CH1CMP1EN	1- generate interrupt request when ch1cmp1f flag is set
6	CH1CMP0EN	1- generate interrupt request when ch1cmp0f flag is set
5	CH1CAPEN	1- generate interrupt request when ch1capf flag is set
4	CH1RLDEN	1- generate interrupt request when ch1rldf flag is set
3	CH0CMP1EN	1- generate interrupt request when ch0cmp1f flag is set
2	CH0CMP0EN	1- generate interrupt request when ch0cmp0f flag is set
1	CH0CAPEN	1- generate interrupt request when ch0capf flag is set
0	CH0RLDEN	1- generate interrupt request when ch0rldf flag is set

IRQEN Bits

45.3.15 GCR (0x208)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD												QEI_TYPE23		QEI_TYPE01		RSVD										SWSYNCT					
N/A												RW		RW		N/A										W1C					
x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

GCR [31:0]

Bits	Name	Description
19-18	QEI_TYPE23	define capture_in process for channel 2/3
17-16	QEI_TYPE01	define capture_in process for channel 0/1, for qei mode only: 00: ud mode. nothing changed to capture_in. 01: ab mode, the capture_in will be asserted at posedge when another channel capture_in is high; 10: pd mode, capture_in[0] as signal, capture_in[1] as direction. if capture_in[1] is high, then capture_in[0] is connected to channel0, 0 is connected to cahnnel1; if capture_in[1] is low, then capture_in[0] is connected to channel1, 0 is connected to cahnnel0; 11: reserved
3-0	SWSYNCT	set this bitfield to trigger software counter sync event

GCR Bits

46 Watchdog EWDG

This chapter describes the functions and features of watchdog EWDG.

46.1 Feature Summary

The main features of the watchdog EWDG are:

- supports dog feeding during specific time windows
- supports unlocking protection before feeding dogs. It can be configured into multiple unlocking methods
- supports locking of critical control registers, which need to be unlocked before updating critical registers
- watchdog working clock source is configurable and can still work at 32K when the bus clock stops
- Critical control registers come with parity function
- The watchdog can be configured to automatically stop or continue working when the chip enters debug mode
- The watchdog can be configured to automatically stop or continue working when the chip enters low power mode

46.2 EWDG Architecture Chart

watchdog is as shown in [Figure 58](#).

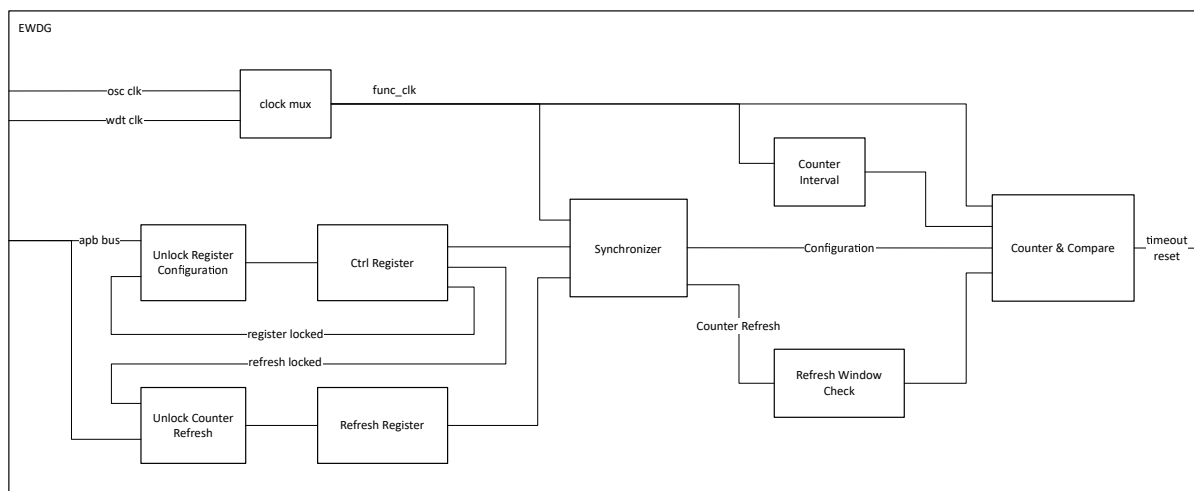


Figure 58: watchdog block diagram

46.3 Function Description

46.3.1 Feature Overview

EWDG has a built-in 32-bit timer. After several working clocks (the value is configurable, the maximum value is 32), the value of the timer increases. If the dog is not fed within the specified time, the value of the timer increases by more than the value specified in the reset value register, and a reset operation will be triggered. The dog feeding operation is to write a specific character to the dog feeding register: 32'h5A45_524F

For safety reasons, EWDG can request unlocking before changing critical registers and performing dog feeding operations respectively. If it is not unlocked or fails to unlock, a violation will occur and a corresponding

interrupt or reset will be triggered. In addition, window mode is supported. Dog feeding operations are only allowed to be performed within a specific time window. Dog feeding at other times will cause violations and trigger an interrupt or reset.

The reset signal generated by EWDG can also be sent to the IO pin output for simultaneous resetting of other devices in the system. Please refer to the PINMUX configuration information for specific whether the current project supports this function.

46.3.2 Clock selection

EWDG has three clock inputs, which are bus configuration clock, external working clock 1 and external working clock 2. The bus working clock is used to configure the registers and dog feeding operations of EWDG. External working clock 1 and external working clock 2 are the main working clocks of EWDG and can be selected by register configuration.

In general, the SoC will choose to connect the external working clock 1 to the bus configuration clock to achieve the purpose of fast response of the EWDG. Connect the external working clock 2 to the 32K clock to achieve a longer dog feeding interval and the EWDG can still work when the bus clock stops.

EWDG's timer can be incremented after several working clocks according to configuration. After receiving a legal dog feeding operation, the timer is cleared to 0, and a reset operation is generated after the counter reaches the pre-configured timeout value.

The timeout time calculation formula for EWDG is:

$$T_{timeout} = T_{clk\_period} * (1 \ll counter\_interval) * counter\_rst\_value$$

46.3.3 register configuration unlocking and parity requirements

EWDG has a parity check in control register 0 and configuration register 1. Each time when writing configuration register 0 and configuration register 1, check bits must be filled in in BIT31 of the corresponding register to ensure that the number of 1s in the register (including check bits) is always even, otherwise the register configuration fails and an interrupt or reset operation is generated

After the register protection enable is turned on, once the EWDG starts working by configuring the EWDG enable register, the register protection function is turned on. In the future, the unlocking operation needs to be performed before updating the control register 0, configuration register 1, and EWDG resetting the value. Registers in a protected state can only be changed after they are correctly unlocked. Updating the register configuration if they are not correctly unlocked will cause register update to fail and trigger an interrupt or reset operation.

After the register protection function takes effect, the register update operation is unlocked by writing the correct password to the Configuration Protection Unlock Register. The password shall be written to the Configuration Protection Unlock Register before the register protection enable function is activated.

Note: Once the register protection enable bit in control register 0 is turned on, the configuration protection unlock register is not allowed to be changed, so changes to this register must be configured before the register protection enable bit in control register 0 is turned on

register configuration protection is unlocked, software can change EWDG's control register 0/1 and reset value and other registers. EWDG will automatically turn on the register configuration protection function under

the following circumstances:

- If the EWDG is currently in a stopped state, it will automatically enter the protection state after the EWDG resumes working
- If the EWDG is currently working, it will automatically enter the protected state after a number of bus clock cycles (the number of cycles can be configured)

46.3.4 Unlock dog feeding operation

EWDG supports dog feeding protection. If this function is turned on, the dog feeding operation needs to be unlocked before feeding the dog each time. Otherwise, the current dog feeding fails and the corresponding interrupt or reset function will be triggered.

46.3.4.1 Dog feeding protection enable

If the dog feed protection enable bit in the control register 0 of the EWDG is configured, the dog feed protection mode will be entered after the EWDG starts working

46.3.4.2 Unlock dog feeding operation

When the EWDG dog feeding protection takes effect, the unlocking operation needs to be performed before feeding the dog. The unlocking operation method can be configured to one of the following methods:

- Fixed configuration password: Write the initial password into the dog feeding protection unlocking register during the initial stage, and then write the same configuration password into this register to unlock
- Loop left shift password: In the initial stage, the initial password is written into the dog feeding protection unlocking register, and then the loop left shift result of the password is written into this register to unlock it. After that, the password is automatically updated to the left shift result of this cycle.
- LSFR password: Write the initial password into the dog feeding protection unlocking register at the initial stage, and then write the LSFR result into this register according to the formula of $X^{15} + 1$ to unlock, and the password is updated to this LSFR result
- Fixed value unlocking: There is no need to configure the initial password. During the unlocking stage, write a fixed value of 16'h55AA to the dog feeding protection unlocking register to unlock it.

Note: The dog feed protection unlock register can only be configured before the dog feed protection enable bit in control register 0 is enabled. Once the dog feed protection enable bit is turned on, the value of the dog feed protection unlock register will not be changed

Every time the dog feeding protection is unlocked, EWDG will automatically turn on the protection function again under the following circumstances:

- If the dog feeding operation has been completed, the protection function will be turned on immediately and needs to be unlocked again before feeding the dog next time
- If the unlocking operation is not completed within a specific time (the time can be configured) after unlocking, the current unlocking time will expire, and the dog feeding protection state will be re-entered, and a dog feeding violation flag will be generated at the same time.

46.3.5 Window Mode

EWDG can be configured in window mode. After window mode is turned on, EWDG only allows dog feeding operations to be performed within specific windows of the timer. Failure to feed the dog within the window time will cause dog feeding failure and trigger corresponding interrupt or reset operations.

The lower limit of the window can be configured to 4/8, 5/8, 6/8, and 7/8 of the watchdog reset value register. The upper limit value of the window is one-sixteenth of several reset value registers added to the lower limit after dividing the reset value register into sixteen parts.

$$window\_lower = reset\_counter\_value * (1/2 + win\_lower\_reg * 1/8)$$

$$window\_upper = window\_lower + reset\_counter\_value/16 * win\_upper\_reg$$

Note: If the upper limit register is not configured, or the upper limit exceeds the value of the reset value register, the value of the reset value register will be used as the upper limit for feeding the dog, and the dog feeding operation can be completed before timeout reset

46.3.6 Working status in Debug mode

EWDG suspends work when the chip is in debug mode by default, and continues to work after the chip exits debug mode. You can configure EWDG to work normally in debug mode through registers.

46.3.7 Operating status in low power mode

EWDG suspends work when the chip is in low power mode by default, and continues to work after the chip exits low power mode. The EWDG can be configured through registers to operate normally in low power mode.

46.4 EWDG register list

EWDG is as follows:

EWDG0 base address: 0xF00B0000

EWDG1 base address: 0xF00B4000

PEWDG base address: 0xF4128000

Offset	Register Name	Description	Reset Value
0x0000	CTRL0	wdog ctrl register 0 Note: Parity check is required once writing to this register. The result should be zero by modular two addition of all bits	0x00000000

Offset	Register Name	Description	Reset Value
0x0004	CTRL1	wdog ctrl register 1 Note: Parity check is required once writing to this register. The result should be zero by modular two addition of all bits	0x00000000
0x0008	OT_INT_VAL	wdog timeout interrupt counter value	0x00000000
0x000C	OT_RST_VAL	wdog timeout reset counter value	0x00000000
0x0010	WDT_REFRESH_REG	wdog refresh register	0x00000000
0x0014	WDT_STATUS	wdog status register	0x00000000
0x0018	CFG_PROT	ctrl register protection register	0x00000000
0x001C	REF_PROT	refresh protection register	0x00000000
0x0020	WDT_EN	Wdog enable	0x00000000
0x0024	REF_TIME	Refresh period value	0x00000000

Table 191: EWDG Register List

46.5 EWDG register description

EWDG is as follows:

46.5.1 CTRL0 (0x0)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD		CLK_SEL	RSVD	DIV_VALUE			WIN_EN	WIN_LOWER		CFG_LOCK	RSVD					REF_OT_REQ	WIN_UPPER			RSVD					REF_LOCK	REF_UNLOCK_MEC		EN_DBG		EN_LP	
N/A		RW	N/A	RW			RW	RW		RW	N/A					RW	RW			N/A					RW	RW		RW		RW	
x	x	0	x	0	0	0	0	0	0	0	x	x	x	x	x	0	0	0	0	x	x	x	x	x	x	0	0	0	0	0	0

CTRL0 [31:0]

Bits	Name	Description
29	CLK_SEL	clock select 0: bus clock 1: ext clock
27-25	DIV_VALUE	clock divider, the clock divider works as $2^{\text{div\_value}}$ for wdt counter
24	WIN_EN	window mode enable

Bits	Name	Description
23-22	WIN_LOWER	Once window mode is opened, the lower counter value to refresh wdt 00: 4/8 overtime value 01: 5/8 of overtime value 10: 6/8 of overtime value 11: 7/8 of overtime value
21	CFG_LOCK	The register is locked and unlock is needed before re-config registers Once the lock mechanism takes effect, the CTRL0, CTRL1, timeout int register, timeout rst register, needs unlock before re-config them. The register update needs to be finished in the required period defined by UPD_OT_TIME register
15	REF_OT_REQ	If refresh event has to be limited into a period after refresh unlocked. Note: the refresh overtime counter works in bus clock domain, not in wdt function clock domain. The wdt divider doesn't take effect for refresh counter
14-12	WIN_UPPER	The upper threshold of window value The window period upper limit is: lower_limit + (overtime_rst_value / 16) * upper_reg_value If this register value is zero, then no upper level limitation
5	REF_LOCK	WDT refresh has to be unlocked firstly once refresh lock is enable.
4-3	REF_UNLOCK_MECH	Unlock refresh mechanism 00: the required unlock password is the same with refresh_psd_register 01: the required unlock password is a ring shift left value of refresh_psd_register 10: the required unlock password is always 16'h55AA, no matter what refresh_psd_register is 11: the required unlock password is a LSFR result of refresh_psd_register, the characteristic polynomial is $X^{15} + 1$
2	EN_DBG	WTD enable or not in debug mode
1-0	EN_LP	WTD enable or not in low power mode 2'b00: wdt is halted once in low power mode 2'b01: wdt will work with 1/4 normal clock freq in low power mode 2'b10: wdt will work with 1/2 normal clock freq in low power mode 2'b11: wdt will work with normal clock freq in low power mode

CTRL0 Bits

46.5.2 CTRL1 (0x4)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								REF_FAIL_RST_EN	REF_FAIL_INT_EN	UNL_REF_FAIL_RST_EN	UNL_REF_FAIL_INT_EN	RSVD		OT_RST_EN	OT_INT_EN	RSVD								CTL_VIO_RST_EN	CTL_VIO_INT_EN	UNL_CTL_FAIL_RST_EN	UNL_CTL_FAIL_INT_EN	PARITY_FAIL_RST_EN	PARITY_FAIL_INT_EN	RSVD	
N/A								RW	RW	RW	RW	N/A		RW	RW	N/A								RW	RW	RW	RW	RW	RW	N/A	
x	x	x	x	x	x	x	x	0	0	0	0	x	x	0	0	x	x	x	x	x	x	x	x	0	0	0	0	0	0	x	x

CTRL1 [31:0]

Bits	Name	Description
23	REF_FAIL_RST_EN	Refresh violation will trigger an reset. These event will be taken as a refresh violation: 1) Not refresh in the window once window mode is enabled 2) Not unlock refresh firstly if unlock is required 3) Not refresh in the required time after unlock, once refresh unlock overtime is enabled. 4) Not write the required word to refresh wdt.
22	REF_FAIL_INT_EN	Refresh violation will trigger an interrupt
21	UNL_REF_FAIL_RST_EN	Refresh unlock fail will trigger a reset
20	UNL_REF_FAIL_INT_EN	Refresh unlock fail will trigger a interrupt
17	OT_RST_EN	WDT overtime will generate a reset
16	OT_INT_EN	WDT can generate an interrupt warning before timeout
7	CTL_VIO_RST_EN	Ctrl update violation will trigger a reset The violation event is to try updating the locked register before unlock them
6	CTL_VIO_INT_EN	Ctrl update violation will trigger a interrupt
5	UNL_CTL_FAIL_RST_EN	Unlock register update failure will trigger a reset
4	UNL_CTL_FAIL_INT_EN	Unlock register update failure will trigger a interrupt
3	PARITY_FAIL_RST_EN	Parity error will trigger a reset A parity check is required once writing to ctrl0 and ctrl1 register. The result should be zero by modular two addition of all bits

Bits	Name	Description
2	PARITY_FAIL_INT_EN	Parity error will trigger a interrupt

CTRL1 Bits

46.5.3 OT_INT_VAL (0x8)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																OT_INT_VAL															
N/A																RW															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OT_INT_VAL [31:0]

Bits	Name	Description
15-0	OT_INT_VAL	WDT timeout interrupt value

OT_INT_VAL Bits

46.5.4 OT_RST_VAL (0xC)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																OT_RST_VAL															
N/A																RW															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OT_RST_VAL [31:0]

Bits	Name	Description
15-0	OT_RST_VAL	WDT timeout reset value

OT_RST_VAL Bits

46.5.5 WDT_REFRESH_REG (0x10)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																WDT_REFRESH_REG															
																WO															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

WDT_REFRESH_REG [31:0]

Bits	Name	Description
31-0	WDT_REFRESH_REG	Write this register by 32'h5A45_524F to refresh wdog Note: Reading this register can read back wdt real time counter value, while it is only used by debug purpose

WDT_REFRESH_REG Bits

46.5.6 WDT_STATUS (0x14)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																RSVD															
																N/A															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0

WDT_STATUS [31:0]

Bits	Name	Description
6	PARITY_ERROR	parity error Write one to clear the bit
5	OT_RST	Timeout happens, a reset will happen once enable bit set This bit can be cleared only by refreshing wdt or reset
4	OT_INT	Timeout happens, a interrupt will happen once enable bit set This bit can be cleared only by refreshing wdt or reset
3	CTL_UNL_FAIL	Unlock ctrl reg update protection fail Write one to clear the bit
2	CTL_VIO	Violate register update protection mechanism Write one to clear the bit
1	REF_UNL_FAIL	Refresh unlock fail Write one to clear the bit

Bits	Name	Description
0	REF_VIO	Refresh fail Write one to clear the bit

WDT_STATUS Bits

46.5.7 CFG_PROT (0x18)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD												UPD_OT_TIME				UPD_PSD															
N/A												RW				RW															
x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CFG_PROT [31:0]

Bits	Name	Description
19-16	UPD_OT_TIME	The period in which register update has to be in after unlock The required period is less than: $128 * 2^{\text{UPD\_OT\_TIME}} * \text{bus\_clock\_cycle}$
15-0	UPD_PSD	The password of unlocking register update

CFG_PROT Bits

46.5.8 REF_PROT (0x1C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD												REF_UNL_PSD																			
N/A												RW																			
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

REF_PROT [31:0]

Bits	Name	Description
15-0	REF_UNL_PSD	The password to unlock refreshing

REF_PROT Bits

46.5.9 WDT_EN (0x20)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																WDOG_EN															
N/A																RW															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0

WDT_EN [31:0]

Bits	Name	Description
0	WDOG_EN	Wdog is enabled, the re-written of this register is impacted by enable lock function

WDT_EN Bits

46.5.10 REF_TIME (0x24)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																REFRESH_PERIOD															
N/A																RW															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

REF_TIME [31:0]

Bits	Name	Description
15-0	REFRESH_PERIOD	The refresh period after refresh unlocked Note: the refresh overtime counter works in bus clock domain, not in wdt function clock domain. The wdt divider doesn't take effect for refresh counter

REF_TIME Bits

46.6 Watchdog configuration instructions

46.6.1 Configuration Reference: Window working mode, dog feeding protection enabled

requires the EWDG clock to select working clock 2 (OSC32K), work in window mode, and allow dog feeding during the 4/8 to 7/8 phase of the timer. At the same time, it needs to be unlocked before feeding the dog, and the unlocking password is shifted to the left in a loop. The

reference configuration is:

- Configure the dog protection register (REF_PROT) and write the initial password PSD. If you expect to complete the dog feeding operation within 512 bus clocks after unlocking, configure the dog feeding limit time register (REF_TIME) to 512.
- The dog protection mode unlocking method (REF_UNLOCK_MEC) value in the configuration control register is 01, and the cyclic left shift method is selected. Configure dog feeding protection mode enable (REF_LOCK).
- configures CLK_SEL in control register 0 to 1, and selects OSC 32K.
- Configure WIN_EN in control register 0 to open window mode. The lower limit value of WIN_LOWER is 00 (4/8 of the timeout time of the selection timer). If the upper limit is increased by 3/8 of the timeout time based on the lower limit, the value of WIN_UPPER is configured to 6 (6 / 16 + lower limit value).
- If you expect EWDG to generate a reset signal after a timeout, configure the timeout reset enable (OT_RST_EN) in Control Register 1. If you want to generate an interrupt by feeding the dog not as required, configure the dog feeding operation violation interrupt enable (REF_FAIL_INT_EN) and the dog feeding protection unlocking failure interrupt enable (UNL_REF_FAIL_INT_EN) in control register 1. In this way, an interrupt operation will occur if the dog fails to feed or the dog fails to unlock the dog.
- Note that when updating configuration register 0 and configuration register 1, ensure that parity passes, and the number of bits with a value of 1 is even.
- Configure the enable register WDT_EN, write 1 to WDOG_EN, and the watchdog starts working
- When the timer is ready to feed the dog during the window allowing dog feeding, the dog feeding protection is first unlocked. Write the result of the left shift of the initial password cycle in the dog feeding protection register (REF_PROT) into the REF_PROT register to unlock the dog feeding operation, and at the same time, the password in the dog feeding protection register is updated to the result of the left shift of the initial password cycle.
- writes the reserved word 32'h5A45524F to the dog feed register (WDT_REFRESH_REG) in 512 bus clocks to complete the dog feed operation.

46.6.2 Configuration Reference: Register Configuration Locking and Unlocking

wants to update configuration register 1 after the configuration register has been locked, and changes the behavior of unlocking failure from interrupt to reset.

- configures the configuration update time (UPD_OD_TIME) of the configuration protection register to 2, hoping to complete the configuration within 512 cycles ($128 * 2^2$) after the configuration is unlocked. Write the initial password PSD to UPD_PSD.
- turns on the register protection enable (CFG_LOCK) of control register 0.
- After EWDG is enabled (the enable register is written to 1), the configuration protection function is turned on. If a violation occurs if the control register and reset value register bits are updated without unlocking, a violation occurs
- When you want to unlock, first write the password into the configuration protection register to unlock the configuration protection. Writing the enable register bit 0x0 stops EWDG (recommended to prevent unexpected situations in critical conditions). Then, control register 1 is updated within 512 system clock cycles to turn on the dog feeding operation violation reset enable (REF_FAIL_RST_EN). Configure enable register 0x01 restarts EWDG, and configure the protection function to automatically restart after EWDG works.

47 Communication Peripherals Overview

This chapter introduces the communication peripherals available on this product.

The communication peripherals on this product include 9 UARTs, 4 SPIs, 4 I2Cs, 4 CANs, 1 ENET, and 1 USB.

47.1 Universal Asynchronous Receiver Transmitter (UART, PUART)

This product supports 9 Universal Asynchronous Receiver Transmitters (UARTs), enabling bidirectional asynchronous communication with external devices.

8 of these UARTs are located in the system power domain, referred to as UART0~7.

1 UART is located in the power management domain, referred to as PUART. PUART can maintain operation when the system power domain loses power, and its generated interrupts can be used to wake the system from low-power modes. PUART does not support generating DMA requests.

47.2 Serial Peripheral Interface (SPI)

This product supports 4 Serial Peripheral Interface (SPI) buses, enabling full-duplex, synchronous communication with external devices. SPI supports the classic unidirectional 2-data-line mode (i.e., MISO, MOSI), as well as bidirectional 2-data-line and 4-data-line modes, namely Dual-SPI and Quad-SPI.

47.3 Inter-Integrated Circuit (I2C)

This product supports 4 Inter-Integrated Circuit (I2C) buses. I2C supports Standard mode (up to 100 Kbps), Fast mode (up to 400 Kbps), and Fast+ mode (up to 1 Mbps).

47.4 Controller Area Network (MCAN)

This product supports 4 Controller Area Network modules, MCAN0~3. MCAN complies with the CAN bus protocol 2.0A and 2.0B specifications, and also supports the CAN-FD protocol.

On this product, MCAN supports Time-Triggered CAN (TTCAN) communication. MCAN can directly load timestamp information from the timestamp output port of the Precision Time Protocol Controller (PTPC) module.

47.5 Precision Time Protocol Controller (PTPC)

This product supports 1 Precision Time Protocol Controller (PTPC) module. PTPC supports 2 sets of timestamps, capable of generating precise timestamps with second and nanosecond resolution for synchronizing clocks between different nodes on a network.

PTPC supports 2 timestamp output ports, which are connected to CAN0~3, respectively. The CAN modules can directly load timestamp information from these ports.

PTPC supports 2 timestamp input capture signals, CAP0 and CAP1, enabling the capture of the current timestamp based on these input signals. CAP0 and CAP1 originate from the interconnect manager TRGM. PTPC supports outputting 2 timestamp output compare signals, CMP0 and CMP1, when a timestamp match occurs. CMP0 and CMP1 are output to the interconnect manager TRGM.

47.6 Ethernet Controller (ENET)

This product supports 1 Ethernet Controller (ENET). ENET supports the IEEE 802.3 standard Ethernet MAC, with data transfer rates of 10/100/1000 Mbps, and supports PHYs in RMII, RGMII, and MII modes.

On this product, the ENET controller supports event output signals EVTO. The PTP outputs 0~1 of ENET can be output via IO through EVTO0~1.

47.7 Universal Serial Bus (USB)

This product supports 1 Universal Serial Bus (USB). The USB integrates an HS PHY, supporting USB 2.0 Low-Speed, Full-Speed, and High-Speed transfers. USB supports Host, Device, and OTG modes.

On this product, the USB controller supports outputting the USB SOF (Start of Frame) signal. This signal can be connected to other on-chip modules via the interconnect manager TRGM.

48 Universal Asynchronous Receiver Transmitter (UART)

This chapter introduces the functions and features of the Universal Asynchronous Receiver Transmitter (UART).

48.1 Feature Summary

This chapter introduces the main features of the UART:

- Supports 5 to 8-bit data length
- Configurable stop bits: 1 bit, 1.5 bits, or 2 bits
- Configurable parity bits: odd parity, even parity, stick parity
- Supports DMA data transfer
- Supports configurable baud rate, independent baud rate generation clock
- Supports hardware flow control
- Supports error detection such as parity error, data FIFO overflow
- 16-byte TX FIFO and RX FIFO
- Supports RX IDLE status flag
- Supports hardware-triggered transmission when used with the TRIGGER MUX
- Supports various interrupts

The block diagram of the UART is shown in ??.

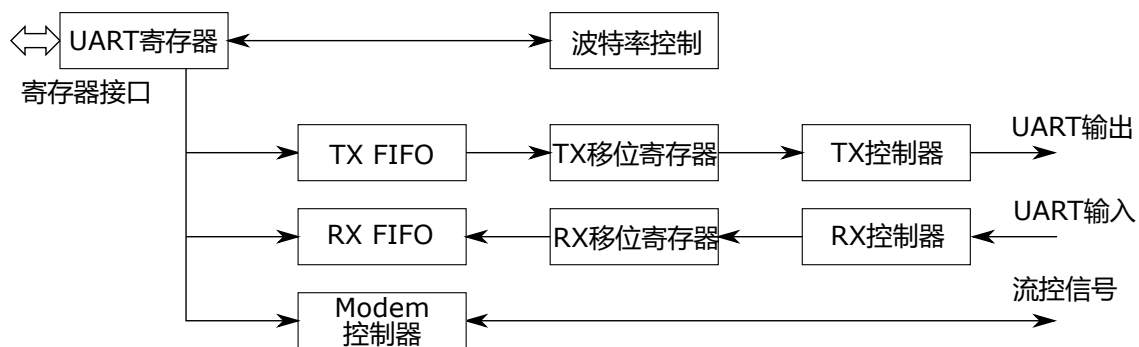


Figure 59: UART Block Diagram

48.2 Functional Description

This section describes the functions of the UART.

48.2.1 UART Transmission

The UART transmitter consists of the TX FIFO, TX shift register, and TX controller.

The TX FIFO holds the data to be transmitted and transfers it to the TX shift register.

The TX shift register is a parallel-to-serial converter that converts the transmit data into a serial bit stream.

When transmitting a character, the TX controller generates a START bit, a configurable PARITY bit, and a configurable-length STOP bit. The user can configure the parity and stop bits via the [LCR](#) register (Line Control Register).

The write port for the TX FIFO is the **THR** register (Transmitter Holding Register). The user must set the FIFOE bit in the **FCR** register (FIFO Control Register) to 1 to enable the TX FIFO.

48.2.2 UART Reception

The UART receiver consists of the RX FIFO, RX shift register, and RX controller.

The RX controller uses the oversampling clock generated by the baud rate control module to sample each bit of the incoming signal, shifting each received bit into the RX shift register.

The RX shift register performs serial-to-parallel conversion on the data and stores it into the RX FIFO.

The read port for the RX FIFO is the **RBR** register (Receiver Buffer Register). The user must set the FIFOE bit in the **FCR** register (FIFO Control Register) to 1 to enable the RX FIFO.

The RX controller supports detecting errors during data reception, such as parity errors, stop bit errors, RX FIFO overflow, etc.

48.2.3 Idle Detection

UART idle detection is divided into transmit idle detection and receive idle detection.

Taking receive idle detection as an example: idle detection is used to detect the idle time of the UART, which can be divided into pin idle and internal UART state machine idle, distinguished by the TX_IDLE_COND bit in IDLE_CFG. This function is enabled when the RX_IDLE_EN bit (idle detection enable) in IDLE_CFG is set to 1.

When the idle time reaches the time set by RX_IDLE_THR in IDLE_CFG (counted in bit units), an interrupt can be triggered. That is, when there is no data on the receive line within the set time (determined by RX_IDLE_THR), an interrupt is triggered. The interrupt indicated by the RXIDLE_FLAG in IIR2 is triggered. The principle of transmit idle detection is the same as receive idle detection.

48.2.4 Baud Rate Control

The baud rate control module divides the UART clock to obtain the baud rate clock. The divisor is 16 bits long, located in two registers, each holding an 8-bit divisor value. The MSB of the divisor is in the **DLM** register (Divisor Latch MSB), and the LSB is in the **DLL** register (Divisor Latch LSB).

The ratio of the UART clock to the baud rate is the oversampling rate. The oversampling rate **OSC** is stored in the **OSCR** register (Over Sample Control Register). **OSC** defaults to 16.

The formula for the divisor is as shown in equation (48.2.4):

$$Divisor = \frac{f_{uart\_clock}}{Baudrate \times OSC}$$

The RX controller uses the oversampling clock to sample the input data. Assuming the oversampling rate **OSC** is 16, when the first falling edge of the input signal (the START bit) is detected, the counter starts counting from 1 to 16. At count 8, the RX controller samples the input data. After counting to 16, the counter resets to 1, sampling the next data bit, and so on, until the STOP bit. The TX controller also uses the oversampling clock to generate the output data stream.

48.2.5 Modem Controller

The Modem controller provides Modem control functions and also supports automatic flow control, which can further reduce software control overhead.

UART transmit/receive flow control is implemented via RTS/CTS handshaking. Flow control prevents over-run errors caused by data transmission rates exceeding data consumption rates.

UART supports automatic flow control, including auto-RTS and auto-CTS functions. The former is used for data reception, and the latter for data transmission.

When auto-RTS is enabled, the UART's RTS output needs to be connected to the other UART's CTS signal.

- N is the RX FIFO trigger threshold
- B_n is the number of bytes received in the RX FIFO
- When $B_n \geq N$, RTS is set low
- When $B_n < N$, RTS is automatically set high, indicating to the other UART that it can send data

When auto-CTS is enabled, the UART's CTS input is connected to the other UART's RTS signal. The UART will only transmit data when CTS detects a low input. CTS must be set high before the stop bit of the current data transmission is issued, so the UART will not send the next data.

48.2.6 Loopback Mode

UART supports Loopback mode.

When Loopback mode is enabled, the UART's TX is connected to RX, and RTS is connected to CTS. Data sent via the [THR](#) register can be received from the [RBR](#) register.

48.2.7 Interrupts

Interrupts supported by the UART mainly include: receive idle interrupt, transmit idle interrupt, address match interrupt, data loss interrupt, flow control interrupt, receive status interrupt, transmit status interrupt, receive data valid interrupt, character timeout interrupt, etc.

By configuring the Interrupt Enable Register IER (0x24), the UART can support the above interrupts. The specific interrupt types and detailed introduction can be distinguished via IIR2 (0x0C) (Interrupt Identification Register 2). It is recommended to use IIR2 for interrupt operations, and not to use IIR.

48.2.8 DMA

UART supports transmit DMA requests and receive DMA requests.

The user configures the RX FIFO trigger threshold [RFIFOT](#) via the [FCR](#) register. When the data received by the RX FIFO exceeds the threshold, a DMA receive request is generated.

When the user configures [TIME_RXDMA_DIS](#) in the [FCR](#) register to 1, DMA moves data only when the trigger threshold is met; TIMEOUT does not move data.

When [TIME_RXDMA_DIS](#) is configured to 0, if the data in the RX FIFO does not reach the threshold and no new data arrives within 4 byte-times, a DMA receive request is also generated to send out the remaining data in the FIFO, preventing data smaller than the threshold from not being received.

The user configures the TX FIFO trigger threshold [TFIFOT](#) via the [FCR](#) register. When the pending data in

the TX FIFO is less than the threshold, a DMA transmit request is generated.

48.3 UART Register List

The UART register list is as follows:

UART0 base address: 0xF0040000

UART1 base address: 0xF0044000

UART2 base address: 0xF0048000

UART3 base address: 0xF004C000

UART4 base address: 0xF0050000

UART5 base address: 0xF0054000

UART6 base address: 0xF0058000

UART7 base address: 0xF005C000

PUART base address: 0xF4124000

Offset	Register Name	Description	Reset Value
0x0004	IDLE_CFG	Idle Configuration Register	0x00000000
0x0008	ADDR_CFG	address match config register	0x00000000
0x000C	IIR2	Interrupt Identification Register2	0x00000001
0x0010	CFG	Configuration Register	0x00000000
0x0014	OSCR	Over Sample Control Register	0x00000010
0x0018	FCRR	FIFO Control Register config	0x00000000
0x001C	MOTO_CFG	moto system control register	0x00000000
0x0020	RBR	Receiver Buffer Register (when DLAB = 0)	0x00000000
0x0020	THR	Transmitter Holding Register (when DLAB = 0)	0x00000000
0x0020	DLL	Divisor Latch LSB (when DLAB = 1)	0x00000001
0x0024	IER	Interrupt Enable Register (when DLAB = 0)	0x00000000
0x0024	DLM	Divisor Latch MSB (when DLAB = 1)	0x00000000
0x0028	IIR	Interrupt Identification Register	0x00000001
0x0028	FCR	FIFO Control Register	0x00000000
0x002C	LCR	Line Control Register	0x00000000
0x0030	MCR	Modem Control Register (	0x00000000
0x0034	LSR	Line Status Register	0x00000000
0x0038	MSR	Modem Status Register	0x00000000
0x003C	GPR	GPR Register	0x00000000

Offset	Register Name	Description	Reset Value

Table 192: UART Register List

48.4 UART Register Details

The detailed description of UART registers is as follows:

48.4.1 IDLE_CFG (0x4)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD						TX_IDLE_COND	TX_IDLE_EN	TX_IDLE_THR								RSVD				RXEN	RSVD	RX_IDLE_COND	RX_IDLE_EN	RX_IDLE_THR							
N/A						RW	RW	RW								N/A				RW	N/A	RW	RW	RW							
x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	x	x	x	x	0	x	0	0	0	0	0	0	0	0	0	0

IDLE_CFG [31:0]

Bits	Name	Description
25	TX_IDLE_COND	IDLE Detection Condition 0 - Treat as idle if TX pin is logic one 1 - Treat as idle if UART state machine state is idle
24	TX_IDLE_EN	UART TX Idle Detect Enable 0 - Disable 1 - Enable
23-16	TX_IDLE_THR	Threshold for UART transmit Idle detection (in terms of bits)
11	RXEN	UART receive enable. 0 - hold RX input to high, avoid wrong data input when config pinmux 1 - bypass RX input from PIN software should set it after config pinmux
9	RX_IDLE_COND	IDLE Detection Condition 0 - Treat as idle if RX pin is logic one 1 - Treat as idle if UART state machine state is idle
8	RX_IDLE_EN	UART Idle Detect Enable 0 - Disable 1 - Enable it should be enabled if enable address match feature
7-0	RX_IDLE_THR	Threshold for UART Receive Idle detection (in terms of bits)

IDLE_CFG Bits

48.4.2 ADDR_CFG (0x8)

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

48 UNIVERSAL ASYNCHRONOUS RECEIVER

TRANSMITTER (UART)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD											TXEN_9BIT	RXEN_ADDR_MSB	RXEN_9BIT	A1_EN	A0_EN	ADDR1								ADDR0							
N/A											RW	RW	RW	RW	RW	RW								RW							
x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

ADDR_CFG [31:0]

Bits	Name	Description
20	TXEN_9BIT	set to use 9bit mode for transmitter, will set the MSB for the first character as address flag, keep 0 for others.
19	RXEN_ADDR_MSB	set to use MSB as address flag at receiver(actually this is done by software set correct MSB in addr0/addr1). Clr to use first character as address. Only needed if enable address match feature
18	RXEN_9BIT	set to use 9bit mode for receiver, only valid if rxen_addr_msb is set
17	A1_EN	enable addr1 compare for the first character. If a1_en OR a0_en, then do not receive data if address not match. If ~a1_en AND ~a0_en, the receive all data like before. NOTE: should set idle_tmout_en if enable address match feature
16	A0_EN	enable addr0 compare for the first character
15-8	ADDR1	address 1 field. in 9bit mode, this is the full address byte. For other mode(8/7/6/5bit), MSB should be set for address flag. If want address==0 to be matched at 8bit mode, should set addr1=0x80
7-0	ADDR0	address 0 field.

ADDR_CFG Bits

48.4.3 IIR2 (0xC)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXIDLE_FLAG	TXIDLE_FLAG	ADDR_MATCH	ADDR_MATCH_IDLE	DATA_LOST	LSR_ERRF_STS	LSR_BREAK_STS	LSR_FRAMING_STS	LSR_PARITY_STS	LSR_OVERRUN_STS	RSVD										FIFOED				RSVD		INTRID					

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

48 UNIVERSAL ASYNCHRONOUS RECEIVER

TRANSMITTER (UART)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	N/A														RO		N/A		RO			
0	0	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	x	x	0	0	0	1

IIR2 [31:0]

Bits	Name	Description
31	RXIDLE_FLAG	UART RX IDLE Flag, assert after rxd high and then rx idle timeout, write one clear 0 - UART RX is busy 1 - UART RX is idle
30	TXIDLE_FLAG	UART TX IDLE Flag, assert after txd high and then tx idle timeout, write one clear 0 - UART TX is busy 1 - UART TX is idle
29	ADDR_MATCH	address match irq status, assert if either address match(and enabled). Write one clear NOTE: the address byte may not moved by DMA at this point. User can wait next addr_match_idle irq for the whole data include address
28	ADDR_MATCH_IDLE	address match and idle irq status, assert at rx bus idle if address match event triggered. Write one clear;
27	DATA_LOST	assert if data lost before address match status, write one clear; It will not assert if no address match occurs
26	LSR_ERRF_STS	Receiving FIFO error status, when occurs parity error、 framing error or break error,it will generate;write 1 clear to 0
25	LSR_BREAK_STS	Transmission break status, when occurs break error,it will generate;write 1 clear 0
24	LSR_FRAMING_STS	Framing error status,when occurs frame error,it will generate;write 1 clear 0
23	LSR_PARITY_STS	Parity error status,when occurs parity error ,it will generate;write 1 clear 0
22	LSR_OVERRUN_STS	Receive data overload status;when occurs overrun error ,it will generate;write 1 clear 0
7-6	FIFOED	FIFOs enabled These two bits are 1 when bit 0 of the FIFO Control Register (FIFOE) is set to 1.
3-0	INTRID	Interrupt ID, see IIR2 for detail decoding

IIR2 Bits

48.4.4 CFG (0x10)



HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

48 UNIVERSAL ASYNCHRONOUS RECEIVER

TRANSMITTER (UART)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																FIFOSIZE															
N/A																RO															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CFG [31:0]

Bits	Name	Description
1-0	FIFOSIZE	The depth of RXFIFO and TXFIFO 0: 16-byte FIFO 1: 32-byte FIFO 2: 64-byte FIFO 3: 128-byte FIFO

CFG Bits

48.4.5 OSCR (0x14)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																OSC															
N/A																RW															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	0	0	0	0

OSCR [31:0]

Bits	Name	Description
4-0	OSC	Over-sample control The value must be an even number; any odd value writes to this field will be converted to an even value. OSC=0: reserved OSC<=8: The over-sample ratio is 8 8 < OSC< 32: The over sample ratio is OSC

OSCR Bits

48.4.6 FCRR (0x18)

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

48 UNIVERSAL ASYNCHRONOUS RECEIVER

TRANSMITTER (UART)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																			
DMA_STOP_EN		DMA_STOPPED_WRITE_CLEAR		RSVD				BAUD_ACT_ENJ_PCLK		TMOUT_RXDMA_DIS		FIFOT4EN		RSVD		TFIFOT4				RSVD				RFIFOT4				RFIFOT		TFIFOT		DMAE		TFIFORST		RFIFORST		FIFOE												
RW	WO	N/A						RW	RW	RW	N/A		RW																			N/A				RW				RW		RW		RW	WO	WO	RW			
0	0	x	x					x	x	0	0	0	x	x	0																	0	0	0	0	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0

FCRR [31:0]

Bits	Name	Description
31	DMA_STOP_EN	enable dma stopped function
30	DMA_STOPPED_WRITE_CLEAR	dma stopped function by writing 1 clear 0
25	BAUD_ACT_ENJ_PCLK	Set the value of the configuration object.
24	TMOUT_RXDMA_DIS	set to disable timeout rx dma request
23	FIFOT4EN	set to use new 4bit fifo threshold(TFIFOT4 and RFIFOT4) clr to use 2bit(TFIFOT and RFIFOT)
20-16	TFIFOT4	txfifo threshold(0 for 1byte, 0xF for 32bytes), uart will send tx_dma_req when data in fifo is less than threshold.
12-8	RFIFOT4	rxfifo threshold(0 for 1byte, 0xF for 32bytes). Uart will send rx_dma_req if data in fifo reaches the threshold, also will set the rxdata irq if enabled
7-6	RFIFOT	Receiver FIFO trigger level
5-4	TFIFOT	Transmitter FIFO trigger level
3	DMAE	DMA enable 0: Disable 1: Enable
2	TFIFORST	Transmitter FIFO reset Write 1 to clear all bytes in the TXFIFO and resets its counter. The Transmitter Shift Register is not cleared. This bit will automatically be cleared.
1	RFIFORST	Receiver FIFO reset Write 1 to clear all bytes in the RXFIFO and resets its counter. The Receiver Shift Register is not cleared. This bit will automatically be cleared.

Bits	Name	Description
0	FIFOE	FIFO enable Write 1 to enable both the transmitter and receiver FIFOs. The FIFOs are reset when the value of this bit toggles.

FCRR Bits

48.4.7 MOTO_CFG (0x1C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWTRG	RSVD						TXSTOP_OPT	TXPRE_DLY								TXSTP_BITS								HWTRG_EN	TRG_MODE	TRG_CLR_RFIFO	TXSTOP_INSERT	RSVD			
WO	N/A						RW	RW								RW								RW	RW	RW	RW	N/A			
0	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x	x	x

MOTO_CFG [31:0]

Bits	Name	Description
31	SWTRG	software trigger. User should avoid use sw/hw trigger at same time, otherwise result unknown. Hardware auto reset.
24	TXSTOP_OPT	set to use baud_mx as txstop_bits counter, for control DE delay after STOP
23-16	TXPRE_DLY	if set to none-zero value, will assert DE before START bit, with the unit of clock dividered after dll/dlm
15-8	TXSTP_BITS	if TXSTOP_INSERT is enabled, the STOP bits to be inserted between each byte. 0 for 1 bit; 0xFF for 256bits
7	HWTRG_EN	set to enable hardware trigger(trigger from moto is shared by other UART)
6	TRG_MODE	set to enable trigger mode. software should push needed data into txbuffer frist, uart will not start transmission at this time. User should send trigger signal(by hw or sw), uart will send all data in txfifo till empty NOTE: the hw_trigger should be pulse signal from trig mux.
5	TRG_CLR_RFIFO	set to enable the feature that, clear rxfifo at tx trigger(sw or hw), avoid unexpected data in rxfifo.
4	TXSTOP_INSERT	set to insert STOP bits between each tx byte till tx fifo empty. NOTE: there will be no 1.5/2 STOP bits if enabled this feature, LCR.STB should be set to 0 if this bit is set

Bits	Name	Description

MOTO_CFG Bits

48.4.8 RBR (0x20)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
RSVD																					RX_BREAK_ERR	FRAMING_ERR	PARITY_ERR	RBR											
N/A																					RO	RO	RO	RO											
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0				

RBR [31:0]

Bits	Name	Description
10	RX_BREAK_ERR	can see if a break error occurs per byte of data(including intermediate state)
9	FRAMING_ERR	can see if a framing error occurs per byte of data(including intermediate state)
8	PARITY_ERR	can see if a parity error occurs per byte of data(including intermediate state)
7-0	RBR	Receive data read port

RBR Bits

48.4.9 THR (0x20)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																							THR								
N/A																							WO								
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	

THR [31:0]

Bits	Name	Description
7-0	THR	Transmit data write port

THR Bits

48.4.10 DLL (0x20)

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

48 UNIVERSAL ASYNCHRONOUS RECEIVER

TRANSMITTER (UART)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								DLL							
N/A																								RW							
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	1

DLL [31:0]

Bits	Name	Description
7-0	DLL	Least significant byte of the Divisor Latch

DLL Bits

48.4.11 IER (0x24)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERXIDLE	ETXIDLE	EADDRM	EADDRM_IDLE	EDATLOST	LSR_ERRF_IRQ_EN	LSR_BREAK_IRQ_EN	LSR_FRAMING_IRQ_EN	LSR_PARITY_IRQ_EN	LSR_OVERRUN_IRQ_EN	RSVD																EMSI	ELSI	ETHEI	ERBI		
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	N/A																RW	RW	RW	RW		
0	0	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	

IER [31:0]

Bits	Name	Description
31	ERXIDLE	Enable Receive Idle interrupt 0 - Disable Idle interrupt 1 - Enable Idle interrupt
30	ETXIDLE	enable transmit idle interrupt
29	EADDRM	enable ADDR_MATCH interrupt
28	EADDRM_IDLE	enable ADDR_MATCH_IDLE interrupt
27	EDATLOST	enable DATA_LOST interrupt
26	LSR_ERRF_IRQ_EN	enable transmission error interrupt
25	LSR_BREAK_IRQ_EN	enable break error interrupt
24	LSR_FRAMING_IRQ_EN	enable framing error interrupt
23	LSR_PARITY_IRQ_EN	enable parity error interrupt

TRANSMITTER (UART)

Bits	Name	Description
22	LSR_OVERRUN_I RQ_EN	enable data overload interrupt
3	EMSI	Enable modem status interrupt The interrupt asserts when the status of one of the following occurs: The status of modem_rin, modem_dcdn, modem_dsrn or modem_ctsn (If the auto-cts mode is disabled) has been changed. If the auto-cts mode is enabled (MCR bit4 (AFE) = 1), modem_ctsn would be used to control the transmitter.
2	ELSI	Enable receiver line status interrupt
1	ETHEI	Enable transmitter holding register interrupt
0	ERBI	Enable received data available interrupt and the character timeout interrupt 0: Disable 1: Enable

IER Bits

48.4.12 DLM (0x24)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																							DLM								
N/A																							RW								
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	

DLM [31:0]

Bits	Name	Description
7-0	DLM	Most significant byte of the Divisor Latch

DLM Bits

48.4.13 IIR (0x28)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXIDLE_FLAG	RSVD																							FIFOED	RSVD		INTRID				
W1C	N/A																							RO	N/A		RO				

TRANSMITTER (UART)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	x	x	0	0	0	1

IIR [31:0]

Bits	Name	Description
31	RXIDLE_FLAG	UART IDLE Flag 0 - UART is busy 1 - UART is idle NOTE: when write one to clear this bit, avoid changging FCR register since it's same address as IIR
7-6	FIFOED	FIFOs enabled These two bits are 1 when bit 0 of the FIFO Control Register (FIFOE) is set to 1.
3-0	INTRID	Interrupt ID, see IIR2 for detail decoding

IIR Bits

48.4.14 FCR (0x28)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																							RFIFOT	TFIFOT		DMAE	TFIFORST	RFIFORST	FIFOE		
N/A																							WO	WO		WO	WO	WO	WO		
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	

FCR [31:0]

Bits	Name	Description
7-6	RFIFOT	Receiver FIFO trigger level(0 for 1byte, 0x3 for 4bytes). Uart will send rx_dma_req if data in fifo reaches the threshold
5-4	TFIFOT	Transmitter FIFO trigger level(0 for 1byte, 0x3 for 4bytes), uart will send tx_dma_req when data in fifo is less than threshold.
3	DMAE	DMA enable 0: Disable 1: Enable
2	TFIFORST	Transmitter FIFO reset Write 1 to clear all bytes in the TXFIFO and resets its counter. The Transmitter Shift Register is not cleared. This bit will automatically be cleared.

TRANSMITTER (UART)

Bits	Name	Description
1	RFIFORST	Receiver FIFO reset Write 1 to clear all bytes in the RXFIFO and resets its counter. The Receiver Shift Register is not cleared. This bit will automatically be cleared.
0	FIFOE	FIFO enable Write 1 to enable both the transmitter and receiver FIFOs. The FIFOs are reset when the value of this bit toggles.

FCR Bits

48.4.15 LCR (0x2C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																							DLAB	BC	SPS	EPS	PEN	STB	WLS		
N/A																							RW	RW	RW	RW	RW	RW	RW		
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	

LCR [31:0]

Bits	Name	Description
7	DLAB	Divisor latch access bit
6	BC	Break control
5	SPS	Stick parity 1: Parity bit is constant 0 or 1, depending on bit4 (EPS). 0: Disable the sticky bit parity.
4	EPS	Even parity select 1: Even parity (an even number of logic-1 is in the data and parity bits) 0: Old parity.
3	PEN	Parity enable When this bit is set, a parity bit is generated in transmitted data before the first STOP bit and the parity bit would be checked for the received data.
2	STB	Number of STOP bits 0: 1 bits 1: The number of STOP bit is based on the WLS setting When WLS = 0, STOP bit is 1.5 bits When WLS = 1, 2, 3, STOP bit is 2 bits

TRANSMITTER (UART)

Bits	Name	Description
1-0	WLS	Word length setting 0: 5 bits 1: 6 bits 2: 7 bits 3: 8 bits

LCR Bits

48.4.16 MCR (0x30)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																										AFE	LOOP	RSVD	RTS	RSVD	
N/A																										RW	RW	N/A	RW	N/A	
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	x	x	0	x	

MCR [31:0]

Bits	Name	Description
5	AFE	Auto flow control enable 0: Disable 1: The auto-CTS and auto-RTS setting is based on the RTS bit setting: When RTS = 0, auto-CTS only When RTS = 1, auto-CTS and auto-RTS
4	LOOP	Enable loopback mode 0: Disable 1: Enable
1	RTS	Request to send This bit controls the modem_rtsn output. 0: The modem_rtsn output signal will be driven HIGH 1: The modem_rtsn output signal will be driven LOW

MCR Bits

48.4.17 LSR (0x34)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXIDLE	TXIDLE	RSVD										RFIFO_NUM				RSVD		TFIFO_NUM				ERRF	TEMT	THRE	LBREAK	FE	PE	OE	DR		
RO	RO	N/A										RO				N/A		RO				RO	RO	RO	RO	RO	RO	RO	RO		

TRANSMITTER (UART)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	x	x	x	x	x	x	x	x	x	0	0	0	0	0	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0

LSR [31:0]

Bits	Name	Description
31	RXIDLE	rxidle after timeout, clear after rx idle condition not match
30	TXIDLE	txidle after timeout, clear after tx idle condition not match
20-16	RFIFO_NUM	data bytes in rxfifo not read
12-8	TFIFO_NUM	data bytes in txfifo not sent
7	ERRF	Error in RXFIFO In the FIFO mode, this bit is set when there is at least one parity error, framing error, or line break associated with data in the RXFIFO. It is cleared when this register is read and there is no more error for the rest of data in the RXFIFO.
6	TEMT	Transmitter empty This bit is 1 when the THR (TXFIFO in the FIFO mode) and the Transmitter Shift Register (TSR) are both empty. Otherwise, it is zero.
5	THRE	Transmitter Holding Register empty This bit is 1 when the THR (TXFIFO in the FIFO mode) is empty. Otherwise, it is zero. If the THRE interrupt is enabled, an interrupt is triggered when THRE becomes 1.
4	LBREAK	Line break This bit is set when the uart_sin input signal was held LOW for longer than the time for a full-word transmission. A full-word transmission is the transmission of the START, data, parity, and STOP bits. It is cleared when this register is read. In the FIFO mode, this bit indicates the line break for the received data at the top of the RXFIFO.
3	FE	Framing error This bit is set when the received STOP bit is not HIGH. It is cleared when this register is read. In the FIFO mode, this bit indicates the framing error for the received data at the top of the RXFIFO.

Bits	Name	Description
2	PE	Parity error This bit is set when the received parity does not match with the parity selected in the LCR[5:4]. It is cleared when this register is read. In the FIFO mode, this bit indicates the parity error for the received data at the top of the RXFIFO.
1	OE	Overrun error This bit indicates that data in the Receiver Buffer Register (RBR) is overrun.
0	DR	Data ready. This bit is set when there are incoming received data in the Receiver Buffer Register (RBR). It is cleared when all of the received data are read.

LSR Bits

48.4.18 MSR (0x38)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																										CTS	RSVD		DCTS		
N/A																										RO	N/A		RC		
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	x	x	x	0	

MSR [31:0]

Bits	Name	Description
4	CTS	Clear to send 0: The modem_ctsn input signal is HIGH. 1: The modem_ctsn input signal is LOW.
0	DCTS	Delta clear to send This bit is set when the state of the modem_ctsn input signal has been changed since the last time this register is read.

MSR Bits

48.4.19 GPR (0x3C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								DATA							

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
N/A																								RW							
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	

GPR [31:0]

Bits	Name	Description
7-0	DATA	A one-byte storage register

GPR Bits

49 Serial Peripheral Interface (SPI) Bus

The Serial Peripheral Interface (SPI) supports both master mode and slave mode. In master mode, it can control various peripherals, while in slave mode, it can accept requests from the master to complete data exchange.

49.1 Module Functions

- Supports master mode and slave mode;
- Configurable MSB-first or LSB-first;
- Configurable clock frequency;
- Supports DMA data transfer;
- Supports DSPI and QSPI modes;
- Configurable clock polarity and phase;
- Tx_FIFO and Rx_FIFO depths are both 8 words;

49.2 Functional Description

This subsection is divided into four parts, describing master mode, slave mode, DSPI mode, and QSPI mode respectively. Master and slave modes primarily describe initialization and transmission formats. DSPI and QSPI modes mainly describe their differences from the normal mode.

The block diagram of this module is shown below. The SPI connects to external devices through six pins:

- SCLK: Serial Clock Line, generated by the master and provided to the slave device;
- CS: Slave Select Line, selects the slave device for master-slave data exchange;
- MOSI: Master Out Slave In line;
- MISO: Master In Slave Out line;
- DATA2: QSPI mode Data line 2;
- DATA3: QSPI mode Data line 3.

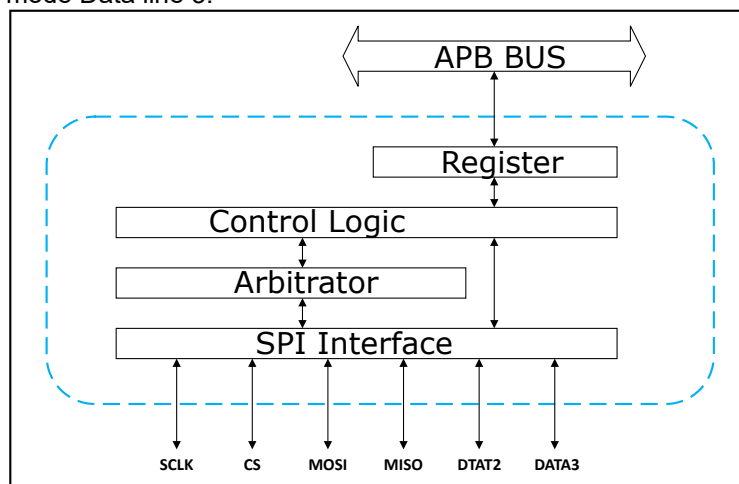


Figure 60: SPI Block Diagram

In normal mode, i.e., when TRANSFMT[MOSIBIDIR]=0, the MOSI direction is from master to slave, and the MISO direction is from slave to master. In master mode, the polarity and phase of SPI transmission can be set via TRANSFMT[CPOL] and TRANSFMT[CPHA]. Here, SPI sends data on the rising or falling edge of the clock

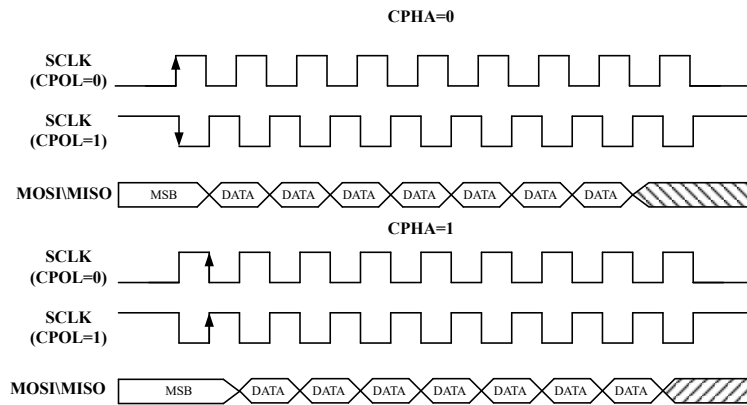


Figure 61: SPI Data Timing Diagram

line SCLK and receives data on the opposite active clock edge.

49.2.1 Master Mode

In master mode, the SPI initiates and controls transmissions. In master mode, the SPI transmission format, transmission control, and interface timing are configured via the configuration bus.

The transmission format includes command, address, and data fields. The command and address fields can also be disabled for pure data field transmission.

Interface timing includes SCLK frequency, minimum time from CS active to SCLK active edge, and minimum CS high time. In master mode, the TIMING register must be configured.

Transmission/reception in master mode can be done via register access, or DMA can be used for data transfer after setting thresholds in CTRL[TXTHRES] / CTRL[RXTHRES]. The SPI controller supports send/receive buffer threshold interrupts, reducing software overhead for flow control. The controller also supports interrupts upon transmission completion and other interrupts.

Furthermore, SPI allows software to directly control pin signals. This is configured in the DIRECTIO register to achieve special timing.

The SPI module supports up to 4 slave chip selects. The chip select signals are controlled by the [CS_EN] bit field in the CTRL control register, with each bit controlling one CS pin. Slave data should be checked against specific package information.

49.2.2 Slave Mode

When the SPI transmission format register TRANSFMT[SLVMODE]=1, SPI is configured into slave mode. In slave mode, there is no need to configure the SPI's interface timing register TIMING. Also, when SPI acts as a slave, the SCLK frequency must be less than or equal to 1/4 of the SPI clock frequency.

In slave mode, the controller receives signals from the interface and interprets them as command, dummy, and data. The command and dummy fields are fixed at 8 bits, and the length of the data segment is determined by the received command and register settings.

In slave mode, there are internally configured fixed commands that can be used directly. This mode also supports user-defined arbitrary 8-bit commands configured in registers. For user-defined instructions, the data segment format is configured by registers. For example, if the transmission mode is set to (dummy, write), the

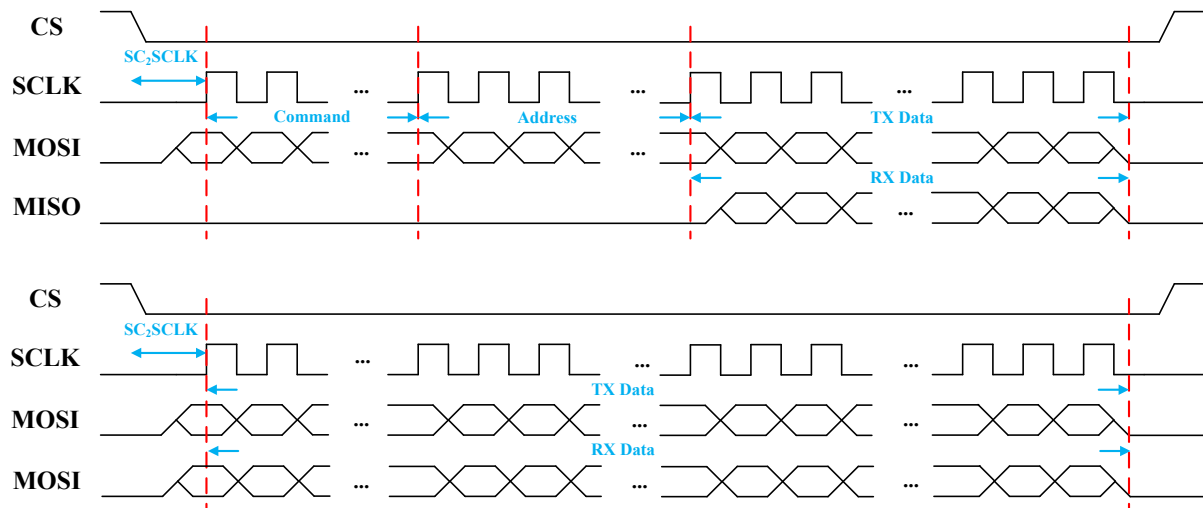


Figure 62: Two SPI Transmission Formats

dummy period is $(\text{dummyCnt}+1) \times ((\text{DataLen}+1)/\text{IO width})$. The dummy field is discarded, and subsequent data is saved into the data register.

49.2.3 DSPI Mode

When register `TRANSCTRL[DUALQUAD]=1`, SPI is configured into DSPI mode. In this mode, setting `TRANSFMT[MOSIBIDIR]=1` makes MOSI and MISO half-duplex, thereby doubling the bandwidth.

DSPI mode provides two transmission methods. One method transmits both address and data using the two lines (MOSI and MISO). In the other method, only data is transmitted via the two lines MOSI and MISO. The transmission method can be configured via the `TRANSCTRL` register.

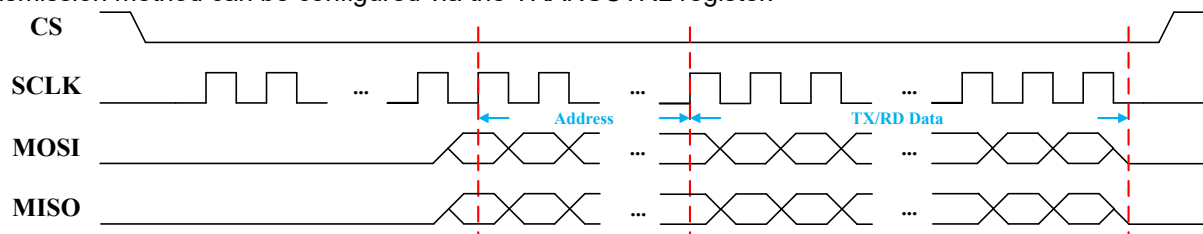


Figure 63: DSPI Address Two-Line Transmission Format Diagram

49.2.4 QSPI Mode

When register `TRANSCTRL[DUALQUAD]=2`, it is configured into QSPI mode. In this mode, setting `TRANSFMT[MOSIBIDIR]=1` makes MOSI, MISO, DATA2, and DATA3 half-duplex, thereby quadrupling the bandwidth.

QSPI mode provides two transmission methods. One method transmits both address and data using the four lines MOSI, MISO, DATA2, and DATA3. In the other method, only data is transmitted using the four lines MOSI, MISO, DATA2, and DATA3. The transmission method can be configured via the `TRANSCTRL` register.

49.2.5 Initiating Transmission and Command Values

When the SPI module is in master mode, transmission (read or write operation) begins after writing the corresponding instruction to the command register (CMD). For example, if the SPI module is set to master mode,

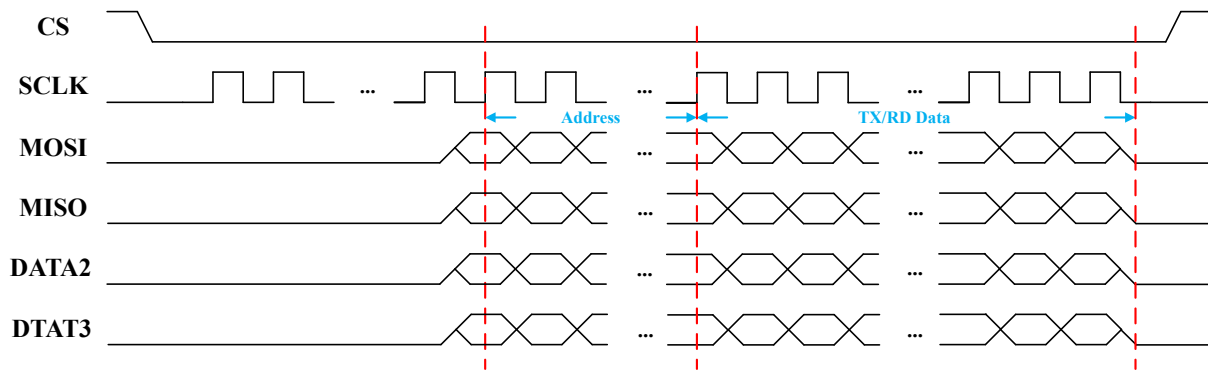


Figure 64: QSPI Address Four-Line Transmission Format Diagram

and both the command segment and address segment are disabled in the transmission control register (i.e., the master operates in pure data mode transmission), writing the value 0xFF to the command register initiates transmission (at this point, the SPI does not send out an instruction value). For instance, if a user configures the SPI module as master and uses DMA to send data from a specified address to a slave in pure data format, after DMA configuration, writing 0xFF to the command register causes the SPI module to fetch data via DMA and send it out. If the user needs to transmit again, they can simply write 0xFF to the command register once more.

If the command segment is enabled in the transmission control register when SPI is master, writing the corresponding command value to the command register initiates transmission, and the SPI module sends out the command value.

When the SPI module acts as a slave, it can support certain command operations. For example, upon receiving a command value of 0x05, the SPI module will return the slave status register (SLVST). Its return length is 32 bits. The following table lists the operations supported by SPI in slave mode.

Command Value	Operation Content
0x05	Read SPI slave status register value in SPI mode, 32 bits
0x15	Read SPI slave status register value in DSPI mode, 32 bits
0x25	Read SPI slave status register value in QSPI mode, 32 bits
0x0B	Read FIFO value in SPI slave in SPI mode
0x0C	Read FIFO value in SPI slave in DSPI mode
0x0E	Read FIFO value in SPI slave in QSPI mode
0x51	Store into FIFO in SPI slave in SPI mode
0x52	Store into FIFO in SPI slave in DSPI mode
0x54	Store into FIFO in SPI slave in QSPI mode

Table 193: Command Values Supported in SPI Slave Mode

49.2.6 FIFO Thresholds, Interrupts, and DMA

The Tx_FIFO and Rx_FIFO depths of the SPI module are both 8 words. When setting FIFO thresholds, note the FIFO depth. The FIFO depth can be obtained from the [TXFIFOSIZE] and [RXFIFOSIZE] bit fields in the CONFIG register. FIFO threshold settings affect the interrupt and DMA functions of the SPI module.

Users can enable the SPI module's interrupt function in the INTREN register as needed. The SPI module supports the following interrupts in master mode:

- Transmission Complete Interrupt;
- Transmit Tx_FIFO Threshold Interrupt (triggered when valid data in Tx_FIFO is less than or equal to the transmit threshold);
- Receive Rx_FIFO Threshold Interrupt (triggered when valid data in Rx_FIFO is greater than or equal to the receive threshold);

SPI supports the following interrupts in slave mode:

- Receive Command Interrupt (effective when command field is enabled);
- Transmit Tx_FIFO Threshold Interrupt (triggered when valid data in Tx_FIFO is less than or equal to the transmit threshold);
- Receive Rx_FIFO Threshold Interrupt (triggered when valid data in Rx_FIFO is greater than or equal to the receive threshold);
- Transmit Tx_FIFO Underflow Interrupt;
- Receive Rx_FIFO Overflow Interrupt;

The SPI module's DMA function is enabled by the [TXDMAEN] and [RXDMAEN] bits in the CTRL register. The FIFO thresholds that trigger DMA transfers in the SPI module are set in the [TXTHRES] and [RXTHRES] bit fields.

49.3 Brief SPI Configuration Instructions

49.3.1 Sending Command–Address–Data Fields Using DMA in Master Mode

1. Confirm that SPIACTIVE=0 in the SPI Status Register (0x034 STATUS) to ensure the previous SPI transfer is complete;
2. Configure the SPI Transmission Format Register (0x010 TRANSFMT) as needed;
3. Configure the SPI Transmission Control Register (0x020 TRANSCTRL) as needed;
4. Set the SPI transmit FIFO threshold in the Control Register (0x030 CTRL) as needed;
5. Configure interface timing via the SPI Timing Register (0x040 TIMING) as needed;
6. Configure interrupts as needed, e.g., interrupt upon transmission completion: set SPI Interrupt Enable Status Register (0x038 INTREN[ENDINTEN]=1);
7. Set up the DMA controller to ensure data moves from memory to the SPI Data Register (0x02c DATA);
8. Write the address to the SPI Address Register (0x028 ADDR) as needed;
9. Write the command to the SPI Command Register (0x024 CMD);
10. Wait for data completion to trigger interrupt, then read the SPI Interrupt Status Register (0x03c INTRST) to check status bits.

49.3.2 Pure Data Send Without DMA in Master Mode

1. Confirm that SPIACTIVE=0 in the SPI Status Register (0x034 STATUS) to ensure the previous SPI transfer is complete;
2. Configure the SPI Transmission Format Register (0x010 TRANSFMT) as needed;
3. Configure the SPI Transmission Control Register (0x020 TRANSCTRL) as needed: disable command and address fields by setting TRANSFMT[CMDEN]=0 and TRANSFMT[ADDREN]=0;
4. Configure interface timing via the SPI Timing Register (0x040 TIMING) as needed;
5. Configure interrupts as needed, e.g., interrupt upon transmission completion: set SPI Interrupt Enable Status Register (0x038 INTREN[ENDINTEN]=1);
6. Write data to the SPI Data Register (0x02c DATA);
7. Fill the Command Register CMD with 0xFF to initiate data transmission;
8. Wait for data completion to trigger interrupt, then read the SPI Interrupt Status Register (0x03c INTRST) to check status bits.

49.4 SPI Register List

SPI0 base address: 0xF0070000

SPI1 base address: 0xF0074000

SPI2 base address: 0xF0078000

SPI3 base address: 0xF007C000

Offset	Register Name	Description	Reset Value
0x0004	WR_TRANS_CNT	Transfer count for write data	0x00000000
0x0008	RD_TRANS_CNT	Transfer count for read data	0x00000000
0x0010	TRANSFMT	Transfer Format Register	0x00020780
0x0014	DIRECTIO	Direct IO Control Register	0x00003100
0x0020	TRANSCTRL	Transfer Control Register	0x00000000
0x0024	CMD	Command Register	0x00000000
0x0028	ADDR	Address Register	0x00000000
0x002C	DATA	Data Register	0x00000000
0x0030	CTRL	Control Register	0x00000000
0x0034	STATUS	Status Register	0x00000000
0x0038	INTREN	Interrupt Enable Register	0x00000000
0x003C	INTRST	Interrupt Status Register	0x00000000
0x0040	TIMING	Interface Timing Register	0x00000000
0x0060	SLVST	Slave Status Register	0x00000000
0x0064	SLVDATAcnt	Slave Data Count Register	0x00000000
0x0068	SLVDATAWCNT	WCnt	0x00000000
0x006C	SLVDATARCNT	RCnt	0x00000000
0x007C	CONFIG	Configuration Register	0x00004311

Offset	Register Name	Description	Reset Value

Table 194: SPI Register List

49.5 SPI Register Description

49.5.1 WR_TRANS_CNT (0x4)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WRTRANCNT																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

WR_TRANS_CNT [31:0]

Bits	Name	Description
31-0	WRTRANCNT	Transfer count for write data WrTranCnt indicates the number of units of data to be transmitted to the SPI bus from the Data Register. The actual transfer count is (WrTranCnt+1). WrTranCnt only takes effect when TransMode is 0, 1, 3, 4, 5, 6 or 8. The size (bit-width) of a data unit is defined by the DataLen field of the Transfer Format Register. For TransMode 0, WrTranCnt must be equal to RdTranCnt.

WR_TRANS_CNT Bits

49.5.2 RD_TRANS_CNT (0x8)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDTRANCNT																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RD_TRANS_CNT [31:0]

Bits	Name	Description
31-0	RDTRANCNT	Transfer count for read data RdTranCnt indicates the number of units of data to be received from SPI bus and stored to the Data Register. The actual received count is (RdTranCnt+1). RdTransCnt only takes effect when TransMode is 0, 2, 3, 4, 5, 6 or 9. The size (bit-width) of a data unit is defined by the DataLen field of the Transfer Format Register. For TransMode 0, WrTranCnt must equal RdTranCnt.

RD_TRANS_CNT Bits

49.5.3 TRANSFMT (0x10)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD														ADDRLEN		RSVD		DATALEN		DATAMERGE		RSVD		MOSIBIDIR		LSB		SLVMODE		CPOL	CPHA
N/A														RW		N/A		RW		RW		N/A		RW		RW		RW		RW	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	x	x	x	0	0	1	1	1	1	x	x	0	0	0	0	0

TRANSFMT [31:0]

Bits	Name	Description
17-16	ADDRLEN	Address length in bytes 0x0: 1 byte 0x1: 2 bytes 0x2: 3 bytes 0x3: 4 bytes
12-8	DATALEN	The length of each data unit in bits The actual bit number of a data unit is (DataLen + 1)
7	DATAMERGE	Enable Data Merge mode, which does automatic data split on write and data coalescing on read. This bit only takes effect when DataLen = 0x7. Under Data Merge mode, each write to the Data Register will transmit all fourbytes of the write data; each read from the Data Register will retrieve four bytes of received data as a single word data. When Data Merge mode is disabled, only the least (DataLen+1) significant bits of the Data Register are valid for read/write operations; no automatic data split/coalescing will be performed.

Bits	Name	Description
4	MOSIBIDIR	Bi-directional MOSI in regular (single) mode 0x0: MOSI is uni-directional signal in regular mode. 0x1: MOSI is bi-directional signal in regular mode. This bi-directional signal replaces the two
3	LSB	Transfer data with the least significant bit first 0x0: Most significant bit first 0x1: Least significant bit first
2	SLVMODE	SPI Master/Slave mode selection 0x0: Master mode 0x1: Slave mode
1	CPOL	SPI Clock Polarity 0x0: SCLK is LOW in the idle states 0x1: SCLK is HIGH in the idle states
0	CPHA	SPI Clock Phase 0x0: Sampling data at odd SCLK edges 0x1: Sampling data at even SCLK edges

TRANSFMT Bits

49.5.4 DIRECTIO (0x14)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD							DIRECTIOEN	RSVD		HOLD_OE	WP_OE	MISO_OE	MOSI_OE	SCLK_OE	CS_OE	RSVD		HOLD_O	WP_O	MISO_O	MOSI_O	SCLK_O	CS_O	RSVD		HOLD_I	WP_I	MISO_I	MOSI_I	SCLK_I	CS_I
							N/A							RW	N/A			RW	RW	RW	RW	RW	N/A			RW	RW	RW	RW	RW	N/A
x	x	x	x	x	x	x	0	x	x	0	0	0	0	0	0	x	x	1	1	0	0	0	1	x	x	0	0	0	0	0	0

DIRECTIO [31:0]

Bits	Name	Description
24	DIRECTIOEN	Enable Direct IO 0x0: Disable 0x1: Enable
21	HOLD_OE	Output enable for the SPI Flash hold signal
20	WP_OE	Output enable for the SPI Flash write protect signal
19	MISO_OE	Output enable fo the SPI MISO signal
18	MOSI_OE	Output enable for the SPI MOSI signal
17	SCLK_OE	Output enable for the SPI SCLK signal
16	CS_OE	Output enable for SPI CS (chip select) signal
13	HOLD_O	Output value for the SPI Flash hold signal

Bits	Name	Description
12	WP_O	Output value for the SPI Flash write protect signal
11	MISO_O	Output value for the SPI MISO signal
10	MOSI_O	Output value for the SPI MOSI signal
9	SCLK_O	Output value for the SPI SCLK signal
8	CS_O	Output value for the SPI CS (chip select) signal
5	HOLD_I	Status of the SPI Flash hold signal
4	WP_I	Status of the SPI Flash write protect signal
3	MISO_I	Status of the SPI MISO signal
2	MOSI_I	Status of the SPI MOSI signal
1	SCLK_I	Status of the SPI SCLK signal
0	CS_I	Status of the SPI CS (chip select) signal

DIRECTIO Bits

49.5.5 TRANSCTRL (0x20)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLVDATAONLY	CMDEN	ADDREN	ADDRFMT	TRANSMODE				DUALQUAD	TOKENEN	WRTRANCNT										TOKENVALUE	DUMMYCNT	RDTRANCNT									
RW	RW	RW	RW	RW				RW	RW	RW										RW	RW	RW									
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TRANSCTRL [31:0]

Bits	Name	Description
31	SLVDATAONLY	Data-only mode (slave mode only) 0x0: Disable the data-only mode 0x1: Enable the data-only mode Note: This mode only works in the uni-directional regular (single) mode so MOSIBiDir, DualQuad and TransMode should be set to 0.
30	CMDEN	SPI command phase enable (Master mode only) 0x0: Disable the command phase 0x1: Enable the command phase
29	ADDREN	SPI address phase enable (Master mode only) 0x0: Disable the address phase 0x1: Enable the address phase

Bits	Name	Description
28	ADDRFMT	SPI address phase format (Master mode only) 0x0: Address phase is the regular (single) mode 0x1: The format of the address phase is the same as the data phase (DualQuad).
27-24	TRANSMODE	Transfer mode The transfer sequence could be 0x0: Write and read at the same time 0x1: Write only 0x2: Read only 0x3: Write, Read 0x4: Read, Write 0x5: Write, Dummy, Read 0x6: Read, Dummy, Write 0x7: None Data (must enable CmdEn or AddrEn in master mode) 0x8: Dummy, Write 0x9: Dummy, Read 0xa~0xf: Reserved
23-22	DUALQUAD	SPI data phase format 0x0: Regular (Single) mode 0x1: Dual I/O mode 0x2: Quad I/O mode 0x3: Reserved
21	TOKENEN	Token transfer enable (Master mode only) Append a one-byte special token following the address phase for SPI read transfers. The value of the special token should be selected in TokenValue. 0x0: Disable the one-byte special token 0x1: Enable the one-byte special token
20-12	WRTRANCNT	Transfer count for write data WrTranCnt indicates the number of units of data to be transmitted to the SPI bus from the Data Register. The actual transfer count is (WrTranCnt+1). WrTranCnt only takes effect when TransMode is 0, 1, 3, 4, 5, 6 or 8. The size (bit-width) of a data unit is defined by the DataLen field of the Transfer Format Register. For TransMode 0, WrTranCnt must be equal to RdTranCnt.

Bits	Name	Description
11	TOKENVALUE	Token value (Master mode only) The value of the one-byte special token following the address phase for SPI read transfers. 0x0: token value = 0x00 0x1: token value = 0x69
10-9	DUMMYCNT	Dummy data count. The actual dummy count is (DummyCnt +1). The number of dummy cycles on the SPI interface will be (DummyCnt+1)* ((DataLen+1)/SPI IO width) The Data pins are put into the high impedance during the dummy data phase. DummyCnt is only used for TransMode 5, 6, 8 and 9, which has dummy data phases.
8-0	RDTRANCNT	Transfer count for read data RdTranCnt indicates the number of units of data to be received from SPI bus and stored to the Data Register. The actual received count is (RdTranCnt+1). RdTransCnt only takes effect when TransMode is 0, 2, 3, 4, 5, 6 or 9. The size (bit-width) of a data unit is defined by the DataLen field of the Transfer Format Register. For TransMode 0, WrTranCnt must equal RdTranCnt.

TRANSCTRL Bits

49.5.6 CMD (0x24)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								CMD							
N/A																								RW							
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	

CMD [31:0]

Bits	Name	Description
7-0	CMD	SPI Command

CMD Bits

49.5.7 ADDR (0x28)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ADDR [31:0]

Bits	Name	Description
31-0	ADDR	SPI Address (Master mode only)

ADDR Bits

49.5.8 DATA (0x2C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DATA [31:0]

Bits	Name	Description
31-0	DATA	Data to transmit or the received data For writes, data is enqueued to the TX FIFO. The least significant byte is always transmitted first. If the TX FIFO is full and the SPIActive bit of the status register is 1, the ready signal hready/pready will be deasserted to insert wait states to the transfer. For reads, data is read and dequeued from the RX FIFO. The least significant byte is the first received byte. If the RX FIFO is empty and the SPIActive bit of the status register is 1, the ready signal hready/pready will be deasserted to insert wait states to the transfer. The FIFOs decouple the speed of the SPI transfers and the software 欵樁 generation/consumption of data. When the TX FIFO is empty, SPI transfers will hold until more data is written to the TX FIFO; when the RX FIFO is full, SPI transfers will hold until there is more room in the RX FIFO. If more data is written to the TX FIFO than the write transfer count (WrTranCnt), the remaining data will stay in the TX FIFO for the next transfer or until the TX FIFO is reset.

DATA Bits

49.5.9 CTRL (0x30)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
RSVD			CMD_OP	CS_EN				TXTHRES									RXTHRES									RSVD			TXDMAEN	RXDMAEN	TXFIFORST	RXFIFORST	SPIRST
N/A			RW	RW				RW									RW									N/A			RW	RW	RW	RW	RW
x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x	x	0	0	0	0	0		

CTRL [31:0]

Bits	Name	Description
28	CMD_OP	Enable DMA tx req and TXFIFOINT work after to write CMD (Master mode only)
27-24	CS_EN	
23-16	TXTHRES	Transmit (TX) FIFO Threshold The TXFIFOInt interrupt or DMA request would be issued to replenish the TX FIFO when the TX data count is less than or equal to the TX FIFO threshold.

Bits	Name	Description
15-8	RXTHRES	Receive (RX) FIFO Threshold The RXFIFOInt interrupt or DMA request would be issued for consuming the RX FIFO when the RX data count is more than or equal to the RX FIFO threshold.
4	TXDMAEN	TX DMA enable
3	RXDMAEN	RX DMA enable
2	TXFIFORST	Transmit FIFO reset Write 1 to reset. It is automatically cleared to 0 after the reset operation completes.
1	RXFIFORST	Receive FIFO reset Write 1 to reset. It is automatically cleared to 0 after the reset operation completes.
0	SPIRST	SPI reset Write 1 to reset. It is automatically cleared to 0 after the reset operation completes.

CTRL Bits

49.5.10 STATUS (0x34)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD		TXNUM_7_6		RSVD		RXNUM_7_6		TXFULL	TXEMPTY							RXFULL	RXEMPTY														SPIACTIVE
N/A		RO		N/A		RO		RO	RO							RO	RO														RO
x	x	0	0	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x	x	x	x	x	0

STATUS [31:0]

Bits	Name	Description
29-28	TXNUM_7_6	Number of valid entries in the Transmit FIFO
25-24	RXNUM_7_6	Number of valid entries in the Receive FIFO
23	TXFULL	Transmit FIFO Full flag
22	TXEMPTY	Transmit FIFO Empty flag
21-16	TXNUM_5_0	Number of valid entries in the Transmit FIFO
15	RXFULL	Receive FIFO Full flag
14	RXEMPTY	Receive FIFO Empty flag
13-8	RXNUM_5_0	Number of valid entries in the Receive FIFO

Bits	Name	Description
0	SPIACTIVE	SPI register programming is in progress. In master mode, SPIActive becomes 1 after the SPI command register is written and becomes 0 after the transfer is finished. In slave mode, SPIActive becomes 1 after the SPI CS signal is asserted and becomes 0 after the SPI CS signal is deasserted. Note that due to clock synchronization, it may take at most two spi_clock cycles for SPIActive to change when the corresponding condition happens. Note this bit stays 0 when Direct IO Control or the memory-mapped interface is used.

STATUS Bits

49.5.11 INTREN (0x38)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								CS_NEGEN	CS_POSEN	SLVCMDEN	ENDINTEN	TXFIFOINTEN	RXFIFOINTEN	TXFIFOURINTEN	RXFIFOURINTEN
N/A																								RW	RW	RW	RW	RW	RW	RW	RW
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0

INTREN [31:0]

Bits	Name	Description
7	CS_NEGEN	Enable detected cs negedge edge interrupt (Slave mode only)
6	CS_POSEN	Enable detected cs posedge edge interrupt (Slave mode only)
5	SLVCMDEN	Enable the Slave Command Interrupt. Control whether interrupts are triggered whenever slave commands are received. (Slave mode only)
4	ENDINTEN	Enable the End of SPI Transfer interrupt. Control whether interrupts are triggered when SPI transfers end. (In slave mode, end of read status transaction doesn't trigger this interrupt.)
3	TXFIFOINTEN	Enable the SPI Transmit FIFO Threshold interrupt. Control whether interrupts are triggered when the valid entries are less than or equal to the TX FIFO threshold.

Bits	Name	Description
2	RXFIFOINTEN	Enable the SPI Receive FIFO Threshold interrupt. Control whether interrupts are triggered when the valid entries are greater than or equal to the RX FIFO threshold.
1	TXFIFOURINTEN	Enable the SPI Transmit FIFO Underrun interrupt. Control whether interrupts are triggered when the Transmit FIFO run out of data. (Slave mode only)
0	RXFIFOORINTEN	Enable the SPI Receive FIFO Overrun interrupt. Control whether interrupts are triggered when the Receive FIFO overflows. (Slave mode only)

INTREN Bits

49.5.12 INTRST (0x3C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																							CS_NEGINT	CS_POSINT	SLVCMDINT	ENDINT	TXFIFOINT	RXFIFOINT	TXFIFOURINT	RXFIFOORINT	
N/A																							W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	

INTRST [31:0]

Bits	Name	Description
7	CS_NEGINT	Cs posedge edge interrupt (Slave mode only)
6	CS_POSINT	Cs posedge edge interrupt (Slave mode only)
5	SLVCMDINT	Slave Command Interrupt. This bit is set when Slave Command interrupts occur. (Slave mode only)
4	ENDINT	End of SPI Transfer interrupt. This bit is set when End of SPI Transfer interrupts occur.
3	TXFIFOINT	TX FIFO Threshold interrupt. This bit is set when TX FIFO Threshold interrupts occur.
2	RXFIFOINT	RX FIFO Threshold interrupt. This bit is set when RX FIFO Threshold interrupts occur.

Bits	Name	Description
1	TXFIFOURINT	TX FIFO Underrun interrupt. This bit is set when TX FIFO Underrun interrupts occur. (Slave mode only)
0	RXFIFOORINT	RX FIFO Overrun interrupt. This bit is set when RX FIFO Overrun interrupts occur. (Slave mode only)

INTRST Bits

49.5.13 TIMING (0x40)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																		CS2SCLK	CSHT			SCLK_DIV									
N/A																		RW	RW			RW									
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	

TIMING [31:0]

Bits	Name	Description
13-12	CS2SCLK	The minimum time between the edges of SPI CS and the edges of SCLK. $SCLK\_period * (CS2SCLK + 1) / 2$
11-8	CSHT	The minimum time that SPI CS should stay HIGH. $SCLK\_period * (CSHT + 1) / 2$
7-0	SCLK_DIV	The clock frequency ratio between the clock source and SPI interface SCLK. $SCLK\_period = ((SCLK\_DIV + 1) * 2) * (\text{Period of the SPI clock source})$ The SCLK_DIV value 0xff is a special value which indicates that the SCLK frequency should be the same as the spi_clock frequency.

TIMING Bits

49.5.14 SLVST (0x60)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD													UNDERRUN	OVERRUN	READY	USR_STATUS															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
N/A													W1C	RW	RW	RW															
x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SLVST [31:0]

Bits	Name	Description
18	UNDERRUN	Data underrun occurs in the last transaction
17	OVERRUN	Data overrun occurs in the last transaction
16	READY	Set this bit to indicate that the ATCSPI200 is ready for data transaction. When an SPI transaction other than slave status-reading command ends, this bit will be cleared to 0.
15-0	USR_STATUS	User defined status flags

SLVST Bits

49.5.15 SLVDATAcnt (0x64)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD						WCNT										RSVD						RCNT									
N/A						RO										N/A						RO									
x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0

SLVDATAcnt [31:0]

Bits	Name	Description
25-16	WCNT	Slave transmitted data count
9-0	RCNT	Slave received data count

SLVDATAcnt Bits

49.5.16 SLVDATAWCNT (0x68)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VAL																															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SLVDATAWCNT [31:0]

Bits	Name	Description
31-0	VAL	

SLVDATAWCNT Bits

49.5.17 SLVDATARCNT (0x6C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VAL																															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SLVDATARCNT [31:0]

Bits	Name	Description
31-0	VAL	

SLVDATARCNT Bits

49.5.18 CONFIG (0x7C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																	SLAVE	RSVD				QUADSPI	DUALSPI	TXFIFOSIZE				RXFIFOSIZE			
N/A																	RO	N/A				RO	RO	RO				RO			
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	x	x	x	x	1	1	0	0	0	1	0	0	0	1

CONFIG [31:0]

Bits	Name	Description
14	SLAVE	Support for SPI Slave mode
9	QUADSPI	Support for Quad I/O SPI
8	DUALSPI	Support for Dual I/O SPI
7-4	TXFIFOSIZE	Depth of TX FIFO 0x0: 2 words 0x1: 4 words 0x2: 8 words 0x3: 16 words 0x4: 32 words 0x5: 64 words 0x6: 128 words

Bits	Name	Description
3-0	RXFIFOSIZE	Depth of RX FIFO 0x0: 2 words 0x1: 4 words 0x2: 8 words 0x3: 16 words 0x4: 32 words 0x5: 64 words 0x6: 128 words

CONFIG Bits

50 Integrated Circuit Bus I2C

This chapter describes the functions and features of I2C.

50.1 Feature Summary

This chapter describes the main features of I2C:

- supports standard mode (100Kb/s), fast mode (400Kb/s) and fast mode + (1Mb/s)
- Configurable master-slave mode
- supports 7-bit and 10-bit address modes
- supports general call address
- Automatic clock stretching
- Configurable clock/data timing
- supports direct memory access (DMA)
- 4-byte FIFO

I2C is shown as [Figure 65](#).

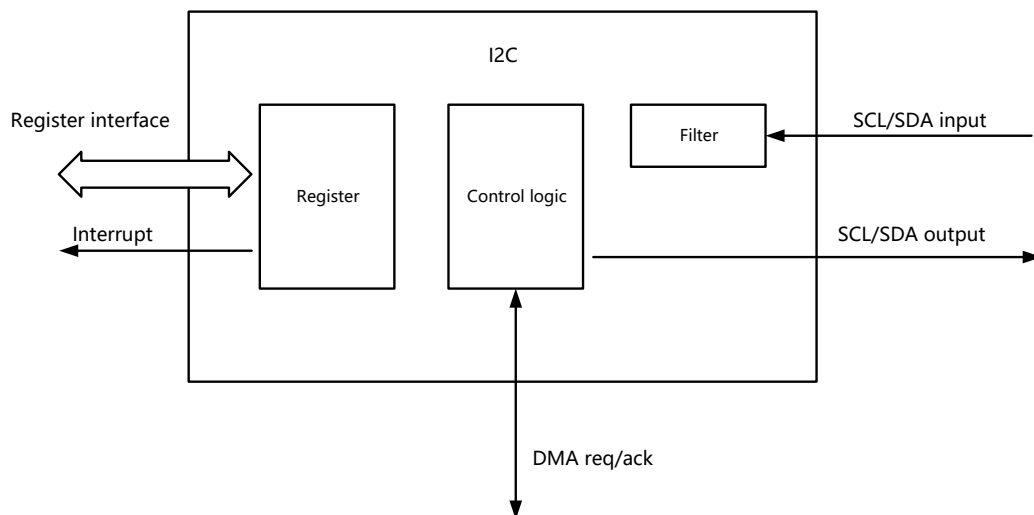


Figure 65: I2C Block Diagram

50.2 Function Description

This section describes the functions of I2C.

50.2.1 Main functions

Use the MASTER bit in the SETUP register to configure I2C to operate in master mode or slave mode. As an I2C master, the controller can efficiently initiate transfers. Each transfer consists of 4 phases: start, address, data and end. A START operation occurs in the beginning stage, an address is sent in the address stage, one or more data bytes are transferred in the data stage, and a STOP operation occurs in the end stage. Each stage can be independently controlled whether it is executed.

acts as an I2C slave. When the address transmitted by I2C matches the address of register ADDR, the controller is selected. The ADDRHIT bit in the INTEN register can be configured to cause the controller to

generate an interrupt to inform the software to prepare for subsequent operations. If a broadcast call address is received, the controller responds with an ACK and sets the GENCALL bit in the STATUS register to 1.

When the software is not ready to send or receive the next byte of data and the FIFO is full, the control will automatically extend the I2C bus clock to suspend bus transmission. Automatic clock extension is supported in both master and slave modes.

The controller enables automatic response by default, that is, an ACK will be automatically issued for every byte of data received except the last byte. The software can disable the automatic response function by enabling the ByteReceive interrupt, and decide whether to send an ACK response after each byte is received.

50.2.2 Timing Configuration

I2C timing parameters are configured in the SETUP register, and the units of all parameters are the number of clock cycles of the I2C module functional clock (CLK_TOP_I2Cx). Due to the limited number of bits of these configurable parameters, when the functional clock frequency is very high, the bit width of the timing parameters may not meet the configuration requirements. Therefore, the controller provides a TPM register to set the timing parameter multiplier, which can expand the range of timing parameters. Modifications to the TPM register should be made when the I2C bus is idle and the I2C controller is turned off (the IICEN bit of Register SETUP is 0).

The following describes the timing parameter configuration using the CLK_TOP_I2C frequency of 40MHz and TPM equal to 0.

Glitch Filtering: The pulse width that can be filtered out on the SCL and SDA input signals is defined by the T_SP bit of the SETUP register. The calculation formula is as follows: (50.2.2) :

$$PulseWidth = T\_SP \times Clk\_Period \times (TPM + 1)$$

(6)

For fast mode and fast mode +, glitches below 50ns are required to be filtered out, and the calculated T_SP should be 2.

Data Setup Time: The data setup time defines the time for which SDA should remain stable before the rising edge of SCL. It is set by the T_SUDAT bit of the SETUP register. The calculation formula is as follows: (50.2.2) :

$$SetupTime = (2 \times Clk\_Period) + (2 + T\_SP + T\_SUDAT) \times Clk\_Period \times (TPM + 1)$$

(7)

For standard mode, when the data setup time requirement is 250ns, the available T_SUDAT should be set to 4.

Data Hold Time: The data hold time defines the time that should remain stable after the falling edge of SCL. It is set by the T_HDDAT bit of the SETUP register. The calculation formula is as follows: (50.2.2) :

$$HoldTime = (2 \times Clk\_Period) + (2 + T\_SP + T\_HDDAT) \times Clk\_Period \times (TPM + 1)$$

(8)

For standard mode, when the data hold time requirement is 300ns, the available T_HDDAT should be set to 6.

bus clock frequency: The I2C bus clock frequency is defined by the T_SCLHI bit and T_SCLRATIO bit in the SETUP register, and the calculation formulas are (50.2.2) and (50.2.2)

$$SCL\ HighPeriod = (2 \times Clk\_Period) + (2 + T\_SP + T\_SCLHI) \times Clk\_Period \times (TPM + 1)$$

(9)

$$SCL\ LowPeriod = (2 \times Clk\_Period) + (2 + T\_SP + T\_SCLHI \times T\_SCLRATIO) \times Clk\_Period \times (TPM + 1)$$

(10)

50.2.3 Master Mode

Data transmission without DMA:

1. Set timing parameters
2. Configure master mode MASTER=1 and enable controller IICEN=1
3. Set the data length DATAcnt, transfer direction DIR and transfer phase selection PHASE_START/ ADDR/ DATA/ STOP
4. Configure destination slave address ADDRESS
5. Set CMPL to generate an interrupt when transfer is completed, and set FIFOEMPTY to generate an interrupt when FIFO is empty
6. Write 0x1 to the COMMAND register to initiate the transfer
7. Wait for interrupt: If the FIFO is empty, write the DATA register to store the data. If all data has been stored in the FIFO, turn off the FIFO empty interrupt. Otherwise, repeat Step 7; If the transfer is completed, check ADDR_HIT to ensure that the target slave received the transfer correctly, write 1 to the CMPL register to clear the interrupt, and jump to step 8.
8. Turn off all interrupts and check DATAcnt to ensure that all data transfers are complete

Data reception without DMA:

1. Set timing parameters
2. Configure master mode MASTER=1 and enable controller IICEN=1
3. Set the data length DATAcnt, transfer direction DIR and transfer phase selection PHASE_START/ ADDR/ DATA/ STOP
4. Configure target slave address ADDRESS
5. Set CMPL to generate an interrupt when the transfer is completed, and set FIFOFULL to generate an interrupt when the FIFO is full

6. Write 0x1 to the COMMAND register to initiate the transfer
7. Wait for interrupt: If the FIFO is full, read data from the DATA register; if all data has been stored in the FIFO, turn off the FIFO full interrupt; otherwise, repeat Step 7; If the transfer is completed, check ADDR_HIT to ensure that the target slave received the transfer correctly, write 1 to the CMPL register to clear the interrupt, and jump to step 8.
8. Turn off all interrupts and check DATA_CNT to ensure that all data transfers are complete

Data transmission using DMA:

1. Set timing parameters
2. Configure master mode MASTER=1, enable controller IICEN=1, enable DMA function DMAEn=1
3. Configure DMA controller IP
4. Set the data length DATA_CNT, transfer direction DIR and transfer phase selection PHASE_START/ ADDR/ DATA/ STOP
5. Configure target slave address ADDRESS
6. Set CMPL to generate an interrupt when the transfer is completed
7. Write 0x1 to the COMMAND register to initiate the transfer
8. Wait for an interrupt, check DATA_CNT to ensure all data transfers are complete

50.2.4 Slave mode

Data transfer without DMA:

1. Set slave address
2. Set timing parameters
3. Configure slave mode MASTER=0 and enable controller IICEN=1
4. Enables address hit interrupt ADDR_HIT and transfer complete interrupt CMPL
5. Waiting for address hit interrupt: Read the transmission direction DIR and check whether GENCALL confirms that it is a broadcast call. If the DIR direction is slave reception, enable FIFO full interrupt and jump to step 6; if the DIR direction is slave transmission, enable FIFO empty interrupt and jump to step 7
6. Slave receive: Wait for the FIFO full interrupt, read data until receiving the transfer completion interrupt
7. Slave send: Wait for FIFO empty interruption, write data until receiving transmission completion interrupt
8. Check the DATA_CNT register to confirm the amount of transferred data and clear the transfer completion interrupt

Data transfer using DMA:

1. Set slave address
2. Set timing parameters
3. Configure slave mode MASTER=0 and enable controller IICEN=1
4. Enables address hit interrupt ADDR_HIT and transfer complete interrupt CMPL
5. Wait for address hit interrupt: Read the transfer direction DIR, and if it is not the expected transfer direction, jump to the data transfer process without using DMA. Check GENCALL to confirm if it is a broadcast call
6. Slave receive: Wait for the FIFO full interrupt, read data until the transmission completion interrupt is

received

7. Slave send: Wait for FIFO empty interruption, write data until receiving transmission completion interrupt
8. Check the DATACNT register to confirm the amount of transferred data and clear the transfer completion interrupt

50.2.5 interrupt

The types of interrupts supported by I2C include: transfer completion interrupt, byte receive interrupt, byte send interrupt, START interrupt, STOP interrupt, bus arbitration loss interrupt, address hit interrupt, FIFO half empty/half full interrupt, FIFO full interrupt, FIFO empty interrupt, etc.

By configuring the interrupt enable register INTEN (0x14), I2C can support the above interrupts. The specific interrupt types and detailed introduction can be distinguished by the STATUS (0x18) register.

50.2.6 DMA

I2C supports DMA functions and supports DMA operations in master mode and slave mode. For the required DMA configuration, there is a detailed description of using DMA for data transfer in the introduction of master mode and slave mode in this chapter.

50.3 I2C register

50.3.1 register description

I2C is as follows:

I2C0 base address: 0xF0060000

I2C1 base address: 0xF0064000

I2C2 base address: 0xF0068000

I2C3 base address: 0xF006C000

Offset	Register Name	Description	Reset Value
0x0010	CFG	Configuration Register	0x00000001
0x0014	INTEN	Interrupt Enable Register	0x00000000
0x0018	STATUS	Status Register	0x00000001
0x001C	ADDR	Address Register	0x00000000
0x0020	DATA	Data Register	0x00000000
0x0024	CTRL	Control Register	0x00905E00
0x0028	CMD	Command Register	0x00000000
0x002C	SETUP	Setup Register	0x05252100
0x0030	TPM	I2C Timing Paramater Multiplier	0x00000000

Table 195: I2C Register List

50.3.2 register details

I2C is as follows:



50.3.3 CFG (0x10)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																FIFOSIZE															
N/A																RO															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

CFG [31:0]

Bits	Name	Description
1-0	FIFOSIZE	FIFO Size: 0: 2 bytes 1: 4 bytes 2: 8 bytes 3: 16 bytes

CFG Bits

50.3.4 INTEN (0x14)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																						CMPL	BYTERECV	BYTETRANS	START	STOP	ARBLOSE	ADDRHIT	FIFOHALF	FIFOFULL	FIFOEMPTY
N/A																						RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

INTEN [31:0]

Bits	Name	Description
9	CMPL	Set to enable the Completion Interrupt. Master: interrupts when a transaction is issued from this master and completed without losing the bus arbitration. Slave: interrupts when a transaction addressing the controller is completed.
8	BYTERECV	Set to enable the Byte Receive Interrupt. Interrupts when a byte of data is received Auto-ACK will be disabled if this interrupt is enabled, that is, the software needs to ACK/NACK the received byte manually.
7	BYTETRANS	Set to enable the Byte Transmit Interrupt. Interrupts when a byte of data is transmitted.

Bits	Name	Description
6	START	Set to enable the START Condition Interrupt. Interrupts when a START condition/repeated START condition is detected.
5	STOP	Set to enable the STOP Condition Interrupt Interrupts when a STOP condition is detected.
4	ARBLOSE	Set to enable the Arbitration Lose Interrupt. Master: interrupts when the controller loses the bus arbitration Slave: not available in this mode.
3	ADDRHIT	Set to enable the Address Hit Interrupt. Master: interrupts when the addressed slave returned an ACK. Slave: interrupts when the controller is addressed.
2	FIFOHALF	Set to enable the FIFO Half Interrupt. Receiver: Interrupts when the FIFO is half-empty, i.e, there is $\geq 1/2$ entries in the FIFO. Transmitter: Interrupts when the FIFO is half-empty, i.e. there is $\leq 1/2$ entries in the FIFO. This interrupt depends on the transaction direction; don't enable this interrupt unless the transfer direction is determined, otherwise unintended interrupts may be triggered.
1	FIFOFULL	Set to enable the FIFO Full Interrupt. Interrupts when the FIFO is full.
0	FIFOEMPTY	Set to enabled the FIFO Empty Interrupt Interrupts when the FIFO is empty.

INTEN Bits

50.3.5 STATUS (0x18)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																	LINESDA	LINESCL	GENCALL	BUSBUSY	ACK	CMPL	BYTERECV	BYTETRANS	START	STOP	ARBLOSE	ADDRHIT	FIFOHALF	FIFOFULL	FIFOEMPTY
N/A																	RO	RO	RO	RO	RO	W1C	W1C	W1C	W1C	W1C	W1C	W1C	RO	RO	RO
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

STATUS [31:0]

Bits	Name	Description
14	LINESDA	Indicates the current status of the SDA line on the bus 1: high 0: low

Bits	Name	Description
13	LINESCL	Indicates the current status of the SCL line on the bus 1: high 0: low
12	GENCALL	Indicates that the address of the current transaction is a general call address: 1: General call 0: Not general call
11	BUSBUSY	Indicates that the bus is busy The bus is busy when a START condition is on bus and it ends when a STOP condition is seen on bus 1: Busy 0: Not busy
10	ACK	Indicates the type of the last received/transmitted acknowledgement bit: 1: ACK 0: NACK
9	CMPL	Transaction Completion Master: Indicates that a transaction has been issued from this master and completed without losing the bus arbitration Slave: Indicates that a transaction addressing the controller has been completed. This status bit must be cleared to receive the next transaction; otherwise, the next incoming transaction will be blocked.
8	BYTERECV	Indicates that a byte of data has been received.
7	BYTETRANS	Indicates that a byte of data has been transmitted.
6	START	Indicates that a START Condition or a repeated START condition has been transmitted/received.
5	STOP	Indicates that a STOP Condition has been transmitted/received.
4	ARBLOSE	Indicates that the controller has lost the bus arbitration.
3	ADDRHIT	Master: indicates that a slave has responded to the transaction. Slave: indicates that a transaction is targeting the controller (including the General Call).
2	FIFOHALF	Transmitter: Indicates that the FIFO is half-empty.
1	FIFOFULL	Indicates that the FIFO is full.
0	FIFOEMPTY	Indicates that the FIFO is empty.

STATUS Bits

50.3.6 ADDR (0x1C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					ADDR										
N/A																					RW										
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

ADDR [31:0]

Bits	Name	Description
9-0	ADDR	The slave address. For 7-bit addressing mode, the most significant 3 bits are ignored and only the least-significant 7 bits of Addr are valid

ADDR Bits

50.3.7 DATA (0x20)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
RSVD																												DATA							
N/A																												RW							
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					

DATA [31:0]

Bits	Name	Description
7-0	DATA	Write this register to put one byte of data to the FIFO. Read this register to get one byte of data from the FIFO.

DATA Bits

50.3.8 CTRL (0x24)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATAcnt_HIGH								RESET_LEN				RSVD				RESET_HOLD_SCKIN	RESET_ON	PHASE_START	PHASE_ADDR	PHASE_DATA	PHASE_STOP	DIR	DATAcnt								
RW								RW				N/A				RW	RW	RW	RW	RW	RW	RW	RW								
0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	1	1	1	1	1	0	0	0	0	0	0	0	0

CTRL [31:0]

Bits	Name	Description
31-24	DATA_CNT_HIGH	Data counts in bytes. Master: The number of bytes to transmit/receive. 0 means max length. DataCnt will be decreased by one for each byte transmitted/received. Slave: the meaning of DataCnt depends on the DMA mode: If DMA is not enabled, DataCnt is the number of bytes transmitted/received from the bus master. It is reset to 0 when the controller is addressed and then increased by one for each byte of data transmitted/received. If DMA is enabled, DataCnt is the number of bytes to transmit/receive. It will not be reset to 0 when the slave is addressed and it will be decreased by one for each byte of data transmitted/received.
23-20	RESET_LEN	reset clock cycles. the clock high/low time is defined by Setup.T_SCLHi, 50% duty cycle.
14	RESET_HOLD_SCKIN	set to hold input clock to high when reset is active
13	RESET_ON	set to send reset signals(just toggle clock bus defined by reset_len). this register is cleared when reset is end, can't be cleared by software
12	PHASE_START	Enable this bit to send a START condition at the beginning of transaction. Master mode only.
11	PHASE_ADDR	Enable this bit to send the address after START condition. Master mode only.
10	PHASE_DATA	Enable this bit to send the data after Address phase. Master mode only.
9	PHASE_STOP	Enable this bit to send a STOP condition at the end of a transaction. Master mode only.
8	DIR	Transaction direction Master: Set this bit to determine the direction for the next transaction. 0: Transmitter 1: Receiver Slave: The direction of the last received transaction. 0: Receiver 1: Transmitter

Bits	Name	Description
7-0	DATA_CNT	Data counts in bytes. Master: The number of bytes to transmit/receive. 0 means max length. DataCnt will be decreased by one for each byte transmitted/received. Slave: the meaning of DataCnt depends on the DMA mode: If DMA is not enabled, DataCnt is the number of bytes transmitted/received from the bus master. It is reset to 0 when the controller is addressed and then increased by one for each byte of data transmitted/received. If DMA is enabled, DataCnt is the number of bytes to transmit/receive. It will not be reset to 0 when the slave is addressed and it will be decreased by one for each byte of data transmitted/received.

CTRL Bits

50.3.9 CMD (0x28)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																												CMD			
N/A																												RW			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

CMD [31:0]

Bits	Name	Description
2-0	CMD	Write this register with the following values to perform the corresponding actions: 0x0: no action 0x1: issue a data transaction (Master only) 0x2: respond with an ACK to the received byte 0x3: respond with a NACK to the received byte 0x4: clear the FIFO 0x5: reset the I2C controller (abort current transaction, set the SDA and SCL line to the open-drain mode, reset the Status Register and the Interrupt Enable Register, and empty the FIFO) When issuing a data transaction by writing 0x1 to this register, the CMD field stays at 0x1 for the duration of the entire transaction, and it is only cleared to 0x0 after when the transaction has completed or when the controller loses the arbitration. Note: No transaction will be issued by the controller when all phases (Start, Address, Data and Stop) are disabled.

CMD Bits

50.3.10 SETUP (0x2C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD			T_SUDAT					T_SP			T_HDDAT					RSVD		T_SCLRADIO	T_SCLHI								DMAEN	MASTER	ADDRESSING	IICEN	
N/A			RW					RW			RW					N/A		RW	RW								RW	RW	RW	RW	
0	0	0	0	0	1	0	1	0	0	1	0	0	1	0	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0

SETUP [31:0]

Bits	Name	Description
28-24	T_SUDAT	T_SUDAT defines the data setup time before releasing the SCL. Setup time = $(2 * tpclk) + (2 + T\_SP + T\_SUDAT) * tpclk * (TPM+1)$ tpclk = PCLK period TPM = The multiplier value in Timing Parameter Multiplier Register
23-21	T_SP	T_SP defines the pulse width of spikes that must be suppressed by the input filter. Pulse width = $T\_SP * tpclk * (TPM+1)$

Bits	Name	Description
20-16	T_HDDAT	T_HDDAT defines the data hold time after SCL goes LOW Hold time = $(2 * tpclk) + (2 + T\_SP + T\_HDDAT) * tpclk * (TPM+1)$
13	T_SCLRADIO	The LOW period of the generated SCL clock is defined by the combination of T_SCLRATIO and T_SCLHi values. When T_SCLRATIO = 0, the LOW period is equal to HIGH period. When T_SCLRATIO = 1, the LOW period is roughly two times of HIGH period. SCL LOW period = $(2 * tpclk) + (2 + T\_SP + T\_SCLHi * ratio) * tpclk * (TPM+1)$ 1: ratio = 2 0: ratio = 1 This field is only valid when the controller is in the master mode.
12-4	T_SCLHI	The HIGH period of generated SCL clock is defined by T_SCLHi. SCL HIGH period = $(2 * tpclk) + (2 + T\_SP + T\_SCLHi) * tpclk * (TPM+1)$ The T_SCLHi value must be greater than T_SP and T_HDDAT values. This field is only valid when the controller is in the master mode.
3	DMAEN	Enable the direct memory access mode data transfer. 1: Enable 0: Disable
2	MASTER	Configure this device as a master or a slave. 1: Master mode 0: Slave mode
1	ADDRESSING	I2C addressing mode: 1: 10-bit addressing mode 0: 7-bit addressing mode
0	IICEN	Enable the I2C controller. 1: Enable 0: Disable

SETUP Bits

50.3.11 TPM (0x30)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																									TPM						
N/A																									RW						
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

TPM [31:0]

Bits	Name	Description
4-0	TPM	A multiplication value for I2C timing parameters. All the timing parameters in the Setup Register are multiplied by (TPM+1).

TPM Bits

51 Controller Area Network MCAN

This chapter introduces the functionality and features of the Controller Area Network (MCAN).

51.1 Overview

Features of the Controller Area Network (MCAN) are as follows:

- Supports CAN 2.0B protocol, with up to 8-byte data payload, data rate up to 1Mbit/s;
- Supports CAN FD protocol, with up to 64-byte data payload, data rate up to 8Mbit/s;
- Supports 1~1/16 baud rate prescaler, combined with system CAN clock divider, allows flexible baud rate configuration;
- Multiple MCAN modules share HRAM (AHB SRAM, 32KB) as message memory
 - CPU can access HRAM via AHB; different MCAN modules share the same HRAM by configuring message RAM to different addresses (lower 16-bit address is sufficient)
 - Can be flexibly configured as different transmit, receive buffers, filters, and event buffers.
- Programmable ID CODE and MASK bits;
- PTB/STB both support single shot transmission mode;
- Supports silent mode;
- Supports loopback mode;
- Supports standby mode;
- Supports capturing transmission error types and locating arbitration failure positions;
- Programmable error warning value;
- Compliant with ISO 11898-1:2015
- Supports AUTOSAR
- Supports SAE J1939
- Supports debugging
- Supports DMA
- Supports hardware timestamp according to CiA 603
 - 16 sets of 32-bit timestamps, or 8 sets of 64-bit timestamps
 - TSU can use its own timestamp or an external timestamp
 - Can output timestamps for use by other system MCAN modules

51.2 CAN Architecture Diagram

The structure diagram of this module is as shown in [Figure 66](#). The functions of each main unit are as follows,

Transmit Processing Unit: Controls the transmission from message memory to the CAN core. It can be configured to have a maximum of 32 transmit buffers.

Receive Processing Unit: Controls which matching messages can be received. It can be configured to have a maximum of 64 receive buffers.

CAN Core: The CAN core controls the transmit and receive shift registers, handles all ISO 11898-1:2015 protocol functions, and supports 11-bit and 29-bit identifiers.

Interrupt Control: Handles the generation and clearing of interrupts.

Message Memory Interface and Message Memory: Stores transmit and receive messages.

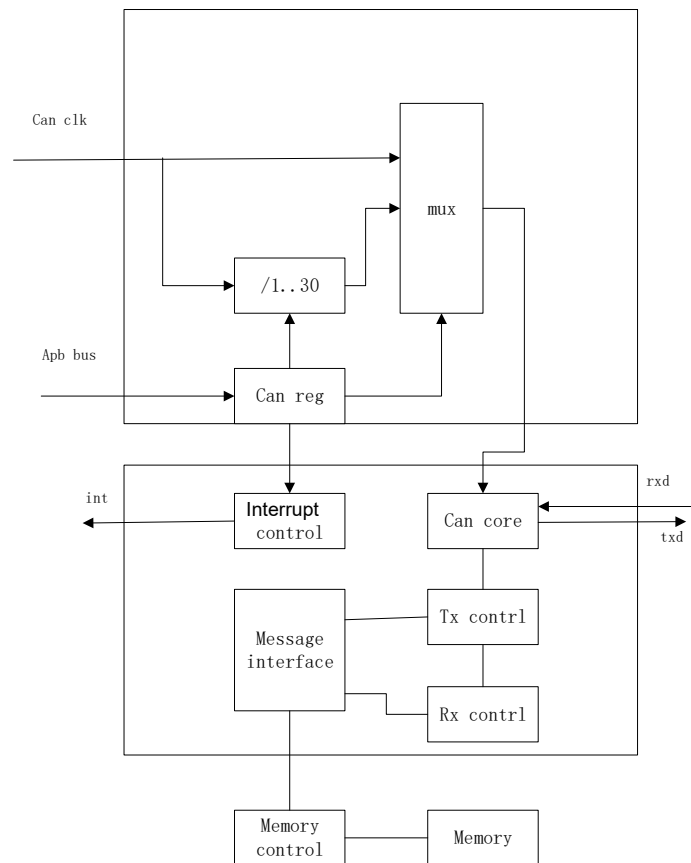


Figure 66: CAN Structure Block Diagram

51.3 Pin Description

Table 196: CAN Pin Description

Pin	Direction	Function Description
CANn ¹ _TXD	Output	CAN bus data output
CANn_RXD	Input	CAN bus data input
CANn_STANDBY	Output	CAN controller is in standby mode

¹ n=0,...,3

51.4 Functional Description

51.4.1 CAN General Description

CAN pin functions

- CANn_TXD: CAN bus output data.
- CANn_RXD: CAN bus input data.
- CANn_STANDBY: When this signal is high, it indicates the CAN controller is in standby mode. The CAN clock can be turned off to save power at this time.

51.4.2 Bit Time

Bit time refers to the period of each bit on the CAN bus $T=1/\text{baud rate}$. It consists of three parts as shown in Figure 67,

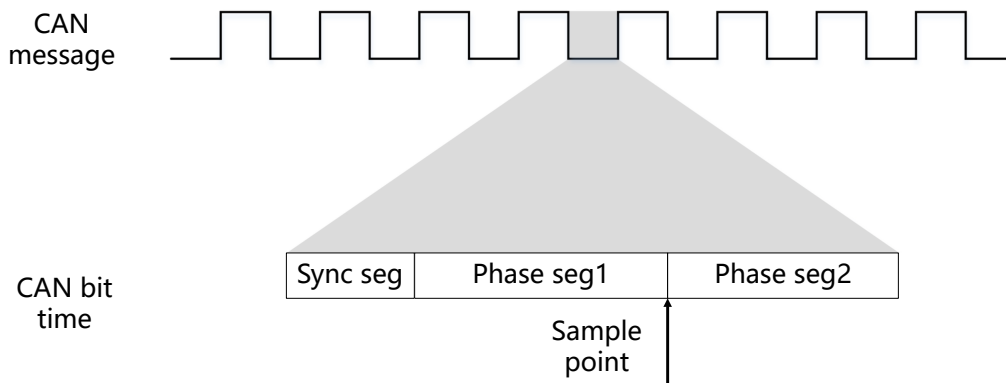


Figure 67: CAN Bit Time

Synchronization Segment: Refers to the time length required for the value to change and stabilize, which is fixed to one time quantum here.

Time Segment 1: Defines the sampling point, including the propagation segment and phase buffer segment 1 defined in the CAN standard. Its period is configurable, with a maximum of 16 time quanta and a minimum of 1 time quantum.

Time Segment 2: Defines the position of the transmission point. This segment refers to phase buffer segment 2 in the CAN standard. Its period is configurable, with a maximum of 8 time quanta and a minimum of 1 time quantum.

Resynchronization Jump Width: The resynchronization jump width (SJW) defines the extension or shortening of the bit segment. Its value is configurable, with a maximum of 4 time quanta and a minimum of 1 time quantum.

Therefore, the time for each segment is calculated as follows:

Time quantum $tq = (\text{CAN\_DBTP.DBRP}[4:0] + 1) * \text{can\_tq\_clk\_period}$

Synchronization segment time = $1 * tq$

Time Segment 1 time = $tq * (\text{CAN\_DBTP.DTSEG1}[4:0] + 1)$

Time Segment 2 time = $tq * (\text{CAN\_DBTP.DTSEG2}[3:0] + 1)$

For example, if the CAN clock frequency is set to 80MHz, i.e., 12.5ns, and you want to configure a baud rate of 1M, you can set the time quantum tq to 8 clock cycles, i.e., configure the CAN_DBTP.DBRP[4:0] register to 7. At this point, one time quantum is configured as 100ns, the synchronization segment time is $1*tq=100\text{ns}$, the Time Segment 1 time can be set according to the sampling point, e.g., set the CAN_DBTP.DTSEG1[4:0] register to 4, meaning Time Segment 1 is $5tq = 500\text{ns}$. The Time Segment 2 time can be determined based on the desired baud rate. For example, to set a 1M baud rate, Time Segment 2 needs to be configured as $1000\text{ns}-100\text{ns}-500\text{ns}=400\text{ns}$, i.e., set the CAN_DBTP.DTSEG2[3:0] register to 3. In summary, to set a 1M baud rate with a CAN clock frequency of 80MHz, you can configure the CAN_DBTP.DBRP[4:0] register to 7, the CAN_DBTP.DTSEG1[4:0] register to 4, and the CAN_DBTP.DTSEG2[3:0] register to 3.

51.4.3 Normal Operating Mode

After CAN initialization is complete, the INIT bit of the CCCR register can be set to 0. At this point, the CAN controller is ready to communicate. When a message received from the CAN bus matches the rules set by the CAN controller, the received data is stored in the receive buffer or receive FIFO0/1. Data placed in the transmit buffer or transmit FIFO0/1 will be automatically prepared for transmission to the CAN bus after the TXBAR register is configured.

51.4.4 CAN FD Operating Mode

Compared to normal mode, the FD mode has two main differences: First, the data payload length is greatly extended, with DLC supporting up to 64 bytes. Second, the control part, data part, and CRC part of the CAN FD frame operate at a higher bit rate.

CAN FD can be enabled by setting the FDOE bit of the CCCR register. When CAN FD is enabled, both FD frames and normal frames can be transmitted and received. When the FDF bit in the transmit or receive buffer is 0, the CAN controller will parse the frame as a normal frame. When the FDF bit is 1 and FDOE is enabled, the CAN controller will parse it as an FD frame. In practical use, after system startup, all nodes transmit in normal frame format. Only after confirming that all nodes in the system support CAN FD frame format, can all nodes switch to FD operation together.

In the CAN FD format, the meaning of DLC is somewhat different from the standard CAN format. The meaning is the same as standard CAN for DLC values 0 to 8. The meaning for values 9 to 15 is as shown in [Table 197](#):

Table 197: DLC Meaning

DLC Value	9	10	11	12	13	14	15
Represented Byte Count	12	16	20	24	32	48	64

51.4.5 Silent Mode

This mode can be enabled by setting the MON bit of the CCCR register to 1. In silent mode (ISO11898-1:2015), the CAN controller only receives data frames and does not transmit data frames. The advantage of this mode is that it can be used to observe the data flow on the CAN bus without affecting it. Its structure is shown in [Figure 68](#).

51.4.6 Standby Mode

This mode can be enabled by setting the CSR bit of the CCCR register to 1. When the CSR bit of CCCR is set to 1, the CAN controller waits for all pending tasks in the transmit buffer to complete. Then, the INIT bit of CCCR is automatically set to 1 to prevent users from adding new tasks. After successfully entering standby mode, the CSA bit of CCCR becomes 1. Then, registers can be configured to set the STANDBY output high, and finally, the input clock can be turned off to save power. To exit standby mode, first turn on the clock for the CAN module, then write 0 to the CSR bit of CCCR, and finally clear the INIT bit of CCCR via software to wake it up.

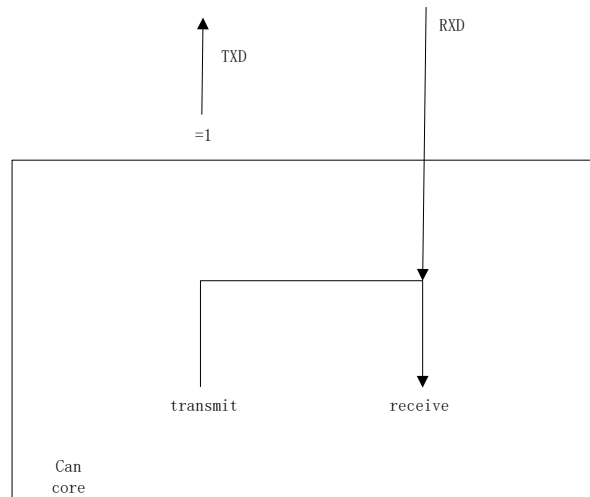


Figure 68: CAN Silent Mode

51.4.7 External Loopback Mode

This mode can be enabled by setting the LBCK bit of the TEST register. Before setting the TEST register, the TEST bit of the CCCR register must be set to 1. In this loopback mode, CAN directly sends the output data frame to the input port and places the data frame in the receive buffer or receive FIFO, and the input RXD is disconnected from the external port. As can be seen in [Figure 69](#), the relationship between the external loopback's transmit and receive ports. External loopback is used for hardware self-test.

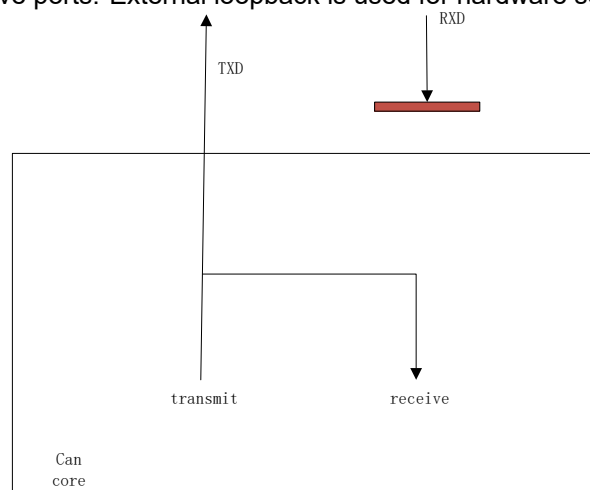


Figure 69: CAN External Loopback

51.4.8 Internal Loopback Mode

This mode can be enabled by setting the LBCK bit of the TEST register and the MON bit of the CCCR register. Before setting the TEST register, the TEST bit of the CCCR register must be set to 1. This mode can be used for hot self-test, allowing self-test of the CAN without affecting the entire running CAN system. In this case, the CAN's TXD is set to 1, and RXD is disconnected from the outside. As shown in [Figure 70](#).

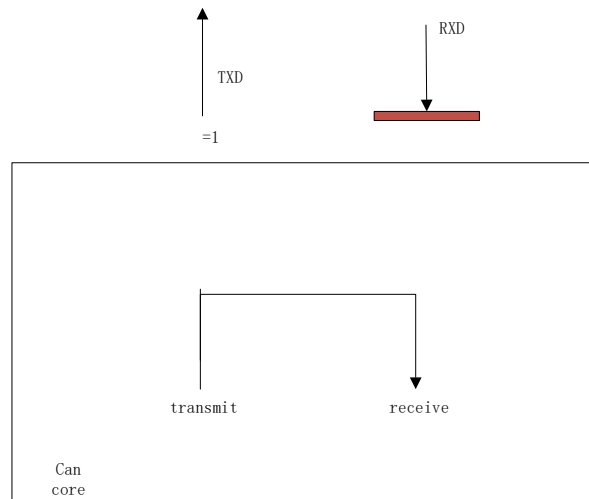


Figure 70: CAN Internal Loopback

51.4.9 Timestamp Generation

The CAN controller internally provides 4 64-bit loop counters. When CAN is in use, the GLB_CTRL register's tsu_tbin_sel can be used to select the time source. The Timestamp Unit (TSU) can store 8 64-bit timestamps or 16 32-bit timestamps. Before use, the UTSU bit of the CCCR register must be configured to 1. When using timestamps to receive messages, at the end of receiving a frame, the Timestamp Unit is triggered to record the current value of the selected counter. According to the value of the Timestamp Pointer (TSP), the counter value is stored in the corresponding timestamp register, and the Timestamp Pointer (TSP) is incremented by 1. Also, the value of the Timestamp Pointer (TSP) is placed in the RXTSP field of the receive buffer format's R1B.

51.4.10 Receive Buffer Data Format

The receive buffer data format is as shown in [Figure 71](#), where

R1A: When the Timestamp Unit (TSU) is not used (UTSU bit of CCCR is 0), the RXTS[15:0] of R1A holds the CAN controller's own internal 16-bit timestamp.

R1B: When the Timestamp Unit TSU is used (UTSU bit of CCCR is 1) and the SSYNC/ESYNC bit in the filter is set, the TSC bit of R1B is 1, indicating successful timestamp capture. R1B's RXTSP holds the TSU timestamp register number. The timestamp register corresponding to that number holds the 32-bit or 64-bit timestamp.

- R0 Bit 31 ESI: Error State Indicator.
 - 0: Transmitting node is error-active
 - 1: Transmitting node is error-passive
- R0 Bit 30 XTD: Extended Identifier.
 - 0: 11-bit standard identifier
 - 1: 29-bit extended identifier
- R0 Bit 29 RTR: Remote Transmission Request.
 - 0: Received frame is a data frame
 - 1: Received frame is a remote frame
- R0 Bits 28 to 0 ID: Identifier.
 - Valid bits for standard identifier are ID[28:18]; valid bits for extended identifier are ID[28:0]

	31			24	23	16		15	8		7	0	
R0	ESI	XTD	RTR	ID[28:0]									
R1A	ANMF	FIDX[6:0]			res	FDF	BRS	DLC[3:0]	RXTS[15:0]				
R1B	ANMF	FIDX[6:0]			res	FDF	BRS	DLC[3:0]	res			TSC	RXTSP [3:0]
R2	DB3[7:0]				DB2[7:0]				DB1[7:0]			DB0[7:0]	
R3	DB7[7:0]				DB6[7:0]				DB5[7:0]			DB4[7:0]	
...	...				...				...			...	
Rn	DBm[7:0]				DBm-1[7:0]				DBm-2[7:0]			DBm-3[7:0]	

Figure 71: CAN Receive Buffer Data Format

- R1 Bit 31 ANMF: Acceptance Non-Matched Frame.
 - 0: Received frame matched a receive filter
 - 1: Received frame did not match any receive filter
- R1 Bits 30 to 24 FIDX[6:0]: Filter Index.
 - Range 0 to 127, representing the index of the matched receive filter.
- R1 Bit 21 FDF: FD Frame Format.
 - 0: Indicates traditional CAN frame format
 - 1: Indicates FD frame format
- R1 Bit 20 BRS: Bit Rate Switch.
 - 0: Indicates the received frame has no bit rate switch
 - 1: Indicates the received frame has bit rate switch
- R1 Bits 19 to 16 DLC[3:0]: Data Length Code.
 - 0 to 8: Indicates the received frame's data length is 0 to 8 bytes
 - 9 to 15: For normal frames, indicates the received frame has 8 bytes; for FD frames, indicates 12/16/20/24/32/48/64 bytes respectively
- R1A Bits 15 to 0 RXTS[15:0]: Receive Timestamp.
 - Timestamp counter value captured at the start of frame reception.
- R1B Bit 4 TSC: Timestamp Capture.
 - 0: Indicates no timestamp was captured
 - 1: Indicates timestamp was captured, and the timestamp register number is in the RXTSP field of R1B
- R1B Bits 3 to 0 RXTSP[3:0]: Receive Timestamp Pointer.
 - Indicates which timestamp register the timestamp upon reception was stored in, i.e., the times-

tamp register number.

- R2 Bits 31 to 24 DB3[7:0]: Data Byte 3.
- R2 Bits 23 to 16 DB2[7:0]: Data Byte 2.
- R2 Bits 15 to 8 DB1[7:0]: Data Byte 1.
- R2 Bits 7 to 0 DB0[7:0]: Data Byte 0.
- R3 Bits 31 to 24 DB7[7:0]: Data Byte 7.
- R3 Bits 23 to 16 DB6[7:0]: Data Byte 6.
- R3 Bits 15 to 8 DB5[7:0]: Data Byte 5.
- R3 Bits 7 to 0 DB4[7:0]: Data Byte 4.
- ... : ...
- Rn Bits 31 to 24 DBm[7:0]: Data Byte m.
- Rn Bits 23 to 16 DB(m-1)[7:0]: Data Byte m-1.
- Rn Bits 15 to 8 DB(m-2)[7:0]: Data Byte m-2.
- Rn Bits 7 to 0 DB(m-3)[7:0]: Data Byte m-3.

51.4.11 Transmit Buffer Data Format

The transmit buffer data format is as shown in [Figure 72](#)

	31	24	23	16	15	8	7	0
T0	ESI	XTD	RTR	ID[28:0]				
T1	MM[7:0]			ELC	FOE	EDF	BR2	DLC[3:0]
	MM[15:8]			res				
T2	DB3[7:0]			DB2[7:0]			DB1[7:0]	
	DB0[7:0]							
T3	DB7[7:0]			DB6[7:0]			DB5[7:0]	
	DB4[7:0]							
...	...			...			...	
Tn	DBm[7:0]			DBm-1[7:0]			DBm-2[7:0]	
	DBm-3[7:0]							

Figure 72: CAN Transmit Buffer Data Format

- T0 Bit 31 ESI: Error State Indicator.
 - 0: Transmitting node is error-active
 - 1: Transmitting node is error-passive
- T0 Bit 30 XTD: Extended Identifier.
 - 0: 11-bit standard identifier
 - 1: 29-bit extended identifier
- T0 Bit 29 RTR: Remote Transmission Request.
 - 0: Received frame is a data frame
 - 1: Received frame is a remote frame

- T2 Bits 31 to 24 DB3[7:0]: Data Byte 3.
- T2 Bits 23 to 16 DB2[7:0]: Data Byte 2.
- T2 Bits 15 to 8 DB1[7:0]: Data Byte 1.
- T2 Bits 7 to 0 DB0[7:0]: Data Byte 0.

- T3 Bits 31 to 24 DB7[7:0]: Data Byte 7.
- T3 Bits 23 to 16 DB6[7:0]: Data Byte 6.
- T3 Bits 15 to 8 DB5[7:0]: Data Byte 5.
- T3 Bits 7 to 0 DB4[7:0]: Data Byte 4.

- :

- Tn Bits 31 to 24 DBm[7:0]: Data Byte m.
- Tn Bits 23 to 16 DB(m-1)[7:0]: Data Byte m-1.
- Tn Bits 15 to 8 DB(m-2)[7:0]: Data Byte m-2.
- Tn Bits 7 to 0 DB(m-3)[7:0]: Data Byte m-3.

51.4.12 Transmit Event Data Format

The transmit event data stores information about the transmission. The CPU can read the transmit event FIFO in the order of transmission. The status of the FIFO can be read from the TXEFS register. The meaning of

the data in the FIFO is as shown in Figure 73. Where E1A: When the Timestamp Unit (TSU) is not used (UTSU bit of CCCR is 0) and the WMM bit of the CCCR register is 0, the TXTS[15:0] of E1A holds the CAN controller's own internal 16-bit timestamp.

E1B: When the Timestamp Unit TSU is used (UTSU bit of CCCR is 1) or the 16-bit message marker is enabled (WMM of CCCR configured to 1), the TSC bit of E1B is 1, indicating successful timestamp capture. T1B's TXTSP holds the TSU timestamp register number. The timestamp register corresponding to that number holds the 32-bit or 64-bit timestamp.

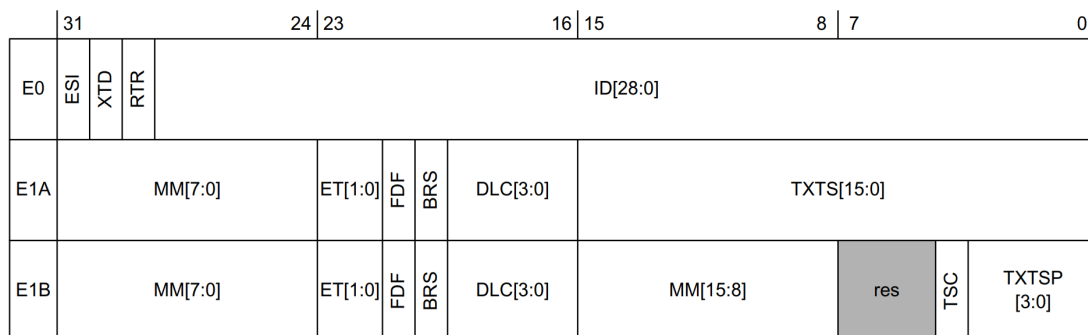


Figure 73: CAN Transmit Event Data Format

- E0 Bit 31 ESI: Error State Indicator.
 - 0: Transmitting node is error-active
 - 1: Transmitting node is error-passive
- E0 Bit 30 XTD: Extended Identifier.
 - 0: 11-bit standard identifier
 - 1: 29-bit extended identifier
- E0 Bit 29 RTR: Remote Transmission Request.
 - 0: Received frame is a data frame
 - 1: Received frame is a remote frame
- E0 Bits 28 to 0 ID: Identifier.
 - Valid bits for standard identifier are ID[28:18]; valid bits for extended identifier are ID[28:0]
- E1 Bits 31 to 24 MM[7:0]: Filter Index.
 - Range 0 to 127, representing the index of the matched receive filter.
- E1 Bits 23 to 22 ET[1:0]: Event Type.
 - 00: Reserved
 - 01: Transmit event
 - 10: Transmit after cancellation
 - 11: Reserved
- E1 Bit 21 FDF: FD Frame Format.
 - 0: Indicates traditional CAN frame format
 - 1: Indicates FD frame format
- E1 Bit 20 BRS: Bit Rate Switch.
 - 0: Indicates the received frame has no bit rate switch
 - 1: Indicates the received frame has bit rate switch
- E1 Bits 19 to 16 DLC[3:0]: Data Length Code.

- 0 to 8: Indicates the received frame's data length is 0 to 8 bytes
- 9 to 15: For normal frames, indicates the received frame has 8 bytes; for FD frames, indicates 12/16/20/24/32/48/64 bytes respectively
- E1A Bits 15 to 0 TXTS[15:0]: Transmit Timestamp.
 - Timestamp counter value captured at the start of frame transmission.
- E1B Bits 15 to 8 MM[15:8]: Message Marker higher 8 bits.
 - Written by CPU when configuring transmit buffer data, then copied to the transmit event FIFO for transmit message identification and judgment.
- E1B Bit 4 TSC: Timestamp Capture.
 - 0: Indicates no timestamp was captured
 - 1: Indicates timestamp was captured, and the timestamp register number is in the TXTSP field
- E1B Bits 3 to 0 TXTSP[3:0]: Transmit Timestamp Pointer.
 - Indicates which timestamp register the timestamp upon transmission was stored in, i.e., the timestamp register number.

51.4.13 Standard Message ID Filter

A maximum of 128 filters can be configured for 11-bit identifiers. When accessing the filter, its address is the starting address register SIDFC value plus the filter index offset. The meaning of the filter content is as shown in Figure 74

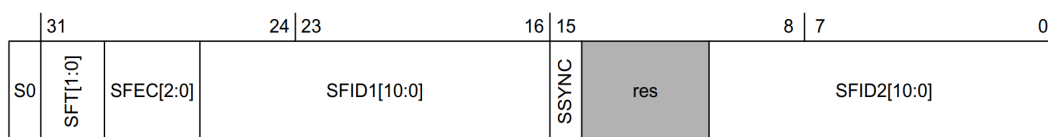


Figure 74: CAN Standard Message ID Filter

- S0 Bits 31 to 30 SFT[1:0]: Standard Filter Type.
 - 00: ID range filter, from SFID1 to SFID2, where SFID2>=SFID1.
 - 01: ID value filter, equals SFID1 or SFID2.
 - 10: Classic mask filter, SFID1 is the filter value, SFID2 is the corresponding mask. When the corresponding bit of the mask is 0, the corresponding bit of SFID1 and the ID bit can be ignored, considered as matching. When the mask bit is 1, the corresponding bit of the ID must strictly equal the corresponding bit of SFID1. For example, if all bits of SFID2 are 1, it means the ID value must exactly equal SFID1. If all bits of SFID2 are 0, it means all IDs match the SFID1 value.
 - 11: Filter disabled
- S0 Bits 29 to 27 SFEC[2:0]: Standard Filter Element Configuration.
 - 000: Filter disabled
 - 001: Store data to Receive FIFO 0 upon filter match
 - 010: Store data to Receive FIFO 1 upon filter match
 - 011: Do not store data upon filter match
 - 100: Set priority only upon filter match, do not store
 - 101: Set priority and store data to Receive FIFO 0 upon filter match
 - 110: Set priority and store data to Receive FIFO 1 upon filter match

- 111: Store data to Receive Buffer upon filter match
- S0 Bits 26 to 16 SFID1[10:0]: Standard Identifier Filter 1.
 - The first standard identifier filter, used for filtering in different situations.
- S0 Bit 15 SSYNC: Standard Synchronous Message. Only valid when the UTSU bit of CCCR is 1. Triggers the Timestamp Unit (TSU) to record a timestamp.
 - 0: Do not enable triggering TSU to record timestamp
 - 1: Enable triggering TSU to record timestamp
- S0 Bits 10 to 0 SFID2[10:0]: Standard Identifier Filter 2. Its meaning varies according to different SFEC values.

When the SFEC value is from 000 to 110, it represents the second part of the standard filter.

When the SFEC value is 111,

- SFID2[10:9] defines whether the received message is stored to a buffer or considered a debug message.
 - * 00: Store to receive buffer (Rx buffer)
 - * 01: Debug message A
 - * 10: Debug message B
 - * 11: Debug message C
- SFID2[8:6] defines the filter event pin control on the extension interface
- SFID2[5:0] defines the offset relative to the storage start address for the received data

51.4.14 Extended Message ID Filter

A maximum of 64 filters can be configured for 29-bit identifiers. When accessing the filter, its address is the starting address register XIDFC value plus twice the filter index offset. The meaning of the filter content is as shown in [Figure 75](#)

	31		24	23		16	15		8	7		0
F0	EFEC[2:0]		EFID1[28:0]									
F1	EFT[1:0]	ESYNC	EFID2[28:0]									

Figure 75: CAN Extended Message ID Filter

- F0 Bits 31 to 29 EFEC[2:0]: Extended Filter Element Configuration.
 - 000: Filter disabled
 - 001: Store data to Receive FIFO 0 upon filter match
 - 010: Store data to Receive FIFO 1 upon filter match
 - 011: Do not store data upon filter match
 - 100: Set priority only upon filter match, do not store
 - 101: Set priority and store data to Receive FIFO 0 upon filter match
 - 110: Set priority and store data to Receive FIFO 1 upon filter match
 - 111: Store data to Receive Buffer upon filter match
- F0 Bits 28 to 0 EFID1[28:0]: Extended Identifier Filter 1.
 - The first extended identifier filter, used for filtering in different situations.

- F1 Bits 31 to 30 EFT[1:0]: Extended Filter Type.
 - 00: ID range filter, from EFID1 to EFID2, where EFID2>=EFID1.
 - 01: ID value filter, equals EFID1 or EFID2.
 - 10: Classic mask filter, EFID1 is the filter value, EFID2 is the corresponding mask.
 - 11: Filter disabled
- F1 Bit 29 ESYNC: Extended Synchronous Message. Only valid when the UTSU bit of CCCR is 1. Triggers the Timestamp Unit (TSU) to record a timestamp.
 - 0: Do not enable triggering TSU to record timestamp
 - 1: Enable triggering TSU to record timestamp
- F1 Bits 28 to 0 EFID2[28:0]: Extended Identifier Filter 2. Its meaning varies according to different SFEC values.

When the EFEC value is from 000 to 110, it represents the second part of the extended filter.

When the EFEC value is 111,

- EFID2[10:9] defines whether the received message is stored to a buffer or considered a debug message.
 - * 00: Store to receive buffer (Rx buffer)
 - * 01: Debug message A
 - * 10: Debug message B
 - * 11: Debug message C
- EFID2[8:6] defines the filter event pin control on the extension interface
- EFID2[5:0] defines the offset relative to the storage start address for the received data

51.4.15 Receive FIFO

Receive FIFO 0 and Receive FIFO 1 can be configured to have 64 elements each via registers RXF0C and RSF1C respectively. Received messages, after passing through the receive filter, can be stored in the receive FIFO. To prevent receive FIFO overflow, the receive FIFO watermark threshold indicator registers RXFnC.FnWM can be used. When the number in the FIFO is greater than or equal to the threshold value, interrupt flag IR.RFnW is raised.

51.4.15.1 Receive FIFO Blocking Mode

Receive FIFO blocking mode is controlled by register RXFnC.FnOM. When this bit is configured as 0 (default is 0), it indicates the receive FIFO is in blocking mode. When the receive FIFO is full, if a message is written to the FIFO, it will not be written in. The written message may be lost. Meanwhile, flag bit RXFns.RFnL is set to 1, and interrupt flag bit IR.RFnL is set.

51.4.15.2 Receive FIFO Overwrite Mode

Receive FIFO overwrite mode is controlled by register RXFnC.FnOM. When this bit is configured as 1, it indicates the receive FIFO is in overwrite mode. When the receive FIFO is full, if a message is written to the FIFO, it will overwrite the oldest message. The oldest message will be lost. In this case, flag bit RXFns.RFnL will not be set to 1.

51.4.16 Interrupt

This module provides two sets of interrupts. The interrupt source can be connected to Interrupt 0 or Interrupt 1 via register configuration. By default, all interrupts are connected to Interrupt 0. Registers ILE.EINT0 and ILE.EINT1 can be used to control the on/off of Interrupt 0 and Interrupt 1 respectively. However, at a higher level, both Interrupt 0 and Interrupt 1 are connected to the same interrupt source. This means that if either Interrupt 0 or Interrupt 1 generates an interrupt, an interrupt will be generated. Therefore, Interrupt 0 and Interrupt 1 can be considered the same; just choose one to use. There are 30 types of interrupts. When the corresponding interrupt bit is enabled and the corresponding interrupt condition occurs, an interrupt will be issued. For detailed interrupt types, refer to the Interrupt Register (IR).

51.4.17 Message Memory Schematic

Message memory width is 32 bits. The maximum space for each part is as shown in Figure 76. The order of the parts is not strictly fixed.

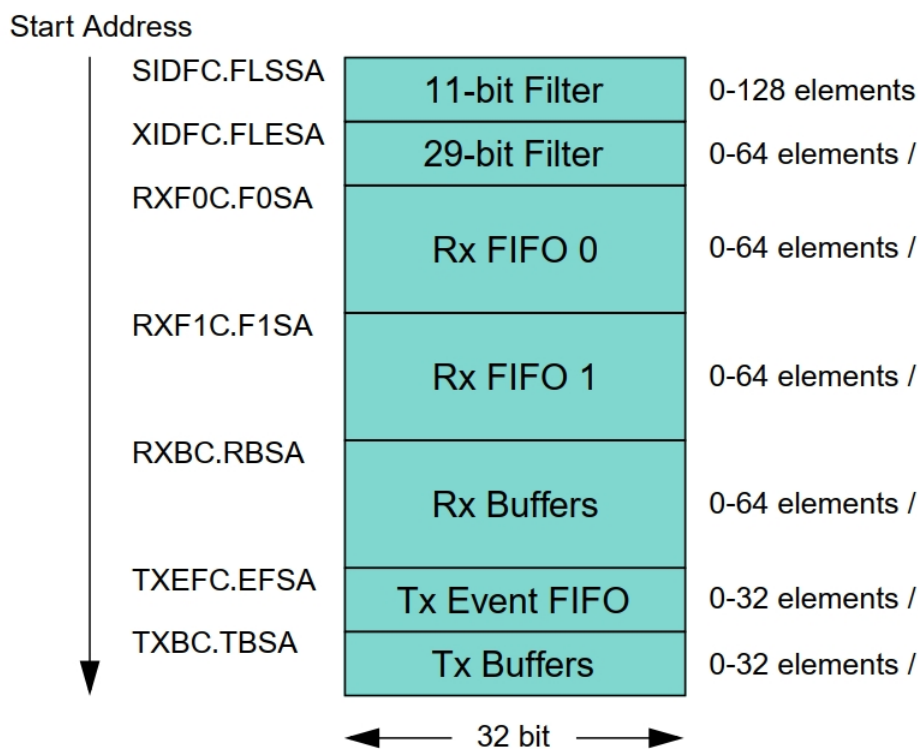


Figure 76: Message Memory Schematic

51.5 CAN Configuration and Usage Instructions

51.5.1 Software Initialization

CAN software can be initialized by setting the INIT bit of the CCCR register. When the INIT bit of the CCCR register is set to 1, the CAN bus stops receiving and transmitting data, and the output TXD remains 1. At this time, together with the CCE bit of CCCR, it controls the read/write permissions of registers. When read/write permission is available, software can perform register initialization. Note that the CCE bit can only be configured when the INIT bit is 1. When INIT is cleared to 0, CCE is also automatically cleared to 0. The following registers are reset when CCE is set:

- HPMS: High Priority Message Status
- RXF0S: Receive FIFO 0 Status
- RXF1S: Receive FIFO 1 Status
- TXFQS: Transmit FIFO/Queue Status
- TXBRP: Transmit Buffer Request Pending
- TXBTO: Transmit Buffer Transmission Occurred
- TXBCF: Transmit Buffer Cancellation Finished
- TXEFS: Transmit Event FIFO Status

And the following two registers can only be written after CCE is cleared to 0:

- TXBAR: Transmit Buffer Add Request
- TXBCR: Transmit Buffer Cancel Request

51.5.2 Receive Processing

The CAN controller provides two types of receive filters: one using standard identifiers and the other using extended identifiers. These filters can be assigned to receive buffers or receive FIFO0/1. For receive filters, each filter starts from element 0 until a matching end is encountered. Elements after a matching element are not queried for matches. The main functions of the receive filter are:

- Each filter can be configured as:
 - A range filter, where the filter matches message identifiers within one or two identifiers in registers SF1ID/SF2ID or EF1ID/EF2ID
 - Filter for one or two specific identifiers. In this case, the configured registers are SF1ID=SF2ID, or EF1ID=EF2ID
 - Classic mask form. Classic bit mask filtering uses the received identifier and a bit mask, with SF1ID/EF1ID used as the message filter and SF2ID/EF2ID used as the filter mask.
- Each filter can be configured to either receive or reject data upon match
- Each filter can be individually enabled and disabled

According to the filter configuration (SFEC/EFEC), a match will trigger one of the following operations:

- Place the received frame into FIFO0 or FIFO1
- Place the received frame into the receive buffer
- Reject the data frame
- Set the HPM bit of the high priority message interrupt flag register IR
- Set the HPM bit of the high priority message interrupt flag register IR and place the received frame into FIFO0 or FIFO1

The receive filter starts screening after receiving the complete identifier. After screening, if it is a matching identifier, the received data frame is placed into the receive buffer or FIFO in 32-bit format. If the data is not to be received or if the received data frame is detected to have an error (e.g., CRC error), then this data is discarded.

51.5.3 Transmit Processing

Transmit processing handles transmit requests from transmit buffers, the transmit FIFO, and the transmit queue. A maximum of 32 transmit buffers can be set for transmitting data, and each transmit buffer can be individually configured for either normal CAN bus mode or CAN FD mode. The specific configuration is shown

in the table below.

Table 198: Frame Transmission Configuration

Register CCCR		Transmit Buffer		Frame Transmission
BRSE	FDOE	FDF	BRS	
Ignore	0	Ignore	Ignore	Normal CAN format transmission
0	1	0	Ignore	Normal CAN format transmission
0	1	1	Ignore	CAN FD format transmission without bit rate switch
1	1	0	Ignore	Normal CAN format transmission
1	1	1	0	CAN FD format transmission without bit rate switch
1	1	1	1	CAN FD format transmission with bit rate switch

When the TXBRP register is updated, the hardware automatically scans the transmit buffer data and sends the highest priority transmit request. The TXBRP register value is automatically updated by the hardware when transmission is complete or when software adds or cancels a transmit task.

51.5.3.1 Transmit Buffer

Each transmit buffer has a configurable message identifier. If multiple transmit buffers are configured with the same message identifier, the transmit buffer with the smaller table index is sent first. After updating the data part for transmission, a transmit request can be added by configuring the TXBAR register. At this point, the transmit request is internally arbitrated for priority with the transmit FIFO or transmit queue based on the message identifier, and the one with higher priority is sent first.

51.5.3.2 Transmit FIFO

Transmit FIFO is implemented by setting the TFQM bit of the TXBC register to 0. When a message is added to the transmit FIFO, the transmit operation is enabled by writing 1 to the corresponding bit of the TXBAR register. For each message sent in the transmit FIFO, the corresponding TFGI in the TXFQS register is incremented by 1, until the transmit FIFO is empty.

51.5.3.3 Transmit Queue

Transmit queue is implemented by setting the TFQM bit of the TXBC register to 1. Messages with lower identifiers in the transmit queue have higher priority. New messages are to be placed in the transmit buffer corresponding to the TXFQS.TFQPI.

51.5.3.4 Transmit Cancellation

The CAN controller supports canceling transmit operations. This function is suitable for gateway applications and Automotive Open System Architecture. To cancel a transmit request, simply set the corresponding bit of the TXBCR register for the transmit request to 1. The cancel transmit request function is only supported for

those in transmit buffers or the transmit queue; it is not currently supported for those in the transmit FIFO. After successfully canceling a transmit request, the corresponding bit of the TXBCF register will be set to 1.

51.6 MCAN Registers

51.6.1 MCAN Register Description

The MCAN register list is as follows:

MCAN0 base address: 0xF0300000

MCAN1 base address: 0xF0304000

MCAN2 base address: 0xF0308000

MCAN3 base address: 0xF030C000

Offset	Register Name	Description	Reset Value
0x0004	ENDN	endian register	0x87654321
0x000C	DBTP	data bit timing and prescaler, writeable when CCCR.CCE and CCCR.INT are set	0x00000A33
0x0010	TEST	test register	0x00000000
0x0014	RWD	ram watchdog	0x00000000
0x0018	CCCR	CC control register	0x00000001
0x001C	NBTP	nominal bit timing and prescaler register	0x06000A03
0x0020	TSCC	timestamp counter configuration	0x00000000
0x0024	TSCV	timestamp counter value	0x00000000
0x0028	TOCC	timeout counter configuration	0xFFFF0000
0x002C	TOCV	timeout counter value	0x0000FFFF
0x0040	ECR	error counter register	0x00000000
0x0044	PSR	protocol status register	0x00000707
0x0048	TDCR	transmitter delay compensation	0x00000000
0x0050	IR	interrupt register	0x00000000
0x0054	IE	interrupt enable	0x00000000
0x0058	ILS	interrupt line select	0x00000000
0x005C	ILE	interrupt line enable	0x00000000
0x0080	GFC	global filter configuration	0x00000000
0x0084	SIDFC	standard ID filter configuration	0x00000000
0x0088	XIDFC	extended ID filter configuration	0x00000000
0x0090	XIDAM	extended id and mask	0x1FFFFFFF
0x0094	HPMS	high priority message status	0x00000000

Offset	Register Name	Description	Reset Value
0x0098	NDAT1	new data1	0x00000000
0x009C	NDAT2	new data2	0x00000000
0x00A0	RXF0C	rx fifo 0 configuration	0x00000000
0x00A4	RXF0S	rx fifo 0 status	0x00000000
0x00A8	RXF0A	rx fifo0 acknowledge	0x00000000
0x00AC	RXBC	rx buffer configuration	0x00000000
0x00B0	RXF1C	rx fifo1 configuration	0x00000000
0x00B4	RXF1S	rx fifo1 status	0x00000000
0x00B8	RXF1A	rx fifo 1 acknowledge	0x00000000
0x00BC	RXESC	rx buffer/fifo element size configuration	0x00000000
0x00C0	TXBC	tx buffer configuration	0x00000000
0x00C4	TXFQS	tx fifo/queue status	0x00000000
0x00C8	TXESC	tx buffer element size configuration	0x00000000
0x00CC	TXBRP	tx buffer request pending	0x00000000
0x00D0	TXBAR	tx buffer add request	0x00000000
0x00D4	TXBCR	tx buffer cancellation request	0x00000000
0x00D8	TXBTO	tx buffer transmission occurred	0x00000000
0x00DC	TXBCF	tx buffer cancellation finished	0x00000000
0x00E0	TXBTIE	tx buffer transmission interrupt enable	0x00000000
0x00E4	TXBCIE	tx buffer cancellation finished interrupt enable	0x00000000
0x00F0	TXEFC	tx event fifo configuration	0x00000000
0x00F4	TXEFS	tx event fifo status	0x00000000
0x00F8	TXEFA	tx event fifo acknowledge	0x00000000
0x0200	TS_SEL[TS_SEL0]	timestamp 0	0x00000000
0x0204	TS_SEL[TS_SEL1]	timestamp 1	0x00000000
0x0208	TS_SEL[TS_SEL2]	timestamp 2	0x00000000
0x020C	TS_SEL[TS_SEL3]	timestamp 3	0x00000000
0x0210	TS_SEL[TS_SEL4]	timestamp 4	0x00000000
0x0214	TS_SEL[TS_SEL5]	timestamp 5	0x00000000
0x0218	TS_SEL[TS_SEL6]	timestamp 6	0x00000000
0x021C	TS_SEL[TS_SEL7]	timestamp 7	0x00000000
0x0220	TS_SEL[TS_SEL8]	timestamp 8	0x00000000
0x0224	TS_SEL[TS_SEL9]	timestamp 9	0x00000000
0x0228	TS_SEL[TS_SEL10]	timestamp 10	0x00000000
0x022C	TS_SEL[TS_SEL11]	timestamp 11	0x00000000
0x0230	TS_SEL[TS_SEL12]	timestamp 12	0x00000000

Offset	Register Name	Description	Reset Value
0x0234	TS_SEL[TS_SEL13]	timestamp 13	0x00000000
0x0238	TS_SEL[TS_SEL14]	timestamp 14	0x00000000
0x023C	TS_SEL[TS_SEL15]	timestamp 15	0x00000000
0x0240	CREL	core release register	0x00000000
0x0244	TSCFG	timestamp configuration	0x00000000
0x0248	TSS1	timestamp status1	0x00000000
0x024C	TSS2	timestamp status2	0x00000000
0x0250	ATB	actual timebase	0x00000000
0x0254	ATBH	actual timebase high	0x00000000
0x0400	GLB_CTL	global control	0x00000000
0x0404	GLB_STATUS	global status	0x00000000

Table 199: MCAN Register List

51.6.2 Register Detailed Information

The detailed description of MCAN registers is as follows:

51.6.3 ENDN (0x4)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVT																															
R																															
1	0	0	0	0	1	1	1	0	1	1	0	0	1	0	1	0	1	0	0	0	0	1	1	0	0	1	0	0	0	0	1

ENDN [31:0]

Bits	Name	Description
31-0	EVT	Endianness Test Value The endianness test value is 0x87654321.

ENDN Bits

51.6.4 DBTP (0xC)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								TDC	RSVD		DBRP				RSVD		DTSEG1				DTSEG2				DSJW						
N/A								RW	N/A		RW				N/A		RW				RW				RW						
x	x	x	x	x	x	x	x	0	x	x	0	0	0	0	0	x	x	x	0	1	0	1	0	0	0	1	1	0	0	1	1

DBTP [31:0]

Bits	Name	Description
23	TDC	transmitter delay compensation enable 0= Transmitter Delay Compensation disabled 1= Transmitter Delay Compensation enabled
20-16	DBRP	Data Bit Rate Prescaler The value by which the oscillator frequency is divided for generating the bit time quanta. The bit time is built up from a multiple of this quanta. Valid values for the Bit Rate Prescaler are 0 to 31. When TDC = '1', the range is limited to 0,1. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.
12-8	DTSEG1	Data time segment before sample point Valid values are 0 to 31. The actual interpretation by the hardware of this value is such that one more than the programmed value is used.
7-4	DTSEG2	Data time segment after sample point Valid values are 0 to 15. The actual interpretation by the hardware of this value is such that one more than the programmed value is used.
3-0	DSJW	Data (Re)Synchronization Jump Width Valid values are 0 to 15. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.

DBTP Bits

51.6.5 TEST (0x10)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD										SVAL	TXBNS					RSVD		PVAL	TXBNP					RX	TX		LBCK	RSVD			
N/A										R	R					N/A		R	R					R	RW		RW	N/A			
x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	x	x	0	0	0	0	0	0	0	0	0	x	x	x	x	

TEST [31:0]

Bits	Name	Description
21	SVAL	Started Valid 0= Value of TXBNS not valid 1= Value of TXBNS valid

Bits	Name	Description
20-16	TXBNS	Tx Buffer Number Started Tx Buffer number of message whose transmission was started last. Valid when SVAL is set. Valid values are 0 to 31.
13	PVAL	Prepared Valid 0= Value of TXBNP not valid 1= Value of TXBNP valid
12-8	TXBNP	Tx Buffer Number Prepared Tx Buffer number of message that is ready for transmission. Valid when PVAL is set. Valid values are 0 to 31.
7	RX	Receive Pin Monitors the actual value of pin m_can_rx 0= The CAN bus is dominant (m_can_rx = '0') 1= The CAN bus is recessive (m_can_rx = '1')
6-5	TX	Control of Transmit Pin 00 Reset value, m_can_tx controlled by the CAN Core, updated at the end of the CAN bit time 01 Sample Point can be monitored at pin m_can_tx 10 Dominant ('0') level at pin m_can_tx 11 Recessive ('1') at pin m_can_tx
4	LBCK	Loop Back Mode 0= Reset value, Loop Back Mode is disabled 1= Loop Back Mode is enabled

TEST Bits

51.6.6 RWD (0x14)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																WDV								WDC							
N/A																R								RW							
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RWD [31:0]

Bits	Name	Description
15-8	WDV	Watchdog Value Actual Message RAM Watchdog Counter Value.
7-0	WDC	Watchdog Configuration Start value of the Message RAM Watchdog Counter. With the reset value of "00" the counter is disabled.

RWD Bits

51.6.7 CCCR (0x18)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																NISO	TXP	EFBI	PXHD	WMM	UTSU	BRSE	FDOE	TEST	DAR	MON	CSR	CSA	ASM	CCE	INIT
N/A																RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	R	RW	RW	RW
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

CCCR [31:0]

Bits	Name	Description
15	NISO	Non ISO Operation If this bit is set, the M_CAN uses the CAN FD frame format as specified by the Bosch CAN FD Specification V1.0. 0= CAN FD frame format according to ISO 11898-1:2015 1= CAN FD frame format according to Bosch CAN FD Specification V1.0 Note: When the generic parameter iso_only_g is set to '1' in hardware synthesis, this bit becomes reserved and is read as '0'. The M_CAN always operates with the CAN FD frame format according to ISO 11898-1:2015.
14	TXP	Transmit Pause If this bit is set, the M_CAN pauses for two CAN bit times before starting the next transmission after itself has successfully transmitted a frame (see Section 3.5). 0= Transmit pause disabled 1= Transmit pause enabled
13	EFBI	Edge Filtering during Bus Integration 0= Edge filtering disabled 1= Two consecutive dominant tq required to detect an edge for hard synchronization
12	PXHD	Protocol Exception Handling Disable 0= Protocol exception handling enabled 1= Protocol exception handling disabled Note: When protocol exception handling is disabled, the M_CAN will transmit an error frame when it detects a protocol exception condition.

Bits	Name	Description
11	WMM	Wide Message Marker Enables the use of 16-bit Wide Message Markers. When 16-bit Wide Message Markers are used (WMM = '1'), 16-bit internal timestamping is disabled for the Tx Event FIFO. 0= 8-bit Message Marker used 1= 16-bit Message Marker used, replacing 16-bit timestamps in Tx Event FIFO
10	UTSU	Use Timestamping Unit When UTSU is set, 16-bit Wide Message Markers are also enabled regardless of the value of WMM. 0= Internal time stamping 1= External time stamping by TSU Note: When generic parameter connected_tsu_g = '0', there is no TSU connected to the M_CAN. In this case bit UTSU is fixed to zero by synthesis.
9	BRSE	Bit Rate Switch Enable 0= Bit rate switching for transmissions disabled 1= Bit rate switching for transmissions enabled Note: When CAN FD operation is disabled FDOE = '0', BRSE is not evaluated.
8	FDOE	FD Operation Enable 0= FD operation disabled 1= FD operation enabled
7	TEST	Test Mode Enable 0= Normal operation, register TEST holds reset values 1= Test Mode, write access to register TEST enabled
6	DAR	Disable Automatic Retransmission 0= Automatic retransmission of messages not transmitted successfully enabled 1= Automatic retransmission disabled
5	MON	Bus Monitoring Mode Bit MON can only be set by the Host when both CCE and INIT are set to '1'. The bit can be reset by the Host at any time. 0= Bus Monitoring Mode is disabled 1= Bus Monitoring Mode is enabled
4	CSR	Clock Stop Request 0= No clock stop is requested 1= Clock stop requested. When clock stop is requested, first INIT and then CSA will be set after all pending transfer requests have been completed and the CAN bus reached idle.

Bits	Name	Description
3	CSA	Clock Stop Acknowledge 0= No clock stop acknowledged 1= M_CAN may be set in power down by stopping m_can_hclk and m_can_cclk
2	ASM	Restricted Operation Mode Bit ASM can only be set by the Host when both CCE and INIT are set to '1'. The bit can be reset by the Host at any time. For a description of the Restricted Operation Mode see Section 3.1.5. 0= Normal CAN operation 1= Restricted Operation Mode active
1	CCE	Configuration Change Enable 0= The CPU has no write access to the protected configuration registers 1= The CPU has write access to the protected configuration registers (while CCCR.INIT = '1')
0	INIT	Initialization 0= Normal Operation 1= Initialization is started

CCCR Bits

51.6.8 NBTP (0x1C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NSJW							NBRP							NTSEG1							RSVD	NTSEG2									
RW							RW							RW							N/A	RW									
0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	x	0	0	0	0	0	1	1

NBTP [31:0]

Bits	Name	Description
31-25	NSJW	Nominal (Re)Synchronization Jump Width Valid values are 0 to 127. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.

Bits	Name	Description
24-16	NBRP	Nominal Bit Rate Prescaler The value by which the oscillator frequency is divided for generating the bit time quanta. The bit time is built up from a multiple of this quanta. Valid values for the Bit Rate Prescaler are 0 to 511. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.
15-8	NTSEG1	Nominal Time segment before sample point Valid values are 1 to 255. The actual interpretation by the hardware of this value is such that one more than the programmed value is used.
6-0	NTSEG2	Nominal Time segment after sample point Valid values are 1 to 127. The actual interpretation by the hardware of this value is such that one more than the programmed value is used.

NBTP Bits

51.6.9 TSCC (0x20)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD												TCP				RSVD														TSS	
N/A												RW				N/A														RW	
x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0

TSCC [31:0]

Bits	Name	Description
19-16	TCP	Timestamp Counter Prescaler Configures the timestamp and timeout counters time unit in multiples of CAN bit times [1...16]. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.
1-0	TSS	timestamp Select 00= Timestamp counter value always 0x0000 01= Timestamp counter value incremented according to TCP 10= External timestamp counter value used 11= Same as "00"

TSCC Bits

51.6.10 TSCV (0x24)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																TSC															
N/A																RC															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TSCV [31:0]

Bits	Name	Description
15-0	TSC	Timestamp Counter The internal/external Timestamp Counter value is captured on start of frame (both Rx and Tx).When TSCC.TSS = “01” , the Timestamp Counter is incremented in multiples of CAN bit times [1…16] depending on the configuration of TSCC.TCP. A wrap around sets interrupt flag IR.TSW. Write access resets the counter to zero. When TSCC.TSS = “10” , TSC reflects the external Timestamp Counter value. A write access has no impact.

TSCV Bits

51.6.11 TOCC (0x28)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOP																RSVD														TOS	RP
RW																N/A														RW	RW
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0

TOCC [31:0]

Bits	Name	Description
31-16	TOP	Timeout Period Start value of the Timeout Counter (down-counter). Configures the Timeout Period.

Bits	Name	Description
2-1	TOS	Timeout Select When operating in Continuous mode, a write to TOCV presets the counter to the value configured by TOCC.TOP and continues down-counting. When the Timeout Counter is controlled by one of the FIFOs, an empty FIFO presets the counter to the value configured by TOCC.TOP. Down-counting is started when the first FIFO element is stored. 00= Continuous operation 01= Timeout controlled by Tx Event FIFO 10= Timeout controlled by Rx FIFO 0 11= Timeout controlled by Rx FIFO 1
0	RP	Enable Timeout Counter 0= Timeout Counter disabled 1= Timeout Counter enabled

TOCC Bits

51.6.12 TOCV (0x2C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																TOC															
N/A																RC															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

TOCV [31:0]

Bits	Name	Description
15-0	TOC	Timeout Counter The Timeout Counter is decremented in multiples of CAN bit times [1...16] depending on the configuration of TSCC.TCP. When decremented to zero, interrupt flag IR.TOO is set and the Timeout Counter is stopped. Start and reset/restart conditions are configured via TOCC.TOS. Note: Byte access: when TOCC.TOS = "00, writing one of the register bytes 3/2/1/0 will preset the Timeout Counter.

TOCV Bits

51.6.13 ECR (0x40)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								CEL							RP	REC							TEC								
N/A								X							R	R							R								
x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

ECR [31:0]

Bits	Name	Description
23-16	CEL	CAN Error Logging The counter is incremented each time when a CAN protocol error causes the 8-bit Transmit Error Counter TEC or the 7-bit Receive Error Counter REC to be incremented. The counter is also incremented when the Bus_Off limit is reached. It is not incremented when only RP is set without changing REC. The increment of CEL follows after the increment of REC or TEC. The counter is reset by read access to CEL. The counter stops at 0xFF; the next increment of TEC or REC sets interrupt flag IR.ELO. Note: Byte access: Reading byte 2 will reset CEL to zero, reading bytes 3/1/0 has no impact.
15	RP	Receive Error Passive 0= The Receive Error Counter is below the error passive level of 128 1= The Receive Error Counter has reached the error passive level of 128
14-8	REC	Receive Error Counter Actual state of the Receive Error Counter, values between 0 and 127
7-0	TEC	Transmit Error Counter Actual state of the Transmit Error Counter, values between 0 and 255 Note: When CCCR.ASM is set, the CAN protocol controller does not increment TEC and REC when a CAN protocol error is detected, but CEL is still incremented.

ECR Bits

51.6.14 PSR (0x44)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD									TDCV							RSVD	PXE	RFDF	RBRS	RESI	DLEC			BO	EW	EP	ACT		LEC		
N/A									R							N/A	X	X	X	X	S			R	R	R	R		S		
x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	x	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1

PSR [31:0]

Bits	Name	Description
22-16	TDCV	Transmitter Delay Compensation Value Position of the secondary sample point, defined by the sum of the measured delay from m_can_tx to m_can_rx and TDCR.TDCO. The SSP position is, in the data phase, the number of mtq between the start of the transmitted bit and the secondary sample point. Valid values are 0 to 127 mtq.
14	PXE	Protocol Exception Event 0= No protocol exception event occurred since last read access 1= Protocol exception event occurred Note: Byte access: Reading byte 0 will reset PXE, reading bytes 3/2/1 has no impact.
13	RFDF	Received a CAN FD Message This bit is set independent of acceptance filtering. 0= Since this bit was reset by the CPU, no CAN FD message has been received 1= Message in CAN FD format with FDF flag set has been received Note: Byte access: Reading byte 0 will reset RFDF, reading bytes 3/2/1 has no impact.
12	RBRS	BRS flag of last received CAN FD Message This bit is set together with RFDF, independent of acceptance filtering. 0= Last received CAN FD message did not have its BRS flag set 1= Last received CAN FD message had its BRS flag set Note: Byte access: Reading byte 0 will reset RBRS, reading bytes 3/2/1 has no impact.

Bits	Name	Description
11	RESI	ESI flag of last received CAN FD Message This bit is set together with RFDF, independent of acceptance filtering. 0= Last received CAN FD message did not have its ESI flag set 1= Last received CAN FD message had its ESI flag set Note: Byte access: Reading byte 0 will reset RESI, reading bytes 3/2/1 has no impact.
10-8	DLEC	Data Phase Last Error Code Type of last error that occurred in the data phase of a CAN FD format frame with its BRS flag set. Coding is the same as for LEC. This field will be cleared to zero when a CAN FD format frame with its BRS flag set has been transferred (reception or transmission) without error. Note: Byte access: Reading byte 0 will set DLEC to “111” , reading bytes 3/2/1 has no impact.
7	BO	Bus_Off Status 0= The M_CAN is not Bus_Off 1= The M_CAN is in Bus_Off state
6	EW	Warning Status 0= Both error counters are below the Error_Warning limit of 96 1= At least one of error counter has reached the Error_Warning limit of 96
5	EP	Error Passive 0= The M_CAN is in the Error_Active state. It normally takes part in bus communication and sends an active error flag when an error has been detected 1= The M_CAN is in the Error_Passive state
4-3	ACT	Activity Monitors the module' s CAN communication state. 00= Synchronizing - node is synchronizing on CAN communication 01= Idle - node is neither receiver nor transmitter 10= Receiver - node is operating as receiver 11= Transmitter - node is operating as transmitter Note: ACT is set to “00” by a Protocol Exception Event.

Bits	Name	Description
2-0	LEC	Last Error Code The LEC indicates the type of the last error to occur on the CAN bus. This field will be cleared to '0' when a message has been transferred (reception or transmission) without error. 0= No Error: No error occurred since LEC has been reset by successful reception or transmission. 1= Stuff Error: More than 5 equal bits in a sequence have occurred in a part of a received message where this is not allowed. 2= Form Error: A fixed format part of a received frame has the wrong format. 3= AckError: The message transmitted by the M_CAN was not acknowledged by another node. 4= Bit1Error: During the transmission of a message (with the exception of the arbitration field), the device wanted to send a recessive level (bit of logical value '1'), but the monitored bus value was dominant. 5= Bit0Error: During the transmission of a message (or acknowledge bit, or active error flag, or overload flag), the device wanted to send a dominant level (data or identifier bit logical value '0'), but the monitored bus value was recessive. During Bus_Off recovery this status is set each time a sequence of 11 recessive bits has been monitored. This enables the CPU to monitor the proceeding of the Bus_Off recovery sequence (indicating the bus is not stuck at dominant or continuously disturbed). 6= CRCError: The CRC check sum of a received message was incorrect. The CRC of an incoming message does not match with the CRC calculated from the received data. 7= NoChange: Any read access to the Protocol Status Register re-initializes the LEC to '7' . When the LEC shows the value '7' , no CAN bus event was detected since the last CPU read access to the Protocol Status Register. Note: When a frame in CAN FD format has reached the data phase with BRS flag set, the next CAN event (error or valid frame) will be shown in DLEC instead of LEC. An error in a fixed stuff bit of a CAN FD CRC sequence will be shown as a Form Error, not Stuff Error. Note: The Bus_Off recovery sequence (see ISO 11898-1:2015) cannot be shortened by setting or resetting CCCR.INIT. If the device goes Bus_Off, it will set CCCR.INIT of its own accord, stopping all bus activities. Once CCCR.INIT has been cleared by the CPU, the device will then wait for 129 occurrences of Bus Idle (129 * 11 consecutive recessive bits) before resuming normal operation. At the end of the Bus_Off recovery sequence, the Error

Bits	Name	Description

PSR Bits

51.6.15 TDCR (0x48)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																TDCO						RSVD	TDCF								
N/A																RW						N/A	RW								
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	x	0	0	0	0	0	0	0

TDCR [31:0]

Bits	Name	Description
14-8	TDCO	Transmitter Delay Compensation SSP Offset Offset value defining the distance between the measured delay from m_can_tx to m_can_rx and the secondary sample point. Valid values are 0 to 127 mtq.
6-0	TDCF	Transmitter Delay Compensation Filter Window Length Defines the minimum value for the SSP position, dominant edges on m_can_rx that would result in an earlier SSP position are ignored for transmitter delay measurement. The feature is enabled when TDCF is configured to a value greater than TDCO. Valid values are 0 to 127 mtq.

TDCR Bits

51.6.16 IR (0x50)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD		ARA	PED	PEA	WDI	BO	EW	EP	ELO	BEU	BEC	DRX	TOO	MRAF	TSW	TEFL	TEFF	TEFW	TEFN	TFE	TCF	TC	HPM	RF1L	RF1F	RF1W	RF1N	RF0L	RF0F	RF0W	RF0N
N/A		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IR [31:0]

Bits	Name	Description
29	ARA	Access to Reserved Address 0= No access to reserved address occurred 1= Access to reserved address occurred

Bits	Name	Description
28	PED	Protocol Error in Data Phase (Data Bit Time is used) 0= No protocol error in data phase 1= Protocol error in data phase detected (PSR.DLEC ≠ 0,7)
27	PEA	Protocol Error in Arbitration Phase (Nominal Bit Time is used) 0= No protocol error in arbitration phase 1= Protocol error in arbitration phase detected (PSR.LEC ≠ 0,7)
26	WDI	Watchdog Interrupt 0= No Message RAM Watchdog event occurred 1= Message RAM Watchdog event due to missing READY
25	BO	Bus_Off Status 0= Bus_Off status unchanged 1= Bus_Off status changed
24	EW	Warning Status 0= Error_Warning status unchanged 1= Error_Warning status changed
23	EP	Error Passive 0= Error_Passive status unchanged 1= Error_Passive status changed
22	ELO	Error Logging Overflow 0= CAN Error Logging Counter did not overflow 1= Overflow of CAN Error Logging Counter occurred
21	BEU	Bit Error Uncorrected Message RAM bit error detected, uncorrected. Controlled by input signal m_can_aeim_berr[1] generated by an optional external parity / ECC logic attached to the Message RAM. An uncorrected Message RAM bit error sets CCCR.INIT to '1'. This is done to avoid transmission of corrupted data. 0= No bit error detected when reading from Message RAM 1= Bit error detected, uncorrected (e.g. parity logic)
20	BEC	Bit Error Corrected Message RAM bit error detected and corrected. Controlled by input signal m_can_aeim_berr[0] generated by an optional external parity / ECC logic attached to the Message RAM. 0= No bit error detected when reading from Message RAM 1= Bit error detected and corrected (e.g. ECC)
19	DRX	Message stored to Dedicated Rx Buffer The flag is set whenever a received message has been stored into a dedicated Rx Buffer. 0= No Rx Buffer updated 1= At least one received message stored into an Rx Buffer

Bits	Name	Description
18	TOO	Timeout Occurred 0= No timeout 1= Timeout reached
17	MRAF	Message RAM Access Failure The flag is set, when the Rx Handler .has not completed acceptance filtering or storage of an accepted message until the arbitration field of the following message has been received. In this case acceptance filtering or message storage is aborted and the Rx Handler starts processing of the following message. .was not able to write a message to the Message RAM. In this case message storage is aborted. In both cases the FIFO put index is not updated resp. the New Data flag for a dedicated Rx Buffer is not set, a partly stored message is overwritten when the next message is stored to this location. The flag is also set when the Tx Handler was not able to read a message from the Message RAM in time. In this case message transmission is aborted. In case of a Tx Handler access failure the M_CAN is switched into Restricted Operation Mode (see Section 3.1.5). To leave Restricted Operation Mode, the Host CPU has to reset CCCR.ASM. 0= No Message RAM access failure occurred 1= Message RAM access failure occurred
16	TSW	Timestamp Wraparound 0= No timestamp counter wrap-around 1= Timestamp counter wrapped around
15	TEFL	Tx Event FIFO Element Lost 0= No Tx Event FIFO element lost 1= Tx Event FIFO element lost, also set after write attempt to Tx Event FIFO of size zero
14	TEFF	Tx Event FIFO Full 0= Tx Event FIFO not full 1= Tx Event FIFO full
13	TEFW	Tx Event FIFO Watermark Reached 0= Tx Event FIFO fill level below watermark 1= Tx Event FIFO fill level reached watermark

Bits	Name	Description
12	TEFN	Tx Event FIFO New Entry 0= Tx Event FIFO unchanged 1= Tx Handler wrote Tx Event FIFO element
11	TFE	Tx FIFO Empty 0= Tx FIFO non-empty 1= Tx FIFO empty
10	TCF	Transmission Cancellation Finished 0= No transmission cancellation finished 1= Transmission cancellation finished
9	TC	Transmission Completed 0= No transmission completed 1= Transmission completed
8	HPM	High Priority Message 0= No high priority message received 1= High priority message received
7	RF1L	Rx FIFO 1 Message Lost 0= No Rx FIFO 1 message lost 1= Rx FIFO 1 message lost, also set after write attempt to Rx FIFO 1 of size zero
6	RF1F	Rx FIFO 1 Full 0= Rx FIFO 1 not full 1= Rx FIFO 1 full
5	RF1W	Rx FIFO 1 Watermark Reached 0= Rx FIFO 1 fill level below watermark 1= Rx FIFO 1 fill level reached watermark
4	RF1N	Rx FIFO 1 New Message 0= No new message written to Rx FIFO 1 1= New message written to Rx FIFO 1
3	RF0L	Rx FIFO 0 Message Lost 0= No Rx FIFO 0 message lost 1= Rx FIFO 0 message lost, also set after write attempt to Rx FIFO 0 of size zero
2	RF0F	Rx FIFO 0 Full 0= Rx FIFO 0 not full 1= Rx FIFO 0 full
1	RF0W	Rx FIFO 0 Watermark Reached 0= Rx FIFO 0 fill level below watermark 1= Rx FIFO 0 fill level reached watermark
0	RF0N	Rx FIFO 0 New Message 0= No new message written to Rx FIFO 0 1= New message written to Rx FIFO 0

Bits	Name	Description

IR Bits

51.6.17 IE (0x54)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD		ARAE	PEDE	PEAE	WDIE	BOE	EWE	EPE	ELOE	BEUE	BECE	DRXE	TOOE	MRAFE	TSWE	TEFLE	TEFFE	TEFWE	TEFNE	TFEE	TCFE	TCE	HPME	RF1LE	RF1FE	RF1WE	RF1NE	RF0LE	RF0FE	RF0WE	RF0NE
N/A		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IE [31:0]

Bits	Name	Description
29	ARAE	Access to Reserved Address Enable
28	PEDE	Protocol Error in Data Phase Enable
27	PEAE	Protocol Error in Arbitration Phase Enable
26	WDIE	Watchdog Interrupt Enable
25	BOE	Bus_Off Status Interrupt Enable
24	EWE	Warning Status Interrupt Enable
23	EPE	Error Passive Interrupt Enable
22	ELOE	Error Logging Overflow Interrupt Enable
21	BEUE	Bit Error Uncorrected Interrupt Enable
20	BECE	Bit Error Corrected Interrupt Enable
19	DRXE	Message stored to Dedicated Rx Buffer Interrupt Enable
18	TOOE	Timeout Occurred Interrupt Enable
17	MRAFE	Message RAM Access Failure Interrupt Enable
16	TSWE	Timestamp Wraparound Interrupt Enable
15	TEFLE	Tx Event FIFO Event Lost Interrupt Enable
14	TEFFE	Tx Event FIFO Full Interrupt Enable
13	TEFWE	Tx Event FIFO Watermark Reached Interrupt Enable
12	TEFNE	Tx Event FIFO New Entry Interrupt Enable
11	TFEE	Tx FIFO Empty Interrupt Enable
10	TCFE	Transmission Cancellation Finished Interrupt Enable
9	TCE	Transmission Completed Interrupt Enable
8	HPME	High Priority Message Interrupt Enable
7	RF1LE	Rx FIFO 1 Message Lost Interrupt Enable
6	RF1FE	Rx FIFO 1 Full Interrupt Enable
5	RF1WE	Rx FIFO 1 Watermark Reached Interrupt Enable
4	RF1NE	Rx FIFO 1 New Message Interrupt Enable
3	RF0LE	Rx FIFO 0 Message Lost Interrupt Enable

Bits	Name	Description
2	RF0FE	Rx FIFO 0 Full Interrupt Enable
1	RF0WE	Rx FIFO 0 Watermark Reached Interrupt Enable
0	RF0NE	Rx FIFO 0 New Message Interrupt Enable

IE Bits

51.6.18 ILS (0x58)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RSVD			ARAL	PEDL	PEAL	WDIL	BOL	EWL	EPL	ELOL	BEUL	BECL	DRXL	TOOL	MRAFL	TSWL	TEFLL	TEFFL	TEFWL	TEFNL	TFEL	TCFL	TCL	HPML	RF1LL	RF1FL	RF1WL	RF1NL	RF0LL	RF0FL	RF0WL	RF0NL
N/A		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ILS [31:0]

Bits	Name	Description
29	ARAL	Access to Reserved Address Line
28	PEDL	Protocol Error in Data Phase Line
27	PEAL	Protocol Error in Arbitration Phase Line
26	WDIL	Watchdog Interrupt Line
25	BOL	Bus_Off Status Interrupt Line
24	EWL	Warning Status Interrupt Line
23	EPL	Error Passive Interrupt Line
22	ELOL	Error Logging Overflow Interrupt Line
21	BEUL	Bit Error Uncorrected Interrupt Line
20	BECL	Bit Error Corrected Interrupt Line
19	DRXL	Message stored to Dedicated Rx Buffer Interrupt Line
18	TOOL	Timeout Occurred Interrupt Line
17	MRAFL	Message RAM Access Failure Interrupt Line
16	TSWL	Timestamp Wraparound Interrupt Line
15	TEFLL	Tx Event FIFO Event Lost Interrupt Line
14	TEFFL	Tx Event FIFO Full Interrupt Line
13	TEFWL	Tx Event FIFO Watermark Reached Interrupt Line
12	TEFNL	Tx Event FIFO New Entry Interrupt Line
11	TFEL	Tx FIFO Empty Interrupt Line
10	TCFL	Transmission Cancellation Finished Interrupt Line
9	TCL	Transmission Completed Interrupt Line
8	HPML	High Priority Message Interrupt Line
7	RF1LL	Rx FIFO 1 Message Lost Interrupt Line
6	RF1FL	Rx FIFO 1 Full Interrupt Line

Bits	Name	Description
5	RF1WL	Rx FIFO 1 Watermark Reached Interrupt Line
4	RF1NL	Rx FIFO 1 New Message Interrupt Line
3	RF0LL	Rx FIFO 0 Message Lost Interrupt Line
2	RF0FL	Rx FIFO 0 Full Interrupt Line
1	RF0WL	Rx FIFO 0 Watermark Reached Interrupt Line
0	RF0NL	Rx FIFO 0 New Message Interrupt Line

ILS Bits

51.6.19 ILE (0x5C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																												EINT1	EINT0		
N/A																												RW	RW		
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	

ILE [31:0]

Bits	Name	Description
1	EINT1	Enable Interrupt Line 1 0= Interrupt line m_can_int1 disabled 1= Interrupt line m_can_int1 enabled
0	EINT0	Enable Interrupt Line 0 0= Interrupt line m_can_int0 disabled 1= Interrupt line m_can_int0 enabled

ILE Bits

51.6.20 GFC (0x80)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								ANFS		ANFE		RRFS		RRFE	
N/A																								RW		RW		RW		RW	
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0

GFC [31:0]

Bits	Name	Description
5-4	ANFS	Accept Non-matching Frames Standard Defines how received messages with 11-bit IDs that do not match any element of the filter list are treated. 00= Accept in Rx FIFO 0 01= Accept in Rx FIFO 1 10= Reject 11= Reject
3-2	ANFE	Accept Non-matching Frames Extended Defines how received messages with 29-bit IDs that do not match any element of the filter list are treated. 00= Accept in Rx FIFO 0 01= Accept in Rx FIFO 1 10= Reject 11= Reject
1	RRFS	Reject Remote Frames Standard 0= Filter remote frames with 11-bit standard IDs 1= Reject all remote frames with 11-bit standard IDs
0	RRFE	Reject Remote Frames Extended 0= Filter remote frames with 29-bit extended IDs 1= Reject all remote frames with 29-bit extended IDs

GFC Bits

51.6.21 SIDFC (0x84)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								LSS								FLSSA										RSVD					
N/A								RW								RW										N/A					
x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x

SIDFC [31:0]

Bits	Name	Description
23-16	LSS	List Size Standard 0= No standard Message ID filter 1-128= Number of standard Message ID filter elements >128= Values greater than 128 are interpreted as 128
15-2	FLSSA	Filter List Standard Start Address Start address of standard Message ID filter list (32-bit word address)

SIDFC Bits

51.6.22 XIDFC (0x88)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD									LSE							FLESA											RSVD				
N/A									RW							RW											N/A				
x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x

XIDFC [31:0]

Bits	Name	Description
22-16	LSE	List Size Extended 0= No extended Message ID filter 1-64= Number of extended Message ID filter elements >64= Values greater than 64 are interpreted as 64
15-2	FLESA	Filter List Extended Start Address Start address of extended Message ID filter list (32-bit word address).

XIDFC Bits

51.6.23 XIDAM (0x90)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD			EIDM																												
N/A			RW																												
x	x	x	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

XIDAM [31:0]

Bits	Name	Description
28-0	EIDM	Extended ID Mask For acceptance filtering of extended frames the Extended ID AND Mask is ANDed with the Message ID of a received frame. Intended for masking of 29-bit IDs in SAE J1939. With the reset value of all bits set to one the mask is not active.

XIDAM Bits

51.6.24 HPMS (0x94)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																FLST	FIDX						MSI		BIDX						
N/A																R	R						R		R						
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

HPMS [31:0]

Bits	Name	Description
15	FLST	Filter List Indicates the filter list of the matching filter element. 0= Standard Filter List 1= Extended Filter List
14-8	FIDX	Filter Index Index of matching filter element. Range is 0 to SIDFC.LSS - 1 resp. XIDFC.LSE - 1.
7-6	MSI	Message Storage Indicator 00= No FIFO selected 01= FIFO message lost 10= Message stored in FIFO 0 11= Message stored in FIFO 1
5-0	BIDX	Buffer Index Index of Rx FIFO element to which the message was stored. Only valid when MSI[1] = '1' .

HPMS Bits

51.6.25 NDAT1 (0x98)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ND1																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

NDAT1 [31:0]

Bits	Name	Description
31-0	ND1	New Data[31:0] The register holds the New Data flags of Rx Buffers 0 to 31. The flags are set when the respective Rx Buffer has been updated from a received frame. The flags remain set until the Host clears them. A flag is cleared by writing a ' 1 ' to the corresponding bit position. Writing a ' 0 ' has no effect. A hard reset will clear the register. 0= Rx Buffer not updated 1= Rx Buffer updated from new message

NDAT1 Bits

51.6.26 NDAT2 (0x9C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ND2																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

NDAT2 [31:0]

Bits	Name	Description
31-0	ND2	New Data[63:32] The register holds the New Data flags of Rx Buffers 32 to 63. The flags are set when the respective Rx Buffer has been updated from a received frame. The flags remain set until the Host clears them. A flag is cleared by writing a ' 1 ' to the corresponding bit position. Writing a ' 0 ' has no effect. A hard reset will clear the register. 0= Rx Buffer not updated 1= Rx Buffer updated from new message

NDAT2 Bits

51.6.27 RXF0C (0xA0)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
F00M	F0WM							RSVD	F0S							F0SA											RSVD				
RW	RW							N/A	RW							RW											N/A				
0	0	0	0	0	0	0	0	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RXF0C [31:0]

Bits	Name	Description
31	F0OM	FIFO 0 Operation Mode FIFO 0 can be operated in blocking or in overwrite mode (see Section 3.4.2). 0= FIFO 0 blocking mode 1= FIFO 0 overwrite mode
30-24	F0WM	Rx FIFO 0 Watermark 0= Watermark interrupt disabled 1-64= Level for Rx FIFO 0 watermark interrupt (IR.RF0W) >64= Watermark interrupt disabled
22-16	F0S	Rx FIFO 0 Size 0= No Rx FIFO 0 1-64= Number of Rx FIFO 0 elements >64= Values greater than 64 are interpreted as 64 The Rx FIFO 0 elements are indexed from 0 to F0S-1
15-2	F0SA	Rx FIFO 0 Start Address Start address of Rx FIFO 0 in Message RAM (32-bit word address)

RXF0C Bits

51.6.28 RXF0S (0xA4)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD						RF0L	F0F	RSVD		F0PI						RSVD		F0GI						RSVD	F0FL						
N/A						R	R	N/A		R						N/A		R						N/A	R						
x	x	x	x	x	x	0	0	x	x	0	0	0	0	0	0	x	x	0	0	0	0	0	0	x	0	0	0	0	0	0	0

RXF0S [31:0]

Bits	Name	Description
25	RF0L	Rx FIFO 0 Message Lost This bit is a copy of interrupt flag IR.RF0L. When IR.RF0L is reset, this bit is also reset. 0= No Rx FIFO 0 message lost 1= Rx FIFO 0 message lost, also set after write attempt to Rx FIFO 0 of size zero Note: Overwriting the oldest message when RXF0C.F0OM = '1' will not set this flag.

Bits	Name	Description
24	F0F	Rx FIFO 0 Full 0= Rx FIFO 0 not full 1= Rx FIFO 0 full
21-16	F0PI	Rx FIFO 0 Put Index Rx FIFO 0 write index pointer, range 0 to 63.
13-8	F0GI	Rx FIFO 0 Get Index Rx FIFO 0 read index pointer, range 0 to 63.
6-0	F0FL	Rx FIFO 0 Fill Level Number of elements stored in Rx FIFO 0, range 0 to 64.

RXF0S Bits

51.6.29 RXF0A (0xA8)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								F0AI							
N/A																								RW							
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	

RXF0A [31:0]

Bits	Name	Description
5-0	F0AI	Rx FIFO 0 Acknowledge Index After the Host has read a message or a sequence of messages from Rx FIFO 0 it has to write the buffer index of the last element read from Rx FIFO 0 to F0AI. This will set the Rx FIFO 0 Get Index RXF0S.F0GI to F0AI + 1 and update the FIFO 0 Fill Level RXF0S.F0FL.

RXF0A Bits

51.6.30 RXBC (0xAC)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
RSVD																RBSA																RSVD	
N/A																RW																N/A	
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x		

RXBC [31:0]

Bits	Name	Description
15-2	RBSA	Rx Buffer Start Address Configures the start address of the Rx Buffers section in the Message RAM (32-bit word address).Also used to reference debug messages A,B,C.

RXBC Bits

51.6.31 RXF1C (0xB0)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
F1OM	F1WM							RSVD	F1S							F1SA												RSVD			
RW	RW							N/A	RW							RW												N/A			
0	0	0	0	0	0	0	0	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x

RXF1C [31:0]

Bits	Name	Description
31	F1OM	FIFO 1 Operation Mode FIFO 1 can be operated in blocking or in overwrite mode (see Section 3.4.2). 0= FIFO 1 blocking mode 1= FIFO 1 overwrite mode
30-24	F1WM	Rx FIFO 1 Watermark 0= Watermark interrupt disabled 1-64= Level for Rx FIFO 1 watermark interrupt (IR.RF1W) >64= Watermark interrupt disabled
22-16	F1S	Rx FIFO 1 Size 0= No Rx FIFO 1 1-64= Number of Rx FIFO 1 elements >64= Values greater than 64 are interpreted as 64 The Rx FIFO 1 elements are indexed from 0 to F1S - 1
15-2	F1SA	Rx FIFO 1 Start Address Start address of Rx FIFO 1 in Message RAM (32-bit word address)

RXF1C Bits

51.6.32 RXF1S (0xB4)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMS		RSVD				RF1L	F1F	RSVD		F1PI						RSVD		F1GI						RSVD	F1FL						

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R		N/A				R	R	N/A		R				N/A		R				N/A	R										
0	0	x	x	x	x	0	0	x	x	0	0	0	0	0	0	x	x	0	0	0	0	0	0	x	0	0	0	0	0	0	0

RXF1S [31:0]

Bits	Name	Description
31-30	DMS	Debug Message Status 00= Idle state, wait for reception of debug messages, DMA request is cleared 01= Debug message A received 10= Debug messages A, B received 11= Debug messages A, B, C received, DMA request is set
25	RF1L	Rx FIFO 1 Message Lost This bit is a copy of interrupt flag IR.RF1L. When IR.RF1L is reset, this bit is also reset. 0= No Rx FIFO 1 message lost 1= Rx FIFO 1 message lost, also set after write attempt to Rx FIFO 1 of size zero Note: Overwriting the oldest message when RXF1C.F1OM = '1' will not set this flag.
24	F1F	Rx FIFO 1 Full 0= Rx FIFO 1 not full 1= Rx FIFO 1 full
21-16	F1PI	Rx FIFO 1 Put Index Rx FIFO 1 write index pointer, range 0 to 63.
13-8	F1GI	Rx FIFO 1 Get Index Rx FIFO 1 read index pointer, range 0 to 63.
6-0	F1FL	Rx FIFO 1 Fill Level Number of elements stored in Rx FIFO 1, range 0 to 64.

RXF1S Bits

51.6.33 RXF1A (0xB8)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								F1A							
N/A																								RW							
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	

RXF1A [31:0]

Bits	Name	Description
5-0	F1AI	Rx FIFO 1 Acknowledge Index After the Host has read a message or a sequence of messages from Rx FIFO 1 it has to write the buffer index of the last element read from Rx FIFO 1 to F1AI. This will set the Rx FIFO 1 Get Index RXF1S.F1GI to F1AI + 1 and update the FIFO 1 Fill Level RXF1S.F1FL.

RXF1A Bits

51.6.34 RXESC (0xBC)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					RBDS			RSVD	F1DS			RSVD	F0DS		
N/A																					RW			N/A	RW			N/A	RW		
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	x	0	0	0	x	0	0	0

RXESC [31:0]

Bits	Name	Description
10-8	RBDS	Rx Buffer Data Field Size 000= 8 byte data field 001= 12 byte data field 010= 16 byte data field 011= 20 byte data field 100= 24 byte data field 101= 32 byte data field 110= 48 byte data field 111= 64 byte data field
6-4	F1DS	Rx FIFO 1 Data Field Size 000= 8 byte data field 001= 12 byte data field 010= 16 byte data field 011= 20 byte data field 100= 24 byte data field 101= 32 byte data field 110= 48 byte data field 111= 64 byte data field

Bits	Name	Description
2-0	F0DS	Rx FIFO 0 Data Field Size 000= 8 byte data field 001= 12 byte data field 010= 16 byte data field 011= 20 byte data field 100= 24 byte data field 101= 32 byte data field 110= 48 byte data field 111= 64 byte data field Note: In case the data field size of an accepted CAN frame exceeds the data field size configured for the matching Rx Buffer or Rx FIFO, only the number of bytes as configured by RXESC are stored to the Rx Buffer resp. Rx FIFO element. The rest of the frame's data field is ignored.

RXESC Bits

51.6.35 TXBC (0xC0)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	TFQM	TFQS						RSVD	NDTB						TBSA													RSVD			
N/A	RW	RW						N/A	RW						RW													N/A			
x	0	0	0	0	0	0	0	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x

TXBC [31:0]

Bits	Name	Description
30	TFQM	Tx FIFO/Queue Mode 0= Tx FIFO operation 1= Tx Queue operation
29-24	TFQS	Transmit FIFO/Queue Size 0= No Tx FIFO/Queue 1-32= Number of Tx Buffers used for Tx FIFO/Queue >32= Values greater than 32 are interpreted as 32
21-16	NDTB	Number of Dedicated Transmit Buffers 0= No Dedicated Tx Buffers 1-32= Number of Dedicated Tx Buffers >32= Values greater than 32 are interpreted as 32

Bits	Name	Description
15-2	TBSA	Tx Buffers Start Address Start address of Tx Buffers section in Message RAM (32-bit word address, see Figure 2). Note: Be aware that the sum of TFQS and NDTB may be not greater than 32. There is no check for erroneous configurations. The Tx Buffers section in the Message RAM starts with the dedicated Tx Buffers.

TXBC Bits

51.6.36 TXFQS (0xC4)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD										TFQF	TFQPI					RSVD			TFGI			RSVD		TFFL							
N/A										R	R					N/A			R			N/A		R							
x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	x	x	x	0	0	0	0	0	x	x	0	0	0	0	0	0

TXFQS [31:0]

Bits	Name	Description
21	TFQF	Tx FIFO/Queue Full 0= Tx FIFO/Queue not full 1= Tx FIFO/Queue full
20-16	TFQPI	Tx FIFO/Queue Put Index Tx FIFO/Queue write index pointer, range 0 to 31.
12-8	TFGI	Tx FIFO Get Index Tx FIFO read index pointer, range 0 to 31. Read as zero when Tx Queue operation is configured (TXBC.TFQM = '1').
5-0	TFFL	Tx FIFO Free Level Number of consecutive free Tx FIFO elements starting from TFGI, range 0 to 32. Read as zero when Tx Queue operation is configured (TXBC.TFQM = '1') Note: In case of mixed configurations where dedicated Tx Buffers are combined with a Tx FIFO or a Tx Queue, the Put and Get Indices indicate the number of the Tx Buffer starting with the first dedicated Tx Buffers. Example: For a configuration of 12 dedicated Tx Buffers and a Tx FIFO of 20 Buffers a Put Index of 15 points to the fourth buffer of the Tx FIFO.

TXFQS Bits

51.6.37 TXESC (0xC8)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																													TBDS		
N/A																													RW		
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	

TXESC [31:0]

Bits	Name	Description
2-0	TBDS	Tx Buffer Data Field Size 000= 8 byte data field 001= 12 byte data field 010= 16 byte data field 011= 20 byte data field 100= 24 byte data field 101= 32 byte data field 110= 48 byte data field 111= 64 byte data field Note: In case the data length code DLC of a Tx Buffer element is configured to a value higher than the Tx Buffer data field size TXESC.TBDS, the bytes not defined by the Tx Buffer are transmitted as “0xCC” (padding bytes).

TXESC Bits

51.6.38 TXBRP (0xCC)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRP																															
R																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

TXBRP [31:0]

Bits	Name	Description
31-0	TRP	Transmission Request Pending Each Tx Buffer has its own Transmission Request Pending bit. The bits are set via register TXBAR. The bits are reset after a requested transmission has completed or has been cancelled via register TXBCR. TXBRP bits are set only for those Tx Buffers configured via TXBC. After a TXBRP bit has been set, a Tx scan (see Section 3.5, Tx Handling) is started to check for the pending Tx request with the highest priority (Tx Buffer with lowest Message ID). A cancellation request resets the corresponding transmission request pending bit of register TXBRP. In case a transmission has already been started when a cancellation is requested, this is done at the end of the transmission, regardless whether the transmission was successful or not. The cancellation request bits are reset directly after the corresponding TXBRP bit has been reset. After a cancellation has been requested, a finished cancellation is signalled via TXBCF ? after successful transmission together with the corresponding TXBTO bit ? when the transmission has not yet been started at the point of cancellation ? when the transmission has been aborted due to lost arbitration ? when an error occurred during frame transmission In DAR mode all transmissions are automatically cancelled if they are not successful. The corresponding TXBCF bit is set for all unsuccessful transmissions. 0= No transmission request pending 1= Transmission request pending Note: TXBRP bits which are set while a Tx scan is in progress are not considered during this particular Tx scan. In case a cancellation is requested for such a Tx Buffer, this Add Request is cancelled immediately, the corresponding TXBRP bit is reset.

TXBRP Bits

51.6.39 TXBAR (0xD0)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AR																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TXBAR [31:0]

Bits	Name	Description
31-0	AR	Add Request Each Tx Buffer has its own Add Request bit. Writing a ‘1’ will set the corresponding Add Request bit; writing a ‘0’ has no impact. This enables the Host to set transmission requests for multiple Tx Buffers with one write to TXBAR. TXBAR bits are set only for those Tx Buffers configured via TXBC. When no Tx scan is running, the bits are reset immediately, else the bits remain set until the Tx scan process has completed. 0= No transmission request added 1= Transmission requested added Note: If an add request is applied for a Tx Buffer with pending transmission request (corresponding TXBRP bit already set), this add request is ignored.

TXBAR Bits

51.6.40 TXBCR (0xD4)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CR																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TXBCR [31:0]

Bits	Name	Description
31-0	CR	Cancellation Request Each Tx Buffer has its own Cancellation Request bit. Writing a ‘1’ will set the corresponding Cancellation Request bit; writing a ‘0’ has no impact. This enables the Host to set cancellation requests for multiple Tx Buffers with one write to TXBCR. TXBCR bits are set only for those Tx Buffers configured via TXBC. The bits remain set until the corresponding bit of TXBRP is reset. 0= No cancellation pending 1= Cancellation pending

TXBCR Bits

51.6.41 TXBTO (0xD8)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0																															

TXBTO [31:0]

Bits	Name	Description
31-0	TO	Transmission Occurred Each Tx Buffer has its own Transmission Occurred bit. The bits are set when the corresponding TXBRP bit is cleared after a successful transmission. The bits are reset when a new transmission is requested by writing a ‘1’ to the corresponding bit of register TXBAR. 0= No transmission occurred 1= Transmission occurred

TXBTO Bits

51.6.42 TXBCF (0xDC)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0																															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

TXBCF [31:0]

Bits	Name	Description
31-0	CF	Cancellation Finished Each Tx Buffer has its own Cancellation Finished bit. The bits are set when the corresponding TXBRP bit is cleared after a cancellation was requested via TXBCR. In case the corresponding TXBRP bit was not set at the point of cancellation, CF is set immediately. The bits are reset when a new transmission is requested by writing a '1' to the corresponding bit of register TXBAR. 0= No transmit buffer cancellation 1= Transmit buffer cancellation finished

TXBCF Bits

51.6.43 TXBTIE (0xE0)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIE																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TXBTIE [31:0]

Bits	Name	Description
31-0	TIE	Transmission Interrupt Enable Each Tx Buffer has its own Transmission Interrupt Enable bit. 0= Transmission interrupt disabled 1= Transmission interrupt enable

TXBTIE Bits

51.6.44 TXBCIE (0xE4)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CFIE																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TXBCIE [31:0]

Bits	Name	Description
31-0	CFIE	Cancellation Finished Interrupt Enable Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0= Cancellation finished interrupt disabled 1= Cancellation finished interrupt enabled

TXBCIE Bits

51.6.45 TXEFC (0xF0)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								EFWM		RSVD						EFS															RSVD
N/A								RW		N/A						RW															N/A
x	x	0	0	0	0	0	0	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x

TXEFC [31:0]

Bits	Name	Description
29-24	EFWM	Event FIFO Watermark 0= Watermark interrupt disabled 1-32= Level for Tx Event FIFO watermark interrupt (IR.TEFW) >32= Watermark interrupt disabled
21-16	EFS	Event FIFO Size 0= Tx Event FIFO disabled 1-32= Number of Tx Event FIFO elements >32= Values greater than 32 are interpreted as 32 The Tx Event FIFO elements are indexed from 0 to EFS - 1
15-2	EFSA	Event FIFO Start Address Start address of Tx Event FIFO in Message RAM (32-bit word address)

TXEFC Bits

51.6.46 TXEFS (0xF4)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						TEFL	EFF			RSVD						EFPI						EFGI									EFFL
						R	R			N/A						R						R									R
x	x	x	x	x	x	0	0	x	x	x	0	0	0	0	0	x	x	x	0	0	0	0	0	x	x	0	0	0	0	0	0

TXEFS [31:0]

Bits	Name	Description
25	TEFL	Tx Event FIFO Element Lost This bit is a copy of interrupt flag IR.TEFL. When IR.TEFL is reset, this bit is also reset. 0= No Tx Event FIFO element lost 1= Tx Event FIFO element lost, also set after write attempt to Tx Event FIFO of size zero.
24	EFF	Event FIFO Full 0= Tx Event FIFO not full 1= Tx Event FIFO full
20-16	EFPI	Event FIFO Put Index Tx Event FIFO write index pointer, range 0 to 31.
12-8	EFGI	Event FIFO Get Index Tx Event FIFO read index pointer, range 0 to 31.
5-0	EFFL	Event FIFO Fill Level Number of elements stored in Tx Event FIFO, range 0 to 32.

TXEFS Bits

51.6.47 TXEFA (0xF8)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																									EFAI						
N/A																									RW						
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	

TXEFA [31:0]

Bits	Name	Description
4-0	EFAI	Event FIFO Acknowledge Index After the Host has read an element or a sequence of elements from the Tx Event FIFO it has to write the index of the last element read from Tx Event FIFO to EFAI. This will set the Tx Event FIFO Get Index TXEFS.EFGI to EFAI + 1 and update the Event FIFO Fill Level TXEFS.EFFL.

TXEFA Bits

51.6.48 TS_SEL (0x200 + 0x4 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TS																															
R																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TS_SEL [31:0]

Bits	Name	Description
31-0	TS	Timestamp Word TS default can save 16 timestamps with 32bit; if ts64_en is set, then work at 64bit mode, can save 8 timestamps with 01/23/45...

TS_SEL Bits

51.6.49 CREL (0x240)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REL				STEP				SUBSTEP				YEAR				MON								DAY							
R				R				R				R				R								R							
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CREL [31:0]

Bits	Name	Description
31-28	REL	Core Release One digit, BCD-coded
27-24	STEP	Step of Core Release One digit, BCD-coded.
23-20	SUBSTEP	Sub-step of Core Release One digit, BCD-coded
19-16	YEAR	Timestamp Year One digit, BCD-coded. This field is set by generic parameter on synthesis.
15-8	MON	Timestamp Month Two digits, BCD-coded. This field is set by generic parameter on synthesis.
7-0	DAY	Timestamp Day Two digits, BCD-coded. This field is set by generic parameter on synthesis.

CREL Bits

51.6.50 TSCFG (0x244)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																TBPRES								RSVD				EN64	SCP	TBCS	TSUE
N/A																RW								N/A				RW	RW	RW	RW
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	x	x	x	x	0	0	0	0

TSCFG [31:0]

Bits	Name	Description
15-8	TBPRES	Timebase Prescaler 0x00 to 0xFF The value by which the oscillator frequency is divided for generating the timebase counter clock. Valid values for the Timebase Prescaler are 0 to 255. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used. Affects only the TSU internal timebase. When the internal timebase is excluded by synthesis, TBPRES[7:0] is fixed to 0x00, the Timestamp Prescaler is not used.
3	EN64	set to use 64bit timestamp. when enabled, tsu can save up to 8 different timestamps, TS(k) and TS(k+1) are used for one 64bit timestamp, k is 0~7. TSP can be used to select different one
2	SCP	Select Capturing Position 0: Capture Timestamp at EOF 1: Capture Timestamp at SOF
1	TBCS	Timebase Counter Select When the internal timebase is excluded by synthesis, TBCS is fixed to '1' . 0: Timestamp value captured from internal timebase counter, ATB.TB[31:0] is the internal timbase counter 1: Timestamp value captured from input tsu_tbin[31:0],ATB.TB[31:0] is tsu_tbin[31:0]
0	TSUE	Timestamp Unit Enable 0: TSU disabled 1: TSU enabled

TSCFG Bits

51.6.51 TSS1 (0x248)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSL																TSN															
R																R															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TSS1 [31:0]

Bits	Name	Description
31-16	TSL	Timestamp Lost Each Timestamp register (TS0-TS15) is assigned one bit. The bits are set when the timestamp stored in the related Timestamp register was overwritten before it was read. Reading a Timestamp register resets the related bit.
15-0	TSN	Timestamp New Each Timestamp register (TS0-TS15) is assigned one bit. The bits are set when a timestamp was stored in the related Timestamp register. Reading a Timestamp register resets the related bit.

TSS1 Bits

51.6.52 TSS2 (0x24C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																TSL															
N/A																R															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

TSS2 [31:0]

Bits	Name	Description
3-0	TSP	Timestamp Pointer The Timestamp Pointer is incremented by one each time a timestamp is captured. From its maximum value (3, 7, or 15 depending on number_ts_g), it is incremented to 0. Value also signalled on output m_can_tsp[3:0].

TSS2 Bits

51.6.53 ATB (0x250)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TB																															
RC																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ATB [31:0]

Bits	Name	Description
31-0	TB	timebase for timestamp generation 31-0

ATB Bits

51.6.54 ATBH (0x254)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TBH																															
RC																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ATBH [31:0]

Bits	Name	Description
31-0	TBH	timebase for timestamp generation 63-32

ATBH Bits

51.6.55 GLB_CTL (0x400)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M_CAN_STBY	STBY_CLR_EN	STBY_POL	RSVD																										TSU_TBIN_SEL		
RW	RW	RW	N/A																										RW		
0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	

GLB_CTL [31:0]

Bits	Name	Description
31	M_CAN_STBY	m_can standby control

Bits	Name	Description
30	STBY_CLR_EN	m_can standby clear control 0:controlled by software by standby bit[bit31] 1:auto clear standby by hardware when rx data is 0
29	STBY_POL	standby polarity selection
1-0	TSU_TBIN_SEL	external timestamp select. each CAN block has 4 timestamp input, this register is used to select one of them as timestamp if TSCFG.TBCS is set to 1

GLB_CTL Bits

51.6.56 GLB_STATUS (0x404)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																												M_CAN_INT1	M_CAN_INT0	RSVD	
N/A																												R	R	N/A	
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	x	x

GLB_STATUS [31:0]

Bits	Name	Description
3	M_CAN_INT1	m_can interrupt status1
2	M_CAN_INT0	m_can interrupt status0

GLB_STATUS Bits

52 Precise Time Protocol Module PTPC

This chapter describes the main functions and features of the precise time protocol module PTPC.

52.1 Feature Summary

This chapter introduces the main features of PTPC, the precise time protocol module:

- supports 2 sets of timestamps
- Timestamp supports 63-bit counters
- counter is divided into a 32-bit second counter and a 31-bit nanosecond counter
- supports entering capture timestamps
- supports generation of interrupts when timestamps match
- supports sending data from the timestamp reading interface to the CAN module

52.2 Function Description

This chapter describes the functions of the precise time protocol module PTPC.

52.2.1 Timestamp Module

Precision Time Protocol Module PTPC contains 2 timestamp modules, each containing a 63-bit counter.

The counter is divided into a 32-bit second counter TIMEH and a 31-bit nanosecond counter TIMEL.

When each update, TIMEL does not simply add 1 like traditional counters, but adds the value of CTRL1 [SS_INCR]:

$$TIMEL = TIMEL + SS\_INCR$$

assumes that the PTPC clock is 100 MHz and each clock cycle is 10 ns. At this point, you need to configure CTRL1 [SS_INCR] to 8'd10. That is, TIMEL is accumulated in units of 10 nanoseconds at a time.

The counter is divided into a 31-bit TIMEL and a 32-bit TIMEH, and TIMEH is accumulated every time TIMEL overflows. TIMEL's overflow points support the following 2 types:

- counter overflows, the lower 31 bits TIMEL of the counter overflows when the count reaches 32'h7FFF. When TIMEL overflows, TIMEH is increased by 1
- seconds count overflows, and the counter has a lower 31-bit TIMEL. It overflows when the count reaches 100000000, which is 32'h3B9ACA00. When TIMEL overflows, TIMEH is increased by 1. TIMEL counts in nanoseconds, and TIMEH counts in seconds.

When the user sets CTRL0 [SUBSEC_DIGITAL_ROLLOVER] to 0, the counter is configured to be in counter mode. When set to 1, the configuration counter will overflow in seconds.

recommends that users set this position to 1, so that by reading the timestamp counter, you can intuitively get the real time timing of TIMEH seconds and TIMEL nanoseconds.

Before using PTPC, you need to complete PTPC initialization first. The steps to initialize the timestamp counters TIMEL and TIMEH are as follows:

- Set the CTRL0[TIMER_ENABLE] bit to 1

- writes the target value in the TS_UPDTH and TS_UPDTL registers. Note that if the TIMEL setting bit seconds count overflows, the value of TS_UPDTL should not exceed 1000000000. i.e. 32'h3B 9ACA00
- Set the CTRL0[INIT_TIMER] position to 1

After PTPC starts counting, users can update the timer as needed. The steps for PTPC to update the timestamp TIMEL and TIMEH are as follows:

- writes the target value in the TS_UPDTH and TS_UPDTL registers.
- Configure the TS_UPDTL[ADD_SUB] bit. When set to 0, TIMEL/TIMEH will add TS_UPDTL and TS_UPDTH to the original value. When set to 1, TIMEL/TIMEH will subtract TS_UPDTL and TS_UPDTH from the original value
- Set the CTRL0[UPDATE_TIMER] position to 1

PTPC supports fine-tuning the timing speed of its timestamp counter by configuring the ADDEND register.

TIMEL acts as a nanosecond counter and accumulates CTR1 [SS_INCR] in each PTPC clock cycle. When the PTPC clock is 100 MHz (i.e., the cycle is 10 ns), SS_INCR should be configured to be 10, that is, TIMEL accumulates 10 in each PTPC clock cycle. When clock synchronization software performs clock synchronization based on a PTPC timer, it is possible to find that the error between the local time and the clock Master of the network is caused by the error of the PTPC clock. At this time, equivalent fine-tuning can be achieved by configuring the ADDEND register.

Suppose the PTPC clock is 100 MHz and the TIMEL accumulation step size is 10 ns (SS_INCR = 10).

assumes that the clock synchronization algorithm finds that the clock is slow and wants to adjust the step size of TIMEL accumulation to 10.1 ns. At this time, $SS\_INCR = 10$ can be maintained and $ADDEND = 0xFFFFFFFF \times (0.1) = 0x19999999$ can be set. At this time, the equivalent step size that can be regarded as TIMEL accumulation is 10.1 ns.

assumes that the clock synchronization algorithm finds that the clock is too fast and wants to adjust the step size of TIMEL accumulation to 9.9 ns. At this time, $SS\_INCR = 9$ and $ADDEND = 0xFFFFFFFF \times (0.9) = 0xE6666666$ need to be set. At this time, the equivalent step size that can be regarded as TIMEL accumulation is 9.9 ns.

52.2.2 Timestamp Capture and Compare

PTPC supports input capture. Each timestamp supports a capture input signal. The effective edge of the CAP0 or CAP1 signal can be configured. On the rising or falling edge of CAP0 or CAP1, the values of the timestamps TIMEL and TIMEH are captured and stored in the registers CAPT_SNAPL and CAPT_SNAPH. When a capture occurs, the capture flag bit is set to 1. If the corresponding interrupt enable bit is set to 1 in the interrupt enable register, a timestamp capture interrupt request will be generated.

PTPC supports the timestamp matching function. The user writes the target timestamp value to the TARH and TARL registers and sets CTRL0 [COMP_EN]. When the timestamp counters TIMEH and TIMEL are greater than or equal to TARH and TARL, a matching event occurs. After the matching event occurs, COMP_EN will be automatically cleared.

At this time, the match flag bit will be set to 1. If the corresponding interrupt enable bit is set to 1 in the interrupt enable register, a timestamp matching interrupt request will be generated.

PTPC supports generating timestamp matching output comparisons. When timestamps match: output

CMP0 or CMP1 to other modules of the system.

52.2.3 Timestamp Output Port

PTPC supports 4 timestamp output ports, and other modules on the system can directly load TIMEL and TIMEH values from this port. The user can directly configure the source of the timestamp information on port x through the TIME_SEL register. When the corresponding position 0 is selected, the timestamp of module 0 is output. When position 1 is selected, the timestamp of module 1 is output.

52.3 PTPC register list

The register list for PTPC is as follows:

PTPC base address: 0xF037C000

Offset	Register Name	Description	Reset Value
0x0000	PTPC[0][CTRL0]	Control Register 0	0x00000000
0x0004	PTPC[0][CTRL1]	Control Register 1	0x00000000
0x0008	PTPC[0][TIMEH]	timestamp high	0x00000000
0x000C	PTPC[0][TIMEL]	timestamp low	0x00000000
0x0010	PTPC[0][TS_UPDTH]	timestamp update high	0x00000000
0x0014	PTPC[0][TS_UPDTL]	timestamp update low	0x00000000
0x0018	PTPC[0][ADDEND]		0x00000000
0x001C	PTPC[0][TARH]		0x00000000
0x0020	PTPC[0][TARL]		0x00000000
0x002C	PTPC[0][PPS_CTRL]		0x00000000
0x0030	PTPC[0][CAPT_SNAPH]		0x00000000
0x0034	PTPC[0][CAPT_SNAPL]		0x00000000
0x1000	PTPC[1][CTRL0]	Control Register 0	0x00000000
0x1004	PTPC[1][CTRL1]	Control Register 1	0x00000000
0x1008	PTPC[1][TIMEH]	timestamp high	0x00000000
0x100C	PTPC[1][TIMEL]	timestamp low	0x00000000
0x1010	PTPC[1][TS_UPDTH]	timestamp update high	0x00000000
0x1014	PTPC[1][TS_UPDTL]	timestamp update low	0x00000000
0x1018	PTPC[1][ADDEND]		0x00000000
0x101C	PTPC[1][TARH]		0x00000000
0x1020	PTPC[1][TARL]		0x00000000
0x102C	PTPC[1][PPS_CTRL]		0x00000000
0x1030	PTPC[1][CAPT_SNAPH]		0x00000000
0x1034	PTPC[1][CAPT_SNAPL]		0x00000000
0x2000	TIME_SEL		0x00000000
0x2004	INT_STS		0x00000000
0x2008	INT_EN		0x00000000
0x3000	PTPC_CAN_TS_SEL		0x00000000

Offset	Register Name	Description	Reset Value

Table 200: PTPC Register List

52.4 PTPC register description

PTPC's register is detailed as follows:

52.4.1 PTPC[CTRL0] (0x0 + 0x1000 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																						SUBSEC_DIGITAL_ROLLOVER	CAPT_SNAP_KEEP	CAPT_SNAP_POS_EN	CAPT_SNAP_NEG_EN	RSVD	COMP_EN	UPDATE_TIMER	INIT_TIMER	FINE_COARSE_SEL	TIMER_ENABLE
																						N/A									
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0

PTPC[CTRL0] [31:0]

Bits	Name	Description
9	SUBSEC_DIGITAL_ROLLOVER	Format for ns counter rollover, 1-digital, overflow time 1000000000/0x3B9ACA00 0-binary, overflow time 0x7FFFFFFF
8	CAPT_SNAP_KEEP	set will keep capture snap till software read capt_snapl. If this bit is set, software should read capt_snaph first to avoid wrong result. If this bit is cleared, capture result will be updated at each capture event
7	CAPT_SNAP_POS_EN	set will use posege of input capture signal to latch timestamp value
6	CAPT_SNAP_NEG_EN	
4	COMP_EN	set to enable compare, will be cleared by HW when compare event triggered
3	UPDATE_TIMER	update timer with +/- ts_updt, pulse, clear after set
2	INIT_TIMER	initial timer with ts_updt, pulse, clear after set
1	FINE_COARSE_SEL	0: coarse update, ns counter add ss_incr[7:0] each clk 1: fine update, ns counter add ss_incr[7:0] each time addend counter overflow

Bits	Name	Description
0	TIMER_ENABLE	

PTPC[CTRL0] Bits

52.4.2 PTPC[CTRL1] (0x4 + 0x1000 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								SS_INCR							
N/A																								RW							
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	

PTPC[CTRL1] [31:0]

Bits	Name	Description
7-0	SS_INCR	constant value used to add ns counter; such as for 50MHz timer clock, set it to 8'd20

PTPC[CTRL1] Bits

52.4.3 PTPC[TIMEH] (0x8 + 0x1000 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																TIMESTAMP_HIGH															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

PTPC[TIMEH] [31:0]

Bits	Name	Description
31-0	TIMESTAMP_HIGH	

PTPC[TIMEH] Bits

52.4.4 PTPC[TIMEH] (0xC + 0x1000 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMESTAMP_LO																RO															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PTPC[TIMEL] [31:0]

Bits	Name	Description
31-0	TIMESTAMP_LO W	

PTPC[TIMEL] Bits

52.4.5 PTPC[TS_UPDTH] (0x10 + 0x1000 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEC_UPDATE																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PTPC[TS_UPDTH] [31:0]

Bits	Name	Description
31-0	SEC_UPDATE	together with ts_updtl, used to initial or update timestamp

PTPC[TS_UPDTH] Bits

52.4.6 PTPC[TS_UPDTL] (0x14 + 0x1000 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADD_SUB																NS_UPDATE															
RW																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PTPC[TS_UPDTL] [31:0]

Bits	Name	Description
31	ADD_SUB	1 for sub; 0 for add, used only at update
30-0	NS_UPDATE	

PTPC[TS_UPDTL] Bits

52.4.7 PTPC[ADDEND] (0x18 + 0x1000 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDEND																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PTPC[ADDEND] [31:0]

Bits	Name	Description
31-0	ADDEND	used in fine update mode only

PTPC[ADDEND] Bits

52.4.8 PTPC[TARH] (0x1C + 0x1000 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TARGET_TIME_HIGH																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PTPC[TARH] [31:0]

Bits	Name	Description
31-0	TARGET_TIME_HI GH	used for generate compare signal if enabled

PTPC[TARH] Bits

52.4.9 PTPC[TARL] (0x20 + 0x1000 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TARGET_TIME_LOW																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PTPC[TARL] [31:0]

Bits	Name	Description
31-0	TARGET_TIME_LOW	

PTPC[TARL] Bits

52.4.10 PTPC[PPS_CTRL] (0x2C + 0x1000 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																PPS_CTRL															
N/A																RW															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

PTPC[PPS_CTRL] [31:0]

Bits	Name	Description
3-0	PPS_CTRL	

PTPC[PPS_CTRL] Bits

52.4.11 PTPC[CAPT_SNAPH] (0x30 + 0x1000 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPT_SNAP_HIGH																RO															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PTPC[CAPT_SNAPH] [31:0]

Bits	Name	Description
31-0	CAPT_SNAP_HIG H	take snapshot for input capture signal, at pos or neg or both; the result can be kept or updated at each event according to cfg0.bit8

PTPC[CAPT_SNAPH] Bits

52.4.12 PTPC[CAPT_SNAPL] (0x34 + 0x1000 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																CAPT_SNAP_LOW															
																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PTPC[CAPT_SNAPL] [31:0]

Bits	Name	Description
31-0	CAPT_SNAP_LO W	

PTPC[CAPT_SNAPL] Bits

52.4.13 TIME_SEL (0x2000)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																												RSVD	CAN3_TIME_SEL	CAN2_TIME_SEL	CAN1_TIME_SEL	CAN0_TIME_SEL
																													RW	RW	RW	RW
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0		

TIME_SEL [31:0]

Bits	Name	Description
3	CAN3_TIME_SEL	
2	CAN2_TIME_SEL	
1	CAN1_TIME_SEL	
0	CAN0_TIME_SEL	set to use ptpc1 for canx clr to use ptpc0 for canx

Bits	Name	Description

TIME_SEL Bits

52.4.14 INT_STS (0x2004)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD													COMP_INT_STS1	CAPTURE_INT_STS1	PPS_INT_STS1	RSVD											COMP_INT_STS0	CAPTURE_INT_STS0	PPS_INT_STS0		
N/A													W1C	W1C	W1C	N/A											W1C	W1C	W1C		
x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	

INT_STS [31:0]

Bits	Name	Description
18	COMP_INT_STS1	
17	CAPTURE_INT_STS1	
16	PPS_INT_STS1	
2	COMP_INT_STS0	
1	CAPTURE_INT_STS0	
0	PPS_INT_STS0	

INT_STS Bits

52.4.15 INT_EN (0x2008)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD													COMP_INT_STS1	CAPTURE_INT_STS1	PPS_INT_STS1	RSVD											COMP_INT_STS0	CAPTURE_INT_STS0	PPS_INT_STS0		
N/A													RW	RW	RW	N/A											RW	RW	RW		
x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0

INT_EN [31:0]

Bits	Name	Description
18	COMP_INT_STS1	
17	CAPTURE_INT_S TS1	
16	PPS_INT_STS1	
2	COMP_INT_STS0	
1	CAPTURE_INT_S TS0	
0	PPS_INT_STS0	

INT_EN Bits

52.4.16 PTPC_CAN_TS_SEL (0x3000)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSU_TBIN3_SEL						TSU_TBIN2_SEL						TSU_TBIN1_SEL						TSU_TBIN0_SEL						RSVD							
RW						RW						RW						RW						N/A							
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

PTPC_CAN_TS_SEL [31:0]

Bits	Name	Description
31-26	TSU_TBIN3_SEL	
25-20	TSU_TBIN2_SEL	
19-14	TSU_TBIN1_SEL	
13-8	TSU_TBIN0_SEL	

PTPC_CAN_TS_SEL Bits

53 1-Wire Protocol Controller OWR

This chapter introduces the functions and features of the 1-Wire controller OWR.

53.1 Feature Summary

The main features of OWR:

- Supports reading and writing a single byte.
- Supports reading and writing multiple bytes via DMA.
- Supports configurable clock division ratio.
- Supports interrupts.

53.2 External Interface Pin Description

Table 201: OWR Pin Description

Pin	Direction	Functional Description
DAT	Input/Output	Data Transmission

53.3 Functional Description

53.3.1 Clock Division

OWR supports dividing the internal system clock. The division ratio is configured in the `div_cfg` register and supports division parameters from 1 to 256. All configurable time parameters use the divided clock cycle as the unit.

53.3.2 Initialization

All transactions on the 1-Wire bus start with an initialization sequence. The initialization sequence consists of a reset pulse sent by the 1-Wire master and a presence pulse sent by the slave device. The presence pulse informs the 1-Wire master that a slave device is on the bus and ready to communicate.

During the initialization sequence, the master OWR sends a reset pulse. OWR sends the reset pulse by pulling the data bus (`onewire_data`) low for a period (the low duration can be configured via `cfg_param0[trstl]`), then releases control of `onewire_data` and enters receive mode. After the master releases `onewire_data`, the pull-up resistor pulls `onewire_data` high.

The slave device detects the rising edge of `onewire_data`, waits for a certain period (the wait time is determined by the specific slave device, typically 15 60μs), then pulls `onewire_data` low for a certain period (typically 60-240μs) to transmit the presence pulse, after which it releases the bus.

OWR samples `onewire_data` at a specific moment during the presence pulse sent by the slave to determine if a slave device is present on the bus. The sampling moment can be configured via `cfg_param0[tpst]`. Timing starts from when OWR releases the bus; `onewire_data` is sampled at time `tpst`.

The total duration of the presence pulse, from when OWR releases the bus until the next operation can be performed by `onewire`, can be configured via `cfg_param1[trsth]`.

The timing relationship between the reset pulse and the presence pulse is shown in Figure `onewire_initial`.

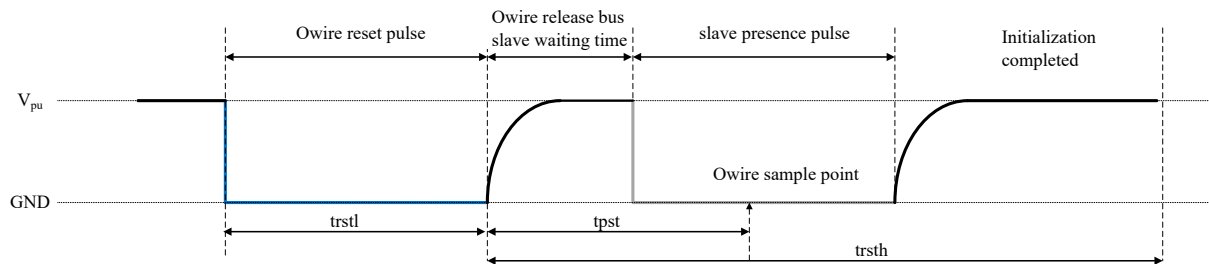


Figure 77: OWR Initialization Sequence

53.3.3 OWR Read/Write Timing

OWR transmits (writes) and receives (reads) data in slots. One slot is typically 60 120μs and represents sending or receiving one bit of data. Read and write operations are determined by the slave device protocol. Generally, a command must be written before data can be read from the slave. All commands and data sent and received are transmitted with the least significant bit (LSB) first.

53.3.3.1 Write Timing

The entire slot for each bit write cycle can be configured via `cfg_param3[twr_clr_bit]`. The low-level duration for writing a "0" is configured via `cfg_param2[tlow0]`. The low-level duration for writing a "1" is configured via `cfg_param1[tlow0]`. Additionally, a time parameter `cfg_param4[tslot_end_sra]` for SRA mode affects the time from the end of one write cycle to the start of the next write cycle. You can configure `cfg_param4[tslot_end_sra] = cfg_param3[twr_clr_bit]` to nullify the effect of this parameter. The write timing is shown in Figure write_slot.

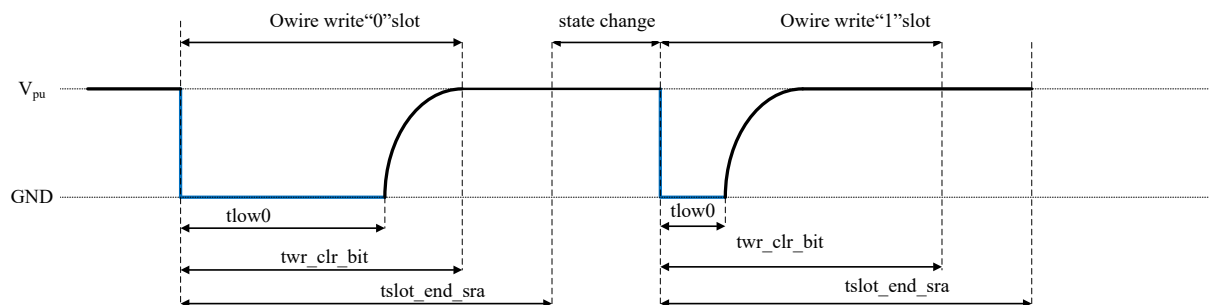


Figure 78: Write Slot Timing

53.3.3.2 Read Timing

The entire slot for each bit read cycle can be configured via `cfg_param3[twr_clr_bit]`. Each bit read operation starts with OWR pulling the data bus low. The duration OWR holds the bus low can be configured via `cfg_param2[tlowr]`. Afterwards, OWR releases the bus and waits for the slave device to return data. Starting from when OWR pulls the data bus low, OWR samples the bus after waiting for the time configured in `cfg_param2[tread_sample]` to obtain the read-back data. The read timing is shown in Figure read_slot below.

53.3.4 Reading/Writing Multiple Bytes via DMA

OWR has both TX and RX DMA ports, which can utilize the system DMA module to complete continuous data transmission or reception.

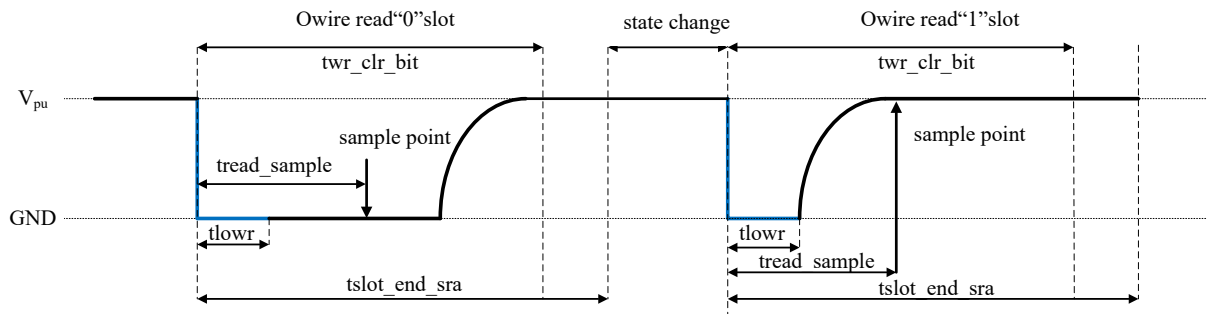


Figure 79: Read Slot Timing

For TX: Configure the system DMA module to write the data to be sent to OWR's data register.

For RX: After configuring the system DMA module, the number of bytes to receive must be written to OWR's rx_cnt register. Then, write 0xFF to OWR's data register to start reception. After each byte is received, if rx_cnt is not 1, OWR automatically sends 0xFF to start the next reception.

53.3.5 SRA (Search ROM Accelerate)

When multiple slave devices are connected on the bus, each slave device has a different ID (typically 64 bits). Before each data transmission, the ID must be sent to confirm which device to communicate with.

After the master sends the Search ROM (SR) command, all slave devices start from the least significant bit, sequentially send one bit of their ID, then send the complement of that ID bit, and then wait for the master to send a selection signal for that bit. Each slave device compares its own ID bit with the selection signal; only those where they match will continue to send their ID in the subsequent steps.

In SRA mode, OWR makes a decision for each bit in three steps: the master first reads the ID bit sent by the slave, then reads the complement of that ID bit, and then the master sends one bit to the slaves to select which slave devices can respond subsequently.

- If the IDs of the slave devices differ at this bit position, the master receives '00'.
- If the IDs of all slave devices are '0' at this bit position, the master receives '01'.
- If the IDs of all slave devices are '1' at this bit position, the master receives '10'.

OWR can be configured via the cmd register to enter SRA mode. In this mode, each time OWR receives 2 bits:

- If '00' is received, OWR sends the specified data, simultaneously writes the sent data to bit 1, and writes '1' to bit 0.
- If '01' is received, OWR sends '0', simultaneously writes '00'.
- If '10' is received, OWR sends '1', simultaneously writes '10'.

The software needs to prepare 16 bytes (128 bits) of transmission data and 16 bytes (128 bits) of reception space via DMA (Note: SRA mode does not require configuring rx_cnt; each reception is initiated by RxDMA). In the 128-bit transmission data, every two bits form a unit. Bit 1 within this two-bit unit is the data to be sent if '00' is received. For example, if the transmission data is binary 0000...10000000, it means that if bit 3 (counting from 0, every two bits as one position) receives '00', then send '1'. During the first round of search, the transmission data is all zeros. After one round of search, 128 bits of data are obtained. The even-numbered bits in the obtained 128-bit data represent the device ID, and the odd-numbered bits indicate whether there are different IDs. Based

on the obtained odd-numbered bits, the transmission data required for the next search can be prepared.

SRA mode has three configurable time parameters: `cfg_param3[tread_sam_sra]`, `cfg_param4[tslot_sra]`, and `cfg_param4[tslot_end_sra]`. Among them, `cfg_param3[tread_sam_sra]` must be greater than or equal to `cfg_param2[tread_sample]` to correctly write the ID into memory. Also, `cfg_param4[tslot_sra]` must be less than or equal to `cfg_param3[twr_clr_bit]`.

53.3.6 Interrupts

OWR supports presence, TX, and RX interrupts. These interrupts share a single interrupt line, so it is necessary to read the corresponding status bits and interrupt enable bits to determine the cause of the interrupt.

53.4 OWR Registers

53.4.1 Register Description

The list of OWR registers is as follows:

OWR0 base address: 0xF0030000

OWR1 base address: 0xF0034000

Offset	Register Name	Description	Reset Value
0x0000	CTRL	Control Register	0x00000000
0x0004	DIV_CFG		0x00000000
0x0008	RESET		0x00000000
0x000C	CMD		0x00000000
0x0010	DATA		0x00000000
0x0054	RX_CNT		0x00000001
0x0058	GLB_CFG		0x00000000
0x0060	DMA_EN		0x00000000
0x0070	IRQ_STS		0x00000000
0x0074	IRQ_EN		0x00000000

Table 202: OWR Register List

53.4.2 Register Details

The detailed description of OWR registers is as follows:

53.4.3 CTRL (0x0)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																							RPPBIT	PSTBIT	RSVD						
N/A																							RW	RO	N/A						
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	x	x	x	x	x	x	

CTRL [31:0]

Bits	Name	Description
7	RPPBIT	Reset/Presence-detect Pulse. This bit is self-clearing and is cleared after the presence or absence of an external device is determined. writing 1 generate reset pulse and sample the bus for the presence pulse from the external device. When reading: 0 Reset pulse complete. 1 Sequence not complete.
6	PSTBIT	Presence status bit, this bit is valid after the RPP bit is self-cleared. 0 Device is not present. 1 Device is present

CTRL Bits

53.4.4 DIV_CFG (0x4)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								DIV_CFG							
N/A																								RW							
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	

DIV_CFG [31:0]

Bits	Name	Description
7-0	DIV_CFG	for 1us clock divider; 2; div3, 3: div4; ...255: div256 0 and 1 are un supported configuration

DIV_CFG Bits

53.4.5 RESET (0x8)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																															SW_RESET
N/A																															RW
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	

RESET [31:0]

Bits	Name	Description
0	SW_RESET	set to reset register 0x00, 0x04,0x10,0x14, 0x18 and related logic. Clr to exit software reset

RESET Bits

53.4.6 CMD (0xC)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																														CMD	RSVD
N/A																														RW	N/A
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	x

CMD [31:0]

Bits	Name	Description
1	CMD	Search ROM Accelerator(srabit). This bit is cleared when the reset-presence-pulse bit rppbit is set. 0 Deactivate the search ROM accelerator. 1 Switch to search ROM accelerator mode.

CMD Bits

53.4.7 DATA (0x10)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																TXRX_DATA															
N/A																RW															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0

DATA [31:0]

Bits	Name	Description
7-0	TXRX_DATA	When writing: The data byte is written to the transmit buffer. When reading: A data byte is read from the receive buffer. The data is valid only when irq_sts.rbf is set.

DATA Bits

53.4.8 RX_CNT (0x54)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								RX_CNT							
N/A																								RW							
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	1	

RX_CNT [31:0]

Bits	Name	Description
7-0	RX_CNT	define RX byte number. After one byte is received and read, if value is not one, hardware will trigger new read process (write 0xFF to txrx_data, after txrx_data is read, or negedge rbf). It's used together with dma for multiple receive bytes NOTE, In SRA mode, must set rx_cnt to 1. txdma will send data to start receive process.

RX_CNT Bits

53.4.9 GLB_CFG (0x58)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RSVD																															FORCE_CLK_ON	RSVD
N/A																															RW	N/A
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	x		

GLB_CFG [31:0]

Bits	Name	Description
1	FORCE_CLK_ON	force enable 1us clock

GLB_CFG Bits

53.4.10 DMA_EN (0x60)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																TX_DMA_EN												RX_DMA_EN			

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
N/A																RW															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0

DMA_EN [31:0]

Bits	Name	Description
1	TX_DMA_EN	
0	RX_DMA_EN	

DMA_EN Bits

53.4.11 IRQ_STS (0x70)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																RSRF RBF TEMT TBE RSVD PST_DET															
N/A																W1C W1C W1C W1C N/A W1C															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	x	0

IRQ_STS [31:0]

Bits	Name	Description
5	RSRF	Receive Shift Register Full.
4	RBF	Receive Buffer Full. This flag prevents new data from being shifted into the receive buffer from the receive shift register. 0 No new data 1 A byte is waiting to be read from the TX/RX register
3	TEMT	Transmit Shift Register Empty.
2	TBE	Transmit Buffer Empty.
0	PST_DET	Presence Detect. After an 1-Wire reset has been issued, this flag is set after the appropriate amount of time(tpst) for a presence detect pulse to have occurred.

IRQ_STS Bits

53.4.12 IRQ_EN (0x74)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																ERSF ERBF ETMT ETBE RSVD EPD															
N/A																RW RW RW RW N/A RW															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	x	0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

IRQ_EN [31:0]

Bits	Name	Description
5	ERSF	
4	ERBF	
3	ETMT	
2	ETBE	
0	EPD	

IRQ_EN Bits

54 Universal Serial Bus USB

54.1 Function Introduction

provides a USB communication solution that is fully compatible with the USB 2.0 protocol

- supports OTG(can be configured in host host mode or device mode)
- Host mode supports high speed (480Mbps), full speed (12Mbps) and low speed (1.5bps)
- Device mode supports high speed (480Mbps) and full speed (12Mbps)
- Device mode supports 16 bidirectional endpoints (16IN + 16OUT endpoints)
- built-in on-chip high-speed usbphy(supports full and low speeds)
- has built-in DMA. Through the descriptor, data interaction between usb and system memory can be completed without cpu participation
- The host-mode software architecture is fully compatible with the EHCI1.0(Enhanced Host Controller Interface Specification for Universal Serial Bus) protocol
- Low power mode, you can complete wake-up through the on-chip osc clock while turning off all PLLs on the chip

54.2 pin description

Table 203: USB pin description

Pin	Direction	Function Description
USB_VBUS	Input	VBUS detect signal.
USB_ID	Input	ID detection signal.
USB_OC	Input	Overcurrent Signal.
USB_DP	bidirectional	differential signal positive terminal.
USB_DN	bidirectional	differential signal negative terminal.

54.3 Workflow

54.3.1 usbphy initialization

Exit reset state: OTG_CTRL0.OTG_UTMI_RESRET_SW= 0

Exit suspend state: OTG_CTRL0.OTG_UTMI_SUSPENDM_SW= 1

Detect usbphy clock: PHY_STATUS.UTMI_CLK_VALID=1

Wait for the usbphy clock to stabilize: while(PHY_STATUS.UTMI_CLK_VALID====1)

54.3.2 Configuration working mode

After the system is powered on, the USB module is in OTG mode. If you need to change the mode, first perform usb reset(USBCMD.RST=1).

users can configure it into host or device mode according to their needs.

can also remain in OTG mode and be configured based on external connections.

If it is a micro-AB interface, keep the ID high (pull-up via gpio configuration) and keep VBUS low

- detects that the ID has become low, indicating that a device is connected (or an otg cable may be inserted), and set to host mode (USBMODE.CM=3)
- detects that VBUS becomes high, indicating that a host is connected and is set to device mode (USB-MODE.CM=2)

If it is a typeC interface, please refer to the typeC protocol to set the host or device mode

54.3.3 Host initialization process

Turn on VBUS(usually via gpio control board power chip).

Open port: PORTSC1.PP=1

Wait for device connection, interrupt when port change is received (USBSTS.PCI), check port connection status (PORTSC1.CCS), 1 indicates that there is equipment connected to

Check linestate: PHY_STATUS.LINE_STATE

is 2, which means that a low-speed device is connected and serial mode is required (Set PORTSC1.STS=1)

is 1, which means that a full-speed or high-speed device is connected, and parallel mode is used (Set PORTSC1.STS=0)

Start USB: USBCMD.RS=1 Device: PORTSC1.PR=1

Wait for the bus reset to end: while(PORTSC1.PR==1), this time is about 55ms, during which time an interrupt of port change will be received (USBSTS.PCI)

Determine the device connection speed: PORTSC1.PSPD

Enumerate the devices (refer to the EHCI protocol for the specific process)

54.3.4 Device initialization process

Set parallel mode (device mode does not support low speed, so you can all work in parallel mode): PORTSC1.STS=0

Prepare the receive descriptor of endpoint 0 and write its address to ENDPTLISTADDR.

waits for VBUS(OTGSC.AVV) to be 1.

starts USB: USBCMD.RS= 1

waits for the host to send bus reset(USBSTS.URI)

waits for port change to interrupt USBSTS.PCI.

determines the host connection speed: PORTSC1.PSPD

waits for the host to initiate enumeration and respond.

54.4 data structure

54.4.1 Host Data Structure

To facilitate software development, the host data structure is fully compatible with the EHCI1.0 protocol. Please refer to the protocol for details.

Here is a brief introduction to the asynchronous queue used for BULK/CTRL transmission. All pictures come

from the EHCI protocol.

The asynchronous queue is a circular linked list. As shown in [Figure 80](#) , each item in the linked list (also called qhead) corresponds to an endpoint (EP, endpoint). A host can be connected to multiple devices, and a device can have multiple endpoints.

The hardware polls the linked list to see if there is any data that needs to be transferred. One of the qheads will mark H as 1, which means the link head. When the hardware polls between the link heads twice and no data is transmitted, polling will stop. During initialization, the software needs to write the address of the link list header to the ASYNCLISTADDR register. The hardware updates the ASYNCLISTADDR in real time during the polling process to the currently executing qhead address. Each qhead can link 0 to multiple qTDs. The qTD is used to transfer data, including the packet address, transmission status, and the next qTD address.

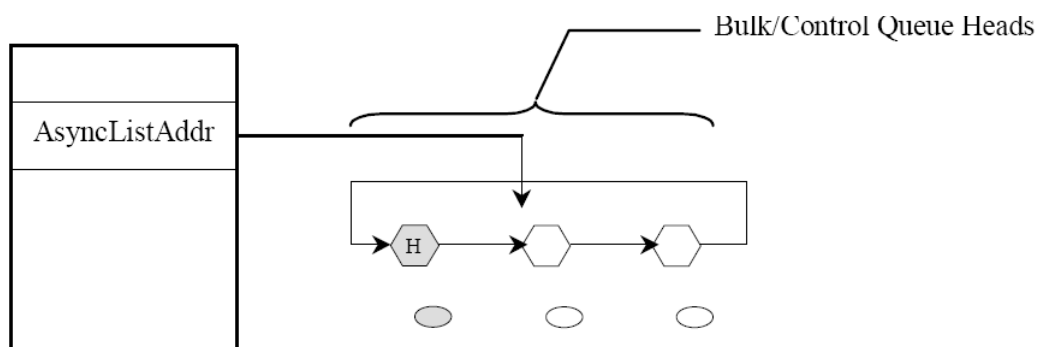


Figure 80: Asynchronous Queue Diagram

qhead is shown in [Figure 81](#) .

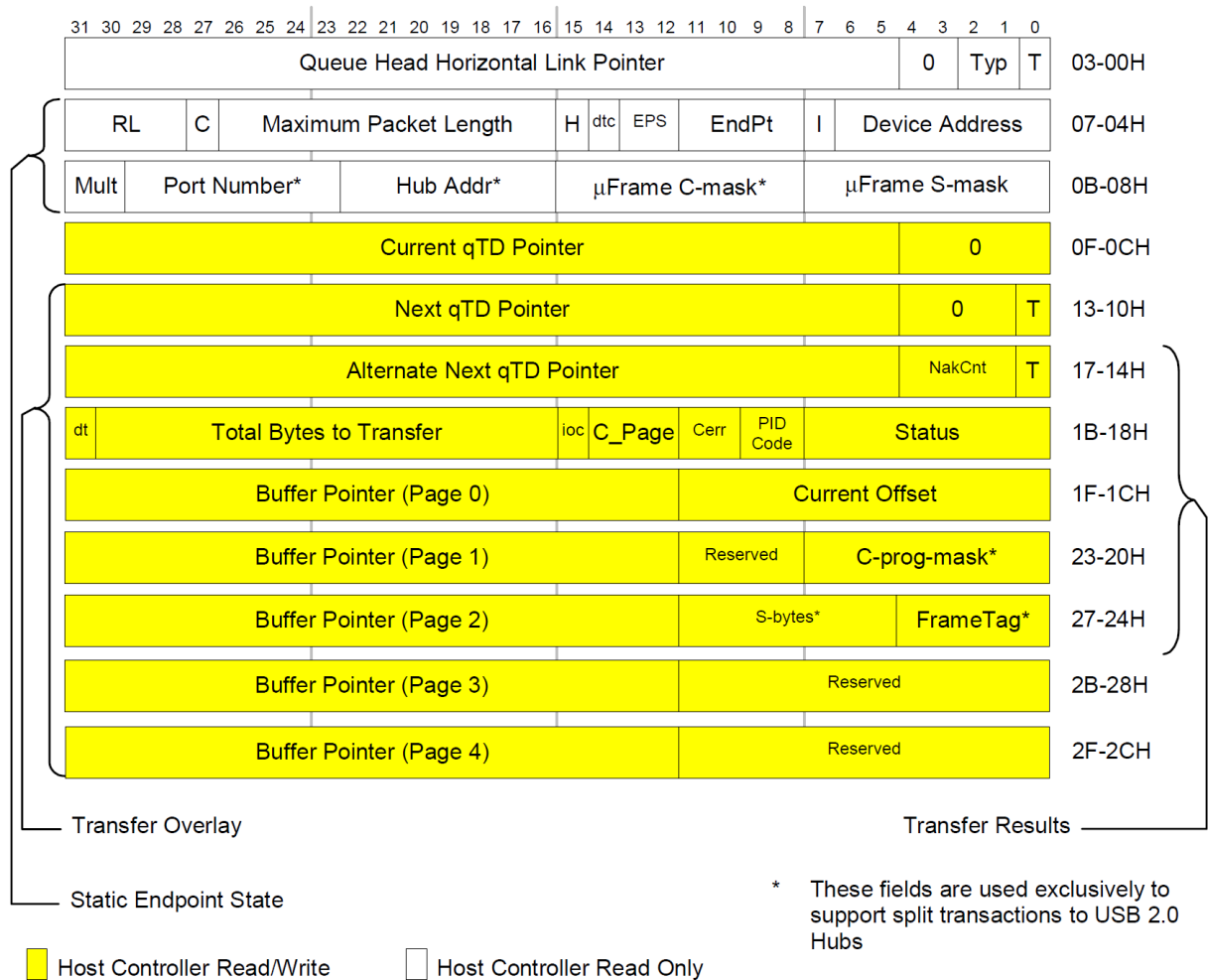


Figure 81: qhead schematic

briefly introduces the parts needed for BULK/CTRL transmission. See the EHCI protocol for those not introduced. Byte 0x10 and later parts of are also called Transfer Overlay. The hardware will write the currently transferred qTD to this part of qhead space. Before the current qTD transfer is completed, the status is updated in qhead, and written back to qTD after the transfer is completed

bytes	bits	Description
0x00	31:5	Queue Head Horizontal Link Pointer Represents the next qhead address. The qhead data structure must be 32byte aligned. If there is only one endpoint of a device in the current application, this address can point to the qhead itself.
0x00	2:1	Typ is a qhead type, and BULK and CTRL transfers need to be set to 01.
0x00	0	T is short for Terminate. Setting 1 means that the current address is invalid. For qhead of BULK/CTRL, T of Queue Head Horizontal Link Pointer needs to be 0.
0x04	31:28	RL Set 0

bytes	bits	Description
0x04	27	C denotes CTRL endpoints. When qhead.EPS is a high-speed device and the current endpoint is a CTRL endpoint, 1 needs to be set, and in other cases, 0 is set
0x04	26:16	Maximum Packet Length, The maximum packet length of the current endpoint, which can be obtained by the device during the host enumeration process, and can be set to 8 before it is obtained
0x04	15	H means that the current qhead is the first one in the linked list. If there is only one qhead in the entire qhead linked list, it is necessary to set H= 1
0x04	14	dtc, data toggle control, set 1 means that the DT bit of qtd is used as the current data toggle bit; 0 means that the data toggle is maintained by hardware; when transmitting CTRL, the data toggle is used in some special ways (see usb2 protocol 8.5.3 for details)
0x04	13:12	EPS, indicating the endpoint speed, full speed 00, low speed 01, high speed 10, 11 are reserved
0x04	11:8	EndPt, indicating the endpoint number corresponding to the current qhead
0x04	7	I is required for synchronization queues. Set 0 for BULK/CTRL transmission
0x04	6:0	DeviceAddress indicates the device address corresponding to the current qhead
0x08	31:0	is used for synchronous transmission and hub related, and is set to 0 for simple applications
0x0C	31:5	Current qTD Pointer, hardware is used to maintain the currently transmitted qTD address, set to 0 during software initialization
0x10	31:5	Next qTD Pointer, indicating the next qTD address
0x10	0	T is the abbreviation of Terminate. Setting 0 means that the current address is valid. When software initializes qhead, if there is no data to be transmitted, you need to set T to 1
0x14	31:0	can set 1 for simple applications (or T=1, other parts are optional, such as 0xbadbadff)
0x18	31	dt, data toggle. When qhead.dtc is 1, the hardware uses this bit as the starting data toggle for the current transmission. In simple applications, it is only used for CTRL transmission of

bytes	bits	Description
0x18	30:16	Total Bytes Transfer, the current data length required to be transferred by dTD can support a maximum of 20 Kbytes, and the recommended maximum is 16 Kbytes. For OUT, the software initializes the data to be sent, and the hardware decrements it after each data packet (the length is the Maximum Packet Length, except the last data packet) is transmitted. It should be 0 after the transmission is completed; For IN, the software initializes the size of the receiving space, and the hardware updates this value after each data packet is transmitted. When it is 0 or the current qTD transmission is completed after receiving a short packet(the packet length is less than the Maximum Packet Length), the result may not be 0
0x18	15	ioc, Interrupted On Complete, setting 1 to indicate that the current qTD transmission is completed and the post-bit USBSTS.UI is used to generate interrupts. For multiple qTD transmissions, the software can only set the ioc bit of the last qTD, which can reduce the number of interrupts and improve transmission efficiency.
0x18	14:12	CPage, pointer to the data buffer. The software needs to be initialized to 0. The hardware will update it in real time, indicating which buffer the current data is in (valid values are 0 to 4, corresponding to Buffer Pointer Page 0 to 4 in qTD)
0x18	11:10	Cerr, transmission error counter, software needs to initialize it to 3, each transmission error will be reduced by one, after 0, the current qTD will end and set USBSTS.UEI, used to generate error interrupt
0x18	9:8	PID Code, the current PID transmitted by qtd, valid values are: OUT-00, IN-01, SETUP-10
0x18	7:0	Status, used to indicate the current qTD status. The software needs to be initialized to 0x80. bit7: Active, which is used to indicate whether the current qTD is valid. The hardware will only consider it valid and transfer it when Active=1. When the transfer is completed, Active will be cleared; bit6: Halt. When the hardware encounters a serious error (babble, Cerr counts to 0, STALL is received), the hardware sets Halt to 1 and clears Active at the same time; bit5: Data Buffer Error, set to 1 when overrun or underrun occurs; bit4: Babble, set to 1 when hardware detects babble(USB2 protocol, 8.7.4); Bit3: XactErr, indicating that a transmission error occurred. For specific error types, please refer to EHCI protocol 4.15.1.1; Bit2:0 can be ignored in simple applications.

bytes	bits	Description
0x1C	31:12	Buffer Pointer (Page0), the starting address of the data packet is specified by buffer0 and current offset. After the hardware transfers the data from buffer0, it will switch the address to buffer1 to start transmission. The five buffers can be consecutive addresses or non-consecutive addresses. Only 4Kbyte alignment is needed to be
0x1C	11:0	Current Offset, software initializes the starting offset of the packet in buffer0. During hardware transmission, it will be updated to the offset of the current buffer(specified by cpage) in real time, indicating how much data has been transmitted
0x20 to 0x2F	31:12	For the rest of qhead, you only need to configure the data buffer pointer address for simple applications (pages 1,2,3,4)

Table 204: qhead structure

qtd is shown in Figure 82 , and the content is the same as the qhead overlay. When the software needs to transfer data, it needs to initialize one or more qtds and write the first qtd address to qhead.Next qTD Pointer.

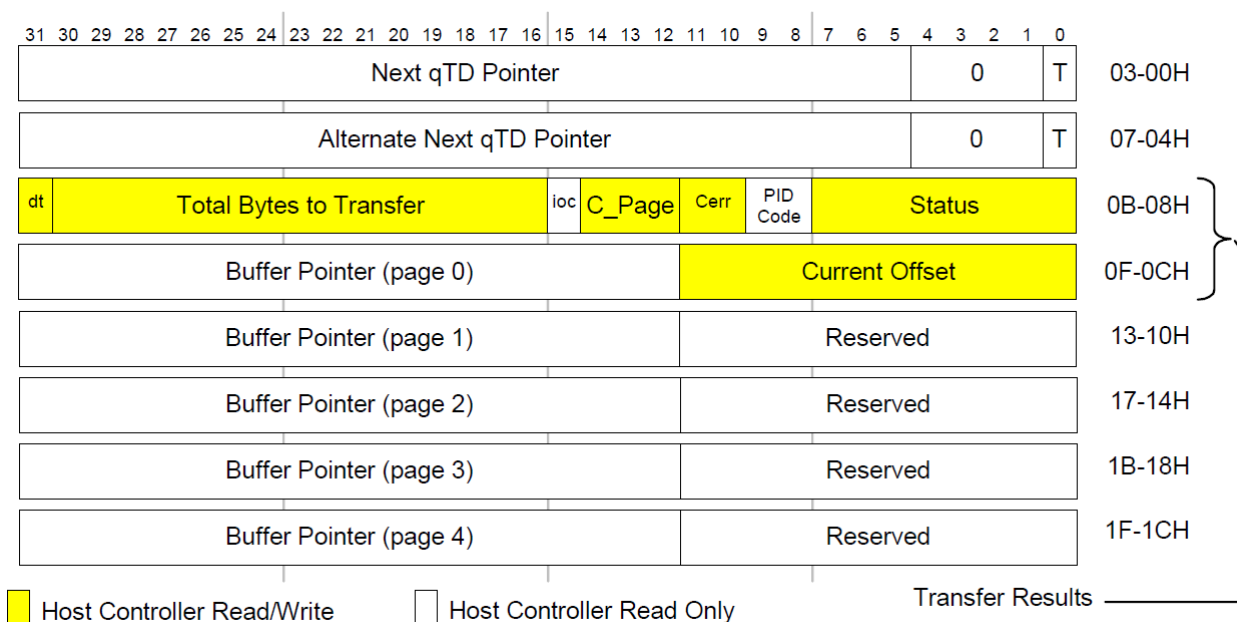


Figure 82: qtd schematic diagram

54.4.2 Device Data Structure

The device data structure refers to a piece of space allocated by software in the storage space and consists of the following parts:

- dqhlist stores all dqh. Each endpoint corresponds to two dqh(IN and OUT). Each dqh occupies 64-bytes. The starting address of the dqhlist must be aligned with 2Kbytes.
- dqh, used to configure this endpoint parameter, including the maximum packet length, current and next

transport descriptor addresses, and will also save the current transport descriptor content, as shown in [Figure 84](#) . Details are shown in [Table 205](#) .

- dtd, transfer descriptor, used to transfer data, protect data length, start address, status and other information, as shown in [Figure 85](#)

As shown in [Figure 83](#) , the ENDPTLISTADDR of the register points to the starting address of the dqhlist. The software needs to prepare at least the receiving dqh of endpoint 0 (OUT dqh of EP0, located at the starting address of dqhlist) before starting USB(USBCMD.RS=1) to receive the first SETUP packet in the enumeration information sent by the host, and then prepare the dqh and dtd of the corresponding endpoint based on the received data.

Here, taking receiving the first SETUP packet enumerated as an example to briefly introduce the software flow:

- Prepare a 2Kbyte aligned space for dqhlist, write the starting address into ENDPTLISTADDR, and the size is the maximum endpoint number required by the device multiplied by 128 bytes. For example, a mass storage device needs endpoint 1 as IN in addition to endpoint 0, and endpoints 2 and 3 as OUT, then dqhlist needs $4 \times 128 = 512$ bytes; in extreme cases, a device only needs endpoint 7 as IN, then dqhlist needs $8 \times 128 = 1\text{Kbyte}$;
- Prepare the receiving dqh for endpoint 0 (the address is located at dqhlist+0x00). The Maximum Packet Length sets the maximum packet length supported by the cost device. Generally, the low speed is 8, and the full speed or high speed is 64; ios sets 1 to generate an interrupt after receiving SETUP; T=1; Status=0; Other parts are optional;
- Prepare to send dqh for endpoint 0 (the address is located at dqhlist+0x40), and the Maximum Packet Length sets the maximum packet length supported by the cost device, Status=0, and other parts are optional;
- Set USBINTR.UI=1 and wait for interrupt;
- After receiving the UI interrupt, the 8byte content of the SETUP packet will be stored in the Setup Buffer where endpoint 0 receives dqh, and a response will be prepared based on the SETUP content;
- Assuming that you receive GET DESC, you need to prepare device information to be returned to the host, and the software needs to prepare an IN dtd to return data and a 0byte OUT dtd to end the Status phase.
- For IN dtd, T=1; TotalBytes is the number of bytes of device information; Status= 0x80; write buffer0 and Current Offset to the data starting address. If necessary (the content of device information spans 4K addresses), buffer1 can be prepared, and other parts are 0;
- For OUT dtd, T=1, TotalBytes=0, Status= 0x80, ioc=1, others are 0;
- writes the IN dtd address to the Next dTD Pointer of endpoint 0IN dqh and clears T, sets bit16 of ENDPT-PRIME, and waits for bit16 to be automatically cleared by the hardware (at this time, the hardware will read the corresponding content from IN0dqh, read the data to be sent into the internal storage space, and clear the corresponding bit of ENDPTPRIME after the end); the above content is called PRIME EP0IN.
- Do the same, PRIME EP0 OUT
- waits for an interrupt; if it completes normally, it should be an OUT interrupt from EP0, because the ioc for OUT dtd is set in the above process, but the ioc for IN is not set.

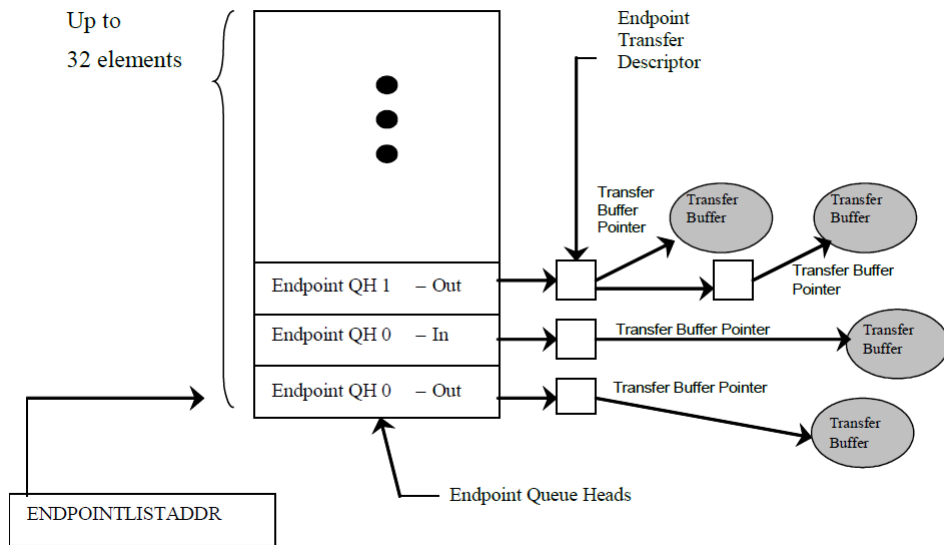


Figure 83: dqhlist diagram

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	word	
mult		zlt		0		max_packet_size										ios		0														03-00h	
cur_dtd_addr																												0		07-04h			
nxt_dtd_addr																												0		T		0B-08h	
0		total_bytes														ioc		c_page		multo		0		status				0F-0Ch					
buffer(page 0)														current_offset														13-10h					
buffer(page 1)														reserved														17-14h					
buffer(page 2)														reserved														1B-18h					
buffer(page 3)														reserved														1F-1Ch					
buffer(page 4)														reserved														23-20h					
reserved																												27-24h					
setup buffer bytes 3..0																												2B-28h					
setup buffer bytes 7..4																												2F-2Ch					

Figure 84: dqh schematic diagram

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Next dTD Pointer																											0		T	03-00h	
0	Total Bytes														ioc	C_Page	MultO	0	Status										07-04h		
Buffer Pointer (page 0)																	Current Offset											0B-08h			
Buffer Pointer (page 1)																		Frame Number										0F-0Ch			
Buffer Pointer (page 2)																	Reserved											13-10h			
Buffer Pointer (page 3)																	Reserved											17-14h			
Buffer Pointer (page 4)																	Reserved											1B-18h			
Host Controller Read/Write																	Host Controller Read Only														

Figure 85: dtd schematic diagram

is similar to the overlay part of the host qhea. The device dtd will also be copied by hardware into dqh for

transmission, so some parts of dqh (0x08 to 0x23) are exactly the same as the dtd. The

device data structure dqh is explained in detail as follows:

bytes	bits	Description
0x00	31:30	mult, ISO endpoint, must be set to 1/2/3, indicating how many packets can be transmitted per dTD Non-ISO endpoints must be set to 0.
0x00	29	zlt, setting 1 means that when the transmission length is an integer multiple of the maximum packet length, a packet of length 0 is finally added as the transmission end flag. Only for non-ISO endpoints
0x00	26: 16	max_packet_size, the maximum packet length of the endpoint.
0x00	15	ios, interrupt on setup, setting 1 means that USBSTS.UI will be set when the CTRL endpoint receives the SETUP packet. The dqh of the CTRL endpoint needs to set this bit
0x04	31:5	cur_dtd_addr, the dtd physical address being transferred in the current dqh, 32-byte aligned. When endpoint prime, or when a dtd transfer is completed, the hardware copies nxt_dtd_addr to the current location
0x08	31:5	nxt_dtd_addr, next dtd physical address to be processed, 32-byte aligned
0x08	0	T(Terminate), indicating whether the nxt_dtd_addr address is valid, 0 means valid, 1 means the current dtd is the last, and there is no next valid dtd in the queue
0x0C	30:16	total_bytes, which indicates the length of the current dtd needs to be transferred. For IN, indicates the length of the packet to be passed to the host; For OUT, it represents the length that can be received from the host. The actual received length may be less than this value.
0x0C	15	ioc, Interrupted On Complete, setting 1 means that after the current dtd transmission is completed, USBSTS.UI will be set to
0x0C	14:12	CPage, current buffer, software needs to be initialized to 0, hardware will update in real time during transmission
0x0C	11:10	multo, used for dtd to cover mult part of dqh. For example, dqh.mult=3, max_packet_size=,512. When 1023 bytes of data need to be transmitted, the software needs to set dtd.multo to 2, which will transmit 2 data packets: 512+511; If the software does not set multo(0), 3 packets will be transmitted: 512+511+0.

bytes	bits	Description
0x0C	7:0	Status, used to indicate the current dtd status. The software needs to be initialized to 0x80. Bit7: Active, which means that the current dtd is valid. Software needs to set this bit to 1 when preparing the dtd, and hardware will clear this bit after completing the transfer. Bit6: Halted. Indicates that the current endpoint is in the Halt state bit5: data buffer error. Represents buffer underrun(IN) or overrun(OUT) bit3: transaction error. Indicates that there is an error in data transmission Bit2:0 can be ignored in simple applications.
0x10	31:12	Buffer Pointer (Page 0), each dtd has 5 buffers, each 4Kbyte, which can be contiguous physical addresses (buffer[n]=buffer[n-1]+0x1000, or it can be discrete addresses. The starting address of Buffer page0 is specified by current_offset, and other buffers must be aligned in 4Kbytes. A maximum of 20 Kbytes of data can be transferred. If the starting address is not 4Kbyte aligned, it is recommended not to exceed 16 Kbytes
0x10	11:0	Current Offset, software initializes the starting offset of the packet in buffer0. During hardware transmission, it will be updated to the offset of the current buffer(specified by cpage) in real time, indicating how much data has been transmitted
0x14 to 0x23	31:12	data buffer pointer address (page1,2,3,4)
0x28 to 0x2F	31:0	setup buffer, the last 8 bytes of dqh, used to store the SETUP data packet. After receiving the SETUP packet, the 8 bytes of SETUP data will be stored in OUT dqh of endpoint 0

Table 205: dqh structure

54.5 USB register list

USB0 base address: 0xF1420000

Offset	Register Name	Description	Reset Value
0x0080	GPTIMER0LD	General Purpose Timer #0 Load Register	0x00000000
0x0084	GPTIMER0CTRL	General Purpose Timer #0 Controller Register	0x00000000
0x0088	GPTIMER1LD	General Purpose Timer #1 Load Register	0x00000000
0x008C	GPTIMER1CTRL	General Purpose Timer #1 Controller Register	0x00000000
0x0090	SBUSCFG	System Bus Config Register	0x00000000
0x0140	USBCMD	USB Command Register	0x00080000
0x0144	USBSTS	USB Status Register	0x00000000

Offset	Register Name	Description	Reset Value
0x0148	USBINTR	Interrupt Enable Register	0x00000000
0x014C	FRINDEX	USB Frame Index Register	0x00000000
0x0154	DEVICEADDR	Device Address Register	0x00000000
0x0154	PERIODICLISTBASE	Frame List Base Address Register	0x00000000
0x0158	ASYNCLISTADDR	Next Asynch. Address Register	0x00000000
0x0158	ENDPTLISTADDR	Endpoint List Address Register	0x00000000
0x0160	BURSTSIZE	Programmable Burst Size Register	0x00000000
0x0164	TXFILLTUNING	TX FIFO Fill Tuning Register	0x00000000
0x0178	ENDPTNAK	Endpoint NAK Register	0x00000000
0x017C	ENDPTNAKEN	Endpoint NAK Enable Register	0x00000000
0x0184	PORTSC1	Port Status & Control	0x00000000
0x01A4	OTGSC	On-The-Go Status & control Register	0x00000000
0x01A8	USBMODE	USB Device Mode Register	0x00000000
0x01AC	ENDPTSETUPSTAT	Endpoint Setup Status Register	0x00000000
0x01B0	ENDPTPRIME	Endpoint Prime Register	0x00000000
0x01B4	ENDPTFLUSH	Endpoint Flush Register	0x00000000
0x01B8	ENDPTSTAT	Endpoint Status Register	0x00000000
0x01BC	ENDPTCOMPLETE	Endpoint Complete Register	0x00000000
0x01C0	ENDPTCTRL[ENDPTCTRL0]	Endpoint Control0 Register... Endpoint Control7 Register	0x00000000
0x01C4	ENDPTCTRL[ENDPTCTRL1]	Endpoint Control0 Register... Endpoint Control7 Register	0x00000000
0x01C8	ENDPTCTRL[ENDPTCTRL2]	Endpoint Control0 Register... Endpoint Control7 Register	0x00000000
0x01CC	ENDPTCTRL[ENDPTCTRL3]	Endpoint Control0 Register... Endpoint Control7 Register	0x00000000
0x01D0	ENDPTCTRL[ENDPTCTRL4]	Endpoint Control0 Register... Endpoint Control7 Register	0x00000000
0x01D4	ENDPTCTRL[ENDPTCTRL5]	Endpoint Control0 Register... Endpoint Control7 Register	0x00000000
0x01D8	ENDPTCTRL[ENDPTCTRL6]	Endpoint Control0 Register... Endpoint Control7 Register	0x00000000
0x01DC	ENDPTCTRL[ENDPTCTRL7]	Endpoint Control0 Register... Endpoint Control7 Register	0x00000000
0x01E0	ENDPTCTRL[ENDPTCTRL8]	Endpoint Control0 Register... Endpoint Control7 Register	0x00000000
0x01E4	ENDPTCTRL[ENDPTCTRL9]	Endpoint Control0 Register... Endpoint Control7 Register	0x00000000

Offset	Register Name	Description	Reset Value
0x01E8	ENDPTCTRL[ENDPTCTRL10]	Endpoint Control0 Register... Endpoint Control7 Register	0x00000000
0x01EC	ENDPTCTRL[ENDPTCTRL11]	Endpoint Control0 Register... Endpoint Control7 Register	0x00000000
0x01F0	ENDPTCTRL[ENDPTCTRL12]	Endpoint Control0 Register... Endpoint Control7 Register	0x00000000
0x01F4	ENDPTCTRL[ENDPTCTRL13]	Endpoint Control0 Register... Endpoint Control7 Register	0x00000000
0x01F8	ENDPTCTRL[ENDPTCTRL14]	Endpoint Control0 Register... Endpoint Control7 Register	0x00000000
0x01FC	ENDPTCTRL[ENDPTCTRL15]	Endpoint Control0 Register... Endpoint Control7 Register	0x00000000
0x0200	OTG_CTRL0		0x00000000
0x0210	PHY_CTRL0		0x00000000
0x0214	PHY_CTRL1		0x00000000
0x0220	TOP_STATUS		0x00000000
0x0224	PHY_STATUS		0x00000000

Table 206: USB Register List

54.6 USB register description

54.6.1 GPTIMER0LD (0x80)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								GPTLD																							
N/A								RW																							
x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPTIMER0LD [31:0]

Bits	Name	Description
23-0	GPTLD	GPTLD General Purpose Timer Load Value These bit fields are loaded to GPTCNT bits when GPTRST bit is set '1b'. This value represents the time in microseconds minus 1 for the timer duration. Example: for a one millisecond timer, load 1000-1=999 or 0x0003E7. NOTE: Max value is 0xFFFFFFFF or 16.777215 seconds.

Bits	Name	Description

GPTIMER0LD Bits

54.6.2 GPTIMER0CTRL (0x84)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPTRUN	GPTRST	RSVD						GPTMODE	GPTCNT																						
RW	WO	N/A						RW	RO																						
0	0	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPTIMER0CTRL [31:0]

Bits	Name	Description
31	GPTRUN	GPTRUN General Purpose Timer Run GPTCNT bits are not effected when setting or clearing this bit. 0 - Stop counting 1 - Run
30	GPTRST	GPTRST General Purpose Timer Reset 0 - No action 1 - Load counter value from GPTLD bits in n_GPTIMER0LD
24	GPTMODE	GPTMODE General Purpose Timer Mode In one shot mode, the timer will count down to zero, generate an interrupt, and stop until the counter is reset by software; In repeat mode, the timer will count down to zero, generate an interrupt and automatically reload the counter value from GPTLD bits to start again. 0 - One Shot Mode 1 - Repeat Mode
23-0	GPTCNT	GPTCNT General Purpose Timer Counter. This field is the count value of the countdown timer.

GPTIMER0CTRL Bits

54.6.3 GPTIMER1LD (0x88)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								GPTLD																							
N/A								RW																							
x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPTIMER1LD [31:0]

Bits	Name	Description
23-0	GPTLD	GPTLD General Purpose Timer Load Value These bit fields are loaded to GPTCNT bits when GPTRST bit is set '1b'. This value represents the time in microseconds minus 1 for the timer duration. Example: for a one millisecond timer, load 1000-1=999 or 0x0003E7. NOTE: Max value is 0xFFFFF or 16.777215 seconds.

GPTIMER1LD Bits

54.6.4 GPTIMER1CTRL (0x8C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPTRUN	GPTRST	RSVD						GPTMODE	GPTCNT																						
RW	WO	N/A						RW	RO																						
0	0	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPTIMER1CTRL [31:0]

Bits	Name	Description
31	GPTRUN	GPTRUN General Purpose Timer Run GPTCNT bits are not effected when setting or clearing this bit. 0 - Stop counting 1 - Run
30	GPTRST	GPTRST General Purpose Timer Reset 0 - No action 1 - Load counter value from GPTLD bits in USB_n_GPTIMER1LD

Bits	Name	Description
24	GPTMODE	GPTMODE General Purpose Timer Mode In one shot mode, the timer will count down to zero, generate an interrupt, and stop until the counter is reset by software. In repeat mode, the timer will count down to zero, generate an interrupt and automatically reload the counter value from GPTLD bits to start again. 0 - One Shot Mode 1 - Repeat Mode
23-0	GPTCNT	GPTCNT General Purpose Timer Counter. This field is the count value of the countdown timer.

GPTIMER1CTRL Bits

54.6.5 SBUSCFG (0x90)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																												AHBBRST			
N/A																												RW			
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	

SBUSCFG [31:0]

Bits	Name	Description
2-0	AHBBRST	AHBBRST AHB master interface Burst configuration These bits control AHB master transfer type sequence (or priority). NOTE: This register overrides n_BURSTSIZE register when its value is not zero. 000 - Incremental burst of unspecified length only 001 - INCR4 burst, then single transfer 010 - INCR8 burst, INCR4 burst, then single transfer 011 - INCR16 burst, INCR8 burst, INCR4 burst, then single transfer 100 - Reserved, don't use 101 - INCR4 burst, then incremental burst of unspecified length 110 - INCR8 burst, INCR4 burst, then incremental burst of unspecified length 111 - INCR16 burst, INCR8 burst, INCR4 burst, then incremental burst of unspecified length

SBUSCFG Bits

54.6.6 USBCMD (0x140)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								ITC								FS_2	ATDTW	SUTW	PRM	ASPE	RSVD	ASP	RSVD	IAA	ASE	PSE	FS_1	RST	RS		
N/A								RW								RW	RW	RW	WO	RW	N/A	RW	N/A	RW	RW	RW	RW	RW	RW	RW	
x	x	x	x	x	x	x	x	0	0	0	0	1	0	0	0	0	0	0	0	0	x	0	0	x	0	0	0	0	0	0	0

USBCMD [31:0]

Bits	Name	Description
23-16	ITC	ITC Interrupt Threshold Control -Read/Write. The system software uses this field to set the maximum rate at which the host/device controller will issue interrupts. ITC contains the maximum interrupt interval measured in micro-frames. Valid values are shown below. Value Maximum Interrupt Interval 00000000 - Immediate (no threshold) 00000001 - 1 micro-frame 00000010 - 2 micro-frames 00000100 - 4 micro-frames 00001000 - 8 micro-frames 00010000 - 16 micro-frames 00100000 - 32 micro-frames 01000000 - 64 micro-frames
15	FS_2	FS_2 Frame List Size - (Read/Write or Read Only). [host mode only] This field is Read/Write only if Programmable Frame List Flag in the HCCPARAMS registers is set to one. This field specifies the size of the frame list that controls which bits in the Frame Index Register should be used for the Frame List Current index. NOTE: This field is made up from USBCMD bits 15, 3 and 2. Value Meaning 0b000 - 1024 elements (4096 bytes) Default value 0b001 - 512 elements (2048 bytes) 0b010 - 256 elements (1024 bytes) 0b011 - 128 elements (512 bytes) 0b100 - 64 elements (256 bytes) 0b101 - 32 elements (128 bytes) 0b110 - 16 elements (64 bytes) 0b111 - 8 elements (32 bytes)
14	ATDTW	ATDTW Add dTD TripWire - Read/Write. [device mode only] This bit is used as a semaphore to ensure proper addition of a new dTD to an active (primed) endpoint's linked list. This bit is set and cleared by software. This bit would also be cleared by hardware when state machine is hazard region for which adding a dTD to a primed endpoint may go unrecognized.

Bits	Name	Description
13	SUTW	SUTW Setup TripWire - Read/Write. [device mode only] This bit is used as a semaphore to ensure that the setup data payload of 8 bytes is extracted from a QH by the DCD without being corrupted. If the setup lockout mode is off (SLOM bit in USB core register n_USBMODE, see USBMODE) then there is a hazard when new setup data arrives while the DCD is copying the setup data payload from the QH for a previous setup packet. This bit is set and cleared by software. This bit would also be cleared by hardware when a hazard detected.
12	PRM	Asynchronous Schedule start- Write only, host mode only。 this bit is used to notify hostcontroller to start async schedule immediately.
11	ASPE	ASPE Asynchronous Schedule Park Mode Enable - Read/Write. If the Asynchronous Park Capability bit in the HCCPARAMS register is a one, then this bit defaults to a 1h and is R/W. Otherwise the bit must be a zero and is RO. Software uses this bit to enable or disable Park mode. When this bit is one, Park mode is enabled. When this bit is a zero, Park mode is disabled. NOTE: ASPE bit reset value: '0b' for OTG controller .
9-8	ASP	ASP Asynchronous Schedule Park Mode Count - Read/Write. If the Asynchronous Park Capability bit in the HCCPARAMS register is a one, then this field defaults to 3h and is R/W. Otherwise it defaults to zero and is Read-Only. It contains a count of the number of successive transactions the host controller is allowed to execute from a high-speed queue head on the Asynchronous schedule before continuing traversal of the Asynchronous schedule. Valid values are 1h to 3h. Software must not write a zero to this bit when Park Mode Enable is a one as this will result in undefined behavior. This field is set to 3h in all controller core.

Bits	Name	Description
6	IAA	IAA Interrupt on Async Advance Doorbell - Read/Write. This bit is used as a doorbell by software to tell the host controller to issue an interrupt the next time it advances asynchronous schedule. Software must write a 1 to this bit to ring the doorbell. When the host controller has evicted all appropriate cached schedule states, it sets the Interrupt on Async Advance status bit in the USBSTS register. If the Interrupt on Sync Advance Enable bit in the USBINTR register is one, then the host controller will assert an interrupt at the next interrupt threshold. The host controller sets this bit to zero after it has set the Interrupt on Sync Advance status bit in the USBSTS register to one. Software should not write a one to this bit when the asynchronous schedule is inactive. Doing so will yield undefined results. This bit is only used in host mode. Writing a one to this bit when device mode is selected will have undefined results.
5	ASE	ASE Asynchronous Schedule Enable - Read/Write. Default 0b. This bit controls whether the host controller skips processing the Asynchronous Schedule. Only the host controller uses this bit. Values Meaning 0 - Do not process the Asynchronous Schedule. 1 - Use the ASYNCLISTADDR register to access the Asynchronous Schedule.
4	PSE	PSE Periodic Schedule Enable- Read/Write. Default 0b. This bit controls whether the host controller skips processing the Periodic Schedule. Only the host controller uses this bit. Values Meaning 0 - Do not process the Periodic Schedule 1 - Use the PERIODICLISTBASE register to access the Periodic Schedule.
3-2	FS_1	FS_1 See description at bit 15

Bits	Name	Description
1	RST	RST Controller Reset (RESET) - Read/Write. Software uses this bit to reset the controller. This bit is set to zero by the Host/Device Controller when the reset process is complete. Software cannot terminate the reset process early by writing a zero to this register. Host operation mode: When software writes a one to this bit, the Controller resets its internal pipelines, timers, counters, state machines etc. to their initial value. Any transaction currently in progress on USB is immediately terminated. A USB reset is not driven on downstream ports. Software should not set this bit to a one when the HCHalted bit in the USBSTS register is a zero. Attempting to reset an actively running host controller will result in undefined behavior. Device operation mode: When software writes a one to this bit, the Controller resets its internal pipelines, timers, counters, state machines etc. to their initial value. Writing a one to this bit when the device is in the attached state is not recommended, because the effect on an attached host is undefined. In order to ensure that the device is not in an attached state before initiating a device controller reset, all primed endpoints should be flushed and the USBCMD Run/Stop bit should be set to 0.

Bits	Name	Description
0	RS	RS Run/Stop (RS) - Read/Write. Default 0b. 1=Run. 0=Stop. Host operation mode: When set to '1b', the Controller proceeds with the execution of the schedule. The Controller continues execution as long as this bit is set to a one. When this bit is set to 0, the Host Controller completes the current transaction on the USB and then halts. The HC Halted bit in the status register indicates when the Controller has finished the transaction and has entered the stopped state. Software should not write a one to this field unless the controller is in the Halted state (that is, HCHalted in the USBSTS register is a one). Device operation mode: Writing a one to this bit will cause the controller to enable a pull-up on D+ and initiate an attach event. This control bit is not directly connected to the pull-up enable, as the pull-up will become disabled upon transitioning into high-speed mode. Software should use this bit to prevent an attach event before the controller has been properly initialized. Writing a 0 to this will cause a detach event.

USBCMD Bits

54.6.7 USBSTS (0x144)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD						T11	T10	RSVD				UPI	UAI	RSVD	NAKI	AS	PS	RCL	HCH	RSVD			SLI	SRI	URI	AAI	SEI	FRI	PCI	UEI	UI
N/A						RWC	RWC	N/A				RWC	RWC	N/A	RO	RO	RO	RO	RO	N/A			RWC	RWC	RWC	RWC	RWC	RWC	RWC	RWC	RWC
x	x	x	x	x	x	0	0	x	x	x	x	0	0	x	0	0	0	0	0	x	x	x	0	0	0	0	0	0	0	0	0

USBSTS [31:0]

Bits	Name	Description
25	T11	T11 General Purpose Timer Interrupt 1(GPTINT1)–R/WC. This bit is set when the counter in the GPTIMER1CTRL register transitions to zero, writing a one to this bit will clear it.

Bits	Name	Description
24	TIO	TIO General Purpose Timer Interrupt 0(GPTINT0)–R/WC. This bit is set when the counter in the GPTIMER0CTRL register transitions to zero, writing a one to this bit clears it.
19	UPI	USB Host Periodic Interrupt –RWC. Default = 0b. This bit is set by the Host Controller when the cause of an interrupt is a completion of a USB transaction where the Transfer Descriptor (TD) has an interrupt on complete (IOC) bit set and the TD was from the periodic schedule. This bit is also set by the Host Controller when a short packet is detected and the packet is on the periodic schedule. A short packet is when the actual number of bytes received was less than expected. This bit is not used by the device controller and will always be zero.
18	UAI	USB Host Asynchronous Interrupt –RWC. Default = 0b. This bit is set by the Host Controller when the cause of an interrupt is a completion of a USB transaction where the Transfer Descriptor (TD) has an interrupt on complete (IOC) bit set AND the TD was from the asynchronous schedule. This bit is also set by the Host when a short packet is detected and the packet is on the asynchronous schedule. A short packet is when the actual number of bytes received was less than expected. This bit is not used by the device controller and will always be zero
16	NAKI	NAKI NAK Interrupt Bit–RO. This bit is set by hardware when for a particular endpoint both the TX/RX Endpoint NAK bit and corresponding TX/RX Endpoint NAK Enable bit are set. This bit is automatically cleared by hardware when all Enabled TX/RX Endpoint NAK bits are cleared.

Bits	Name	Description
15	AS	AS Asynchronous Schedule Status - Read Only. This bit reports the current real status of the Asynchronous Schedule. When set to zero the asynchronous schedule status is disabled and if set to one the status is enabled. The Host Controller is not required to immediately disable or enable the Asynchronous Schedule when software transitions the Asynchronous Schedule Enable bit in the USBCMD register. When this bit and the Asynchronous Schedule Enable bit are the same value, the Asynchronous Schedule is either enabled (1) or disabled (0). Only used in the host operation mode.
14	PS	PS Periodic Schedule Status - Read Only. This bit reports the current real status of the Periodic Schedule. When set to zero the periodic schedule is disabled, and if set to one the status is enabled. The Host Controller is not required to immediately disable or enable the Periodic Schedule when software transitions the Periodic Schedule Enable bit in the USBCMD register. When this bit and the Periodic Schedule Enable bit are the same value, the Periodic Schedule is either enabled (1) or disabled (0). Only used in the host operation mode.
13	RCL	RCL Reclamation - Read Only. This is a read-only status bit used to detect an empty asynchronous schedule. Only used in the host operation mode.

Bits	Name	Description
12	HCH	HCH HCHalted - Read Only. This bit is a zero whenever the Run/Stop bit is a one. The Controller sets this bit to one after it has stopped executing because of the Run/Stop bit being set to 0, either by software or by the Controller hardware (for example, an internal error). Only used in the host operation mode. Default value is '0b' for OTG core . This is because OTG core is not operating as host in default. Please see CM bit in USB_n_USBMODE register. NOTE: HCH bit reset value: '0b' for OTG controller core .
8	SLI	SLI DCSuspend - R/WC. When a controller enters a suspend state from an active state, this bit will be set to a one. The device controller clears the bit upon exiting from a suspend state. Only used in device operation mode.
7	SRI	SRI SOF Received - R/WC. When the device controller detects a Start Of (micro) Frame, this bit will be set to a one. When a SOF is extremely late, the device controller will automatically set this bit to indicate that an SOF was expected. Therefore, this bit will be set roughly every 1ms in device FS mode and every 125ms in HS mode and will be synchronized to the actual SOF that is received. Because the device controller is initialized to FS before connect, this bit will be set at an interval of 1ms during the prelude to connect and chirp. In host mode, this bit will be set every 125us and can be used by host controller driver as a time base. Software writes a 1 to this bit to clear it.
6	URI	URI USB Reset Received - R/WC. When the device controller detects a USB Reset and enters the default state, this bit will be set to a one. Software can write a 1 to this bit to clear the USB Reset Received status bit. Only used in device operation mode.

Bits	Name	Description
5	AAI	AAI Interrupt on Async Advance - R/WC. System software can force the host controller to issue an interrupt the next time the host controller advances the asynchronous schedule by writing a one to the Interrupt on Async Advance Doorbell bit in the n_USBCMD register. This status bit indicates the assertion of that interrupt source. Only used in host operation mode.
4	SEI	System Error -RWC. Default = 0b. In the BVCI implementation of the USBHS core, this bit is not used, and will always be cleared to '0b'. In the AMBA implementation, this bit will be set to '1b' when an Error response is seen by the master interface (HRESP[1:0]=ERROR)
3	FRI	FRI Frame List Rollover - R/WC. The Host Controller sets this bit to a one when the Frame List Index rolls over from its maximum value to zero. The exact value at which the rollover occurs depends on the frame list size. For example. If the frame list size (as programmed in the Frame List Size field of the USB_n_USBCMD register) is 1024, the Frame Index Register rolls over every time FRINDEX [13] toggles. Similarly, if the size is 512, the Host Controller sets this bit to a one every time FHINDEX [12] toggles. Only used in host operation mode.
2	PCI	PCI Port Change Detect - R/WC. The Host Controller sets this bit to a one when on any port a Connect Status occurs, a Port Enable/Disable Change occurs, or the Force Port Resume bit is set as the result of a J-K transition on the suspended port. The Device Controller sets this bit to a one when the port controller enters the full or high-speed operational state. When the port controller exits the full or high-speed operation states due to Reset or Suspend events, the notification mechanisms are the USB Reset Received bit and the DCSuspend bits Respectively.

Bits	Name	Description
1	UEI	UEI USB Error Interrupt (USBERRINT) - R/WC. When completion of a USB transaction results in an error condition, this bit is set by the Host/Device Controller. This bit is set along with the USBINT bit, if the TD on which the error interrupt occurred also had its interrupt on complete (IOC) bit set.
0	UI	UI USB Interrupt (USBINT) - R/WC. This bit is set by the Host/Device Controller when the cause of an interrupt is a completion of a USB transaction where the Transfer Descriptor (TD) has an interrupt on complete (IOC) bit set. This bit is also set by the Host/Device Controller when a short packet is detected. A short packet is when the actual number of bytes received was less than the expected number of bytes.

USBSTS Bits

54.6.8 USBINTR (0x148)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD						TIE1	TIE0	RSVD				UPIE	UAIE	RSVD	NAKE	RSVD						SLE	SRE	URE	AAE	SEE	FRE	PCE	UEE	UE	
N/A						RW	RW	N/A				RW	RW	N/A	RO	N/A						RW	RW	RW	RW	RW	RW	RW	RWC	RW	
x	x	x	x	x	x	0	0	x	x	x	x	0	0	x	0	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0

USBINTR [31:0]

Bits	Name	Description
25	TIE1	TIE1 General Purpose Timer #1 Interrupt Enable When this bit is one and the TI1 bit in n_USBSTS register is a one the controller will issue an interrupt.
24	TIE0	TIE0 General Purpose Timer #0 Interrupt Enable When this bit is one and the TI0 bit in n_USBSTS register is a one the controller will issue an interrupt.

Bits	Name	Description
19	UPIE	UPIE USB Host Periodic Interrupt Enable When this bit is one, and the UPI bit in the n_USBSTS register is one, host controller will issue an interrupt at the next interrupt threshold.
18	UAIE	UAIE USB Host Asynchronous Interrupt Enable When this bit is one, and the UAI bit in the n_USBSTS register is one, host controller will issue an interrupt at the next interrupt threshold.
16	NAKE	NAKE NAK Interrupt Enable When this bit is one and the NAKI bit in n_USBSTS register is a one the controller will issue an interrupt.
8	SLE	SLE Sleep Interrupt Enable When this bit is one and the SLI bit in n_n_USBSTS register is a one the controller will issue an interrupt. Only used in device operation mode.
7	SRE	SRE SOF Received Interrupt Enable When this bit is one and the SRI bit in n_USBSTS register is a one the controller will issue an interrupt.
6	URE	URE USB Reset Interrupt Enable When this bit is one and the URI bit in n_USBSTS register is a one the controller will issue an interrupt. Only used in device operation mode.
5	AAE	AAE Async Advance Interrupt Enable When this bit is one and the AAI bit in n_USBSTS register is a one the controller will issue an interrupt. Only used in host operation mode.
4	SEE	SEE System Error Interrupt Enable When this bit is one and the SEI bit in n_USBSTS register is a one the controller will issue an interrupt. Only used in host operation mode.

Bits	Name	Description
3	FRE	FRE Frame List Rollover Interrupt Enable When this bit is one and the FRI bit in n_USBSTS register is a one the controller will issue an interrupt. Only used in host operation mode.
2	PCE	PCE Port Change Detect Interrupt Enable When this bit is one and the PCI bit in n_USBSTS register is a one the controller will issue an interrupt.
1	UEE	UEE USB Error Interrupt Enable When this bit is one and the UEI bit in n_USBSTS register is a one the controller will issue an interrupt.
0	UE	UE USB Interrupt Enable When this bit is one and the UI bit in n_USBSTS register is a one the controller will issue an interrupt.

USBINTR Bits

54.6.9 FRINDEX (0x14C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																		FRINDEX													
N/A																		RW													
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	

FRINDEX [31:0]

Bits	Name	Description
13-0	FRINDEX	FRINDEX Frame Index. The value, in this register, increments at the end of each time frame (micro-frame). Bits [N: 3] are used for the Frame List current index. This means that each location of the frame list is accessed 8 times (frames or micro-frames) before moving to the next index. The following illustrates values of N based on the value of the Frame List Size field in the USBCMD register, when used in host mode. USBCMD [Frame List Size] Number Elements N In device mode the value is the current frame number of the last frame transmitted. It is not used as an index. In either mode bits 2:0 indicate the current microframe. The bit field values description below is represented as (Frame List Size) Number Elements N. 0000000000000000 - (1024) 12 0000000000000001 - (512) 11 0000000000000010 - (256) 10 0000000000000011 - (128) 9 0000000000000100 - (64) 8 0000000000000101 - (32) 7 0000000000000110 - (16) 6 0000000000000111 - (8) 5

FRINDEX Bits

54.6.10 DEVICEADDR (0x154)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBADR							USBADRA	RSVD																							
RW							RW	N/A																							
0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	

DEVICEADDR [31:0]

Bits	Name	Description
31-25	USBADR	USBADR Device Address. These bits correspond to the USB device address

Bits	Name	Description
24	USBADRA	USBADRA Device Address Advance. Default=0. When this bit is '0', any writes to USBADR are instantaneous. When this bit is written to a '1' at the same time or before USBADR is written, the write to the USBADR field is staged and held in a hidden register. After an IN occurs on endpoint 0 and is ACKed, USBADR will be loaded from the holding register. Hardware will automatically clear this bit on the following conditions: 1) IN is ACKed to endpoint 0. (USBADR is updated from staging register). 2) OUT/SETUP occur to endpoint 0. (USBADR is not updated). 3) Device Reset occurs (USBADR is reset to 0). NOTE: After the status phase of the SET_ADDRESS descriptor, the DCD has 2 ms to program the USBADR field. This mechanism will ensure this specification is met when the DCD can not write of the device address within 2ms from the SET_ADDRESS status phase. If the DCD writes the USBADR with USBADRA=1 after the SET_ADDRESS data phase (before the prime of the status phase), the USBADR will be programmed instantly at the correct time and meet the 2ms USB requirement.

DEVICEADDR Bits

54.6.11 PERIODICLISTBASE (0x154)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
BASEADR																				RSVD																
RW																				N/A																
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x					

PERIODICLISTBASE [31:0]

Bits	Name	Description
31-12	BASEADR	BASEADR Base Address (Low). These bits correspond to memory address signals [31:12], respectively. Only used by the host controller.

PERIODICLISTBASE Bits

54.6.12 ASYNCLISTADDR (0x158)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RW																		ASYBASE									RSVD				
																											N/A				
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x	x	x	x

ASYNCLISTADDR [31:0]

Bits	Name	Description
31-5	ASYBASE	ASYBASE Link Pointer Low (LPL). These bits correspond to memory address signals [31:5], respectively. This field may only reference a Queue Head (QH). Only used by the host controller.

ASYNCLISTADDR Bits

54.6.13 ENDPTLISTADDR (0x158)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RW																		EPBASE									RSVD				
																											N/A				
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x	x	x	x

ENDPTLISTADDR [31:0]

Bits	Name	Description
31-11	EPBASE	EPBASE Endpoint List Pointer(Low). These bits correspond to memory address signals [31:11], respectively. This field will reference a list of up to 32 Queue Head (QH) (that is, one queue head per endpoint & direction).

ENDPTLISTADDR Bits

54.6.14 BURSTSIZE (0x160)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																TXPBURST						RXPBURST									
N/A																RW						RW									
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

BURSTSIZE [31:0]

Bits	Name	Description
15-8	TXPBURST	TXPBURST Programmable TX Burst Size. Default value is determined by TXBURST bits in n_HWTXBUF. This register represents the maximum length of a the burst in 32-bit words while moving data from system memory to the USB bus.
7-0	RXPBURST	RXPBURST Programmable RX Burst Size. Default value is determined by TXBURST bits in n_HWRXBUF. This register represents the maximum length of a the burst in 32-bit words while moving data from the USB bus to system memory.

BURSTSIZE Bits

54.6.15 TXFILLTUNING (0x164)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD										TXFIFOTHRES						RSVD				TXSCHHEALTH				RSVD		TXSCHOH					
N/A										RW						N/A				RWC				N/A		RW					

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	x	x	x	0	0	0	0	0	x	0	0	0	0	0	0	0

TXFILLTUNING [31:0]

Bits	Name	Description
21-16	TXFIFOTHRES	TXFIFOTHRES FIFO Burst Threshold. (Read/Write) This register controls the number of data bursts that are posted to the TX latency FIFO in host mode before the packet begins on to the bus. The minimum value is 2 and this value should be as low as possible to maximize USB performance. A higher value can be used in systems with unpredictable latency and/or insufficient bandwidth where the FIFO may underrun because the data transferred from the latency FIFO to USB occurs before it can be replenished from system memory. This value is ignored if the Stream Disable bit in USB_n_USBMODE register is set.
12-8	TXSCHHEALTH	TXSCHHEALTH Scheduler Health Counter. (Read/Write To Clear) Table continues on the next page This register increments when the host controller fails to fill the TX latency FIFO to the level programmed by TXFIFOTHRES before running out of time to send the packet before the next Start-Of-Frame. This health counter measures the number of times this occurs to provide feedback to selecting a proper TXSCHOH. Writing to this register will clear the counter and this counter will max. at 31.

Bits	Name	Description
6-0	TXSCHOH	TXSCHOH Scheduler Overhead. (Read/Write) [Default = 0] This register adds an additional fixed offset to the schedule time estimator described above as Tff. As an approximation, the value chosen for this register should limit the number of back-off events captured in the TXSCHHEALTH to less than 10 per second in a highly utilized bus. Choosing a value that is too high for this register is not desired as it can needlessly reduce USB utilization. The time unit represented in this register is 1.267us when a device is connected in High-Speed Mode. The time unit represented in this register is 6.333us when a device is connected in Low/Full Speed Mode. Default value is '08h' for OTG controller core .

TXFILLTUNING Bits

54.6.16 ENDPTNAK (0x178)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EPTN																EPRN															
RWC																RWC															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENDPTNAK [31:0]

Bits	Name	Description
31-16	EPTN	EPTN TX Endpoint NAK - R/WC. Each TX endpoint has 1 bit in this field. The bit is set when the device sends a NAK handshake on a received IN token for the corresponding endpoint. Bit [N] - Endpoint #[N], N is 0-7
15-0	EPRN	EPRN RX Endpoint NAK - R/WC. Each RX endpoint has 1 bit in this field. The bit is set when the device sends a NAK handshake on a received OUT or PING token for the corresponding endpoint. Bit [N] - Endpoint #[N], N is 0-7

ENDPTNAK Bits

54.6.17 ENDPTNAKEN (0x17C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EPTNE																EPRNE															
RW																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENDPTNAKEN [31:0]

Bits	Name	Description
31-16	EPTNE	EPTNE TX Endpoint NAK Enable - R/W. Each bit is an enable bit for the corresponding TX Endpoint NAK bit. If this bit is set and the corresponding TX Endpoint NAK bit is set, the NAK Interrupt bit is set. Bit [N] - Endpoint #[N], N is 0-7
15-0	EPRNE	EPRNE RX Endpoint NAK Enable - R/W. Each bit is an enable bit for the corresponding RX Endpoint NAK bit. If this bit is set and the corresponding RX Endpoint NAK bit is set, the NAK Interrupt bit is set. Bit [N] - Endpoint #[N], N is 0-7

ENDPTNAKEN Bits

54.6.18 PORTSC1 (0x184)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD		STS	PTW	PSPD		RSVD	PFSC	PHCD	WKOC	WKDC	WKN	PTC				RSVD			PP	LS		HSP	PR	SUSP	FPR	OCC	OCA	PEC	PE	CSC	CCS
N/A		RW	RW	RO		N/A	RW	RW	RW	RW	RW	RW				N/A			RW	RO		RO	RW	RW	RW	RW	RO	RWC	RWC	RWC	RWC
x	x	0	0	0	0	x	0	0	0	0	0	0	0	0	0	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0

PORTSC1 [31:0]

Bits	Name	Description
29	STS	STS Serial Transceiver Select 1 Serial Interface Engine is selected 0 Parallel Interface signals is selected Serial Interface Engine can be used in combination with UTMI+/ULPI physical interface to provide FS/LS signaling instead of the parallel interface signals. When this bit is set '1b', serial interface engine will be used instead of parallel interface signals.
28	PTW	PTW Parallel Transceiver Width This bit has no effect if serial interface engine is used. 0 - Select the 8-bit UTMI interface [60MHz] 1 - Select the 16-bit UTMI interface [30MHz]
27-26	PSPD	PSPD Port Speed - Read Only. This register field indicates the speed at which the port is operating. 00 - Full Speed 01 - Low Speed 10 - High Speed 11 - Undefined
24	PFSC	PFSC Port Force Full Speed Connect - Read/Write. Default = 0b. When this bit is set to '1b', the port will be forced to only connect at Full Speed, It disables the chirp sequence that allows the port to identify itself as High Speed. 0 - Normal operation 1 - Forced to full speed

Bits	Name	Description
23	PHCD	PHCD PHY Low Power Suspend - Clock Disable (PLPSCD) - Read/Write. Default = 0b. When this bit is set to '1b', the PHY clock is disabled. Reading this bit will indicate the status of the PHY clock. NOTE: The PHY clock cannot be disabled if it is being used as the system clock. In device mode, The PHY can be put into Low Power Suspend when the device is not running (USBCMD Run/Stop=0b) or the host has signalled suspend (PORTSC1 SUSPEND=1b). PHY Low power suspend will be cleared automatically when the host initials resume. Before forcing a resume from the device, the device controller driver must clear this bit. In host mode, the PHY can be put into Low Power Suspend when the downstream device has been put into suspend mode or when no downstream device is connected. Low power suspend is completely under the control of software. 0 - Enable PHY clock 1 - Disable PHY clock
22	WKOC	WKOC Wake on Over-current Enable (WKOC_E) - Read/Write. Default = 0b. Writing this bit to a one enables the port to be sensitive to over-current conditions as wake-up events. This field is zero if Port Power(PORTSC1) is zero.
21	WKDC	WKDC Wake on Disconnect Enable (WKDSCNNT_E) - Read/Write. Default=0b. Writing this bit to a one enables the port to be sensitive to device disconnects as wake-up events. This field is zero if Port Power(PORTSC1) is zero or in device mode.
20	WKN	WKN Wake on Connect Enable (WKNNT_E) - Read/Write. Default=0b. Writing this bit to a one enables the port to be sensitive to device connects as wake-up events. This field is zero if Port Power(PORTSC1) is zero or in device mode.

Bits	Name	Description
19-16	PTC	PTC Port Test Control - Read/Write. Default = 0000b. Refer to Port Test Mode for the operational model for using these test modes and the USB Specification Revision 2.0, Chapter 7 for details on each test mode. The FORCE_ENABLE_FS and FORCE_ENABLE_LS are extensions to the test mode support specified in the EHCI specification. Writing the PTC field to any of the FORCE_ENABLE_{HS/FS/LS} values will force the port into the connected and enabled state at the selected speed. Writing the PTC field back to TEST_MODE_DISABLE will allow the port state machines to progress normally from that point. NOTE: Low speed operations are not supported as a peripheral device. Any other value than zero indicates that the port is operating in test mode. Value Specific Test 0000 - TEST_MODE_DISABLE 0001 - J_STATE 0010 - K_STATE 0011 - SE0 (host) / NAK (device) 0100 - Packet 0101 - FORCE_ENABLE_HS 0110 - FORCE_ENABLE_FS 0111 - FORCE_ENABLE_LS 1000-1111 - Reserved

Bits	Name	Description
12	PP	PP Port Power (PP)-Read/Write or Read Only. The function of this bit depends on the value of the Port Power Switching (PPC) field in the HCSPARAMS register. The behavior is as follows: PPC PP Operation 0 1b Read Only - Host controller does not have port power control switches. Each port is hard-wired to power. 1 1b/0b - Read/Write. OTG controller requires port power control switches. This bit represents the current setting of the switch (0=off, 1=on). When power is not available on a port (that is, PP equals a 0), the port is non-functional and will not report attaches, detaches, etc. When an over-current condition is detected on a powered port and PPC is a one, the PP bit in each affected port may be transitional by the host controller driver from a one to a zero (removing power from the port). This feature is implemented in all controller cores (PPC = 1).
11-10	LS	LS Line Status-Read Only. These bits reflect the current logical levels of the D+ (bit 11) and D- (bit 10) signal lines. In host mode, the use of linestate by the host controller driver is not necessary (unlike EHCI), because the port controller state machine and the port routing manage the connection of LS and FS. In device mode, the use of linestate by the device controller driver is not necessary. The encoding of the bits are: Bits [11:10] Meaning 00 - SE0 01 - K-state 10 - J-state 11 - Undefined

Bits	Name	Description
9	HSP	HSP High-Speed Port - Read Only. Default = 0b. When the bit is one, the host/device connected to the port is in high-speed mode and if set to zero, the host/device connected to the port is not in a high-speed mode. NOTE: HSP is redundant with PSPD(bit 27, 26) but remained for compatibility.
8	PR	PR Port Reset - Read/Write or Read Only. Default = 0b. In Host Mode: Read/Write. 1=Port is in Reset. 0=Port is not in Reset. Default 0. When software writes a one to this bit the bus-reset sequence as defined in the USB Specification Revision 2.0 is started. This bit will automatically change to zero after the reset sequence is complete. This behavior is different from EHCI where the host controller driver is required to set this bit to a zero after the reset duration is timed in the driver. In Device Mode: This bit is a read only status bit. Device reset from the USB bus is also indicated in the USBSTS register.

Bits	Name	Description
7	SUSP	SUSP Suspend - Read/Write or Read Only. Default = 0b. 1=Port in suspend state. 0=Port not in suspend state. In Host Mode: Read/Write. Port Enabled Bit and Suspend bit of this register define the port states as follows: Bits [Port Enabled, Suspend] Port State 0x Disable 10 Enable 11 Suspend When in suspend state, downstream propagation of data is blocked on this port, except for port reset. The blocking occurs at the end of the current transaction if a transaction was in progress when this bit was written to 1. In the suspend state, the port is sensitive to resume detection. Note that the bit status does not change until the port is suspended and that there may be a delay in suspending a port if there is a transaction currently in progress on the USB. The host controller will unconditionally set this bit to zero when software sets the Force Port Resume bit to zero. The host controller ignores a write of zero to this bit. If host software sets this bit to a one when the port is not enabled (that is, Port enabled bit is a zero) the results are undefined. This field is zero if Port Power(PORTSC1) is zero in host mode. In Device Mode: Read Only. In device mode this bit is a read only status bit.

Bits	Name	Description
6	FPR	FPR Force Port Resume -Read/Write. 1= Resume detected/driven on port. 0=No resume (K-state) detected driven on port. Default = 0. In Host Mode: Software sets this bit to one to drive resume signaling. The Host Controller sets this bit to one if a J-to-K transition is detected while the port is in the Suspend state. When this bit transitions to a one because a J-to-K transition is detected, the Port Change Detect bit in the USBSTS register is also set to one. This bit will automatically change to zero after the resume sequence is complete. This behavior is different from EHCI where the host controller driver is required to set this bit to a zero after the resume duration is timed in the driver. Note that when the Host controller owns the port, the resume sequence follows the defined sequence documented in the USB Specification Revision 2.0. The resume signaling (Full-speed 'K') is driven on the port as long as this bit remains a one. This bit will remain a one until the port has switched to the high-speed idle. Writing a zero has no effect because the port controller will time the resume operation, clear the bit the port control state switches to HS or FS idle. This field is zero if Port Power(PORTSC1) is zero in host mode. This bit is not-EHCI compatible. In Device mode: After the device has been in Suspend State for 5ms or more, software must set this bit to one to drive resume signaling before clearing. The Device Controller will set this bit to one if a J-to-K transition is detected while the port is in the Suspend state. The bit will be cleared when the device returns to normal operation. Also, when this bit will be cleared because a K-to-J transition detected, the Port Change Detect bit in the USBSTS register is also set to one.

Bits	Name	Description
5	OCC	OCC Over-current Change-R/WC. Default=0. This bit is set '1b' by hardware when there is a change to Over-current Active. Software can clear this bit by writing a one to this bit position.
4	OCA	OCA Over-current Active-Read Only. Default 0. This bit will automatically transition from one to zero when the over current condition is removed. 0 - This port does not have an over-current condition. 1 - This port currently has an over-current condition
3	PEC	PEC Port Enable/Disable Change-R/WC. 1=Port enabled/disabled status has changed. 0=No change. Default = 0. In Host Mode: For the root hub, this bit is set to a one only when a port is disabled due to disconnect on the port or due to the appropriate conditions existing at the EOF2 point (See Chapter 11 of the USB Specification). Software clears this by writing a one to it. This field is zero if Port Power(PORTSC1) is zero. In Device mode: The device port is always enabled, so this bit is always '0b'.
2	PE	PE Port Enabled/Disabled-Read/Write. 1=Enable. 0=Disable. Default 0. In Host Mode: Ports can only be enabled by the host controller as a part of the reset and enable. Software cannot enable a port by writing a one to this field. Ports can be disabled by either a fault condition (disconnect event or other fault condition) or by the host software. Note that the bit status does not change until the port state actually changes. There may be a delay in disabling or enabling a port due to other host controller and bus events. When the port is disabled, (0b) downstream propagation of data is blocked except for reset. This field is zero if Port Power(PORTSC1) is zero in host mode. In Device Mode: The device port is always enabled, so this bit is always '1b'.

Bits	Name	Description
1	CSC	CSC Connect Status Change-R/WC. 1 =Change in Current Connect Status. 0=No change. Default 0. In Host Mode: Indicates a change has occurred in the port's Current Connect Status. The host/device controller sets this bit for all changes to the port device connect status, even if system software has not cleared an existing connect status change. For example, the insertion status changes twice before system software has cleared the changed condition, hub hardware will be 'setting' an already-set bit (that is, the bit will remain set). Software clears this bit by writing a one to it. This field is zero if Port Power(PORTSC1) is zero in host mode. In Device Mode: This bit is undefined in device controller mode.
0	CCS	CCS Current Connect Status-Read Only. In Host Mode: 1=Device is present on port. 0=No device is present. Default = 0. This value reflects the current state of the port, and may not correspond directly to the event that caused the Connect Status Change bit (Bit 1) to be set. This field is zero if Port Power(PORTSC1) is zero in host mode. In Device Mode: 1=Attached. 0=Not Attached. Default=0. A one indicates that the device successfully attached and is operating in either high speed or full speed as indicated by the High Speed Port bit in this register. A zero indicates that the device did not attach successfully or was forcibly disconnected by the software writing a zero to the Run bit in the USBCMD register. It does not state the device being disconnected or Suspended.

PORTSC1 Bits

54.6.19 OTGSC (0x1A4)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD					ASVIE	AVVIE	IDIE			RSVD			ASVIS	AVVIS	IDIS			RSVD			ASV	AVV	ID		RSVD	IDPU			RSVD	VC	VD

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
N/A					RW	RW	RW	N/A					RWC	RWC	RWC	N/A					RO	RO	RO	N/A	RW	N/A			RW	RW	
x	x	x	x	x	0	0	0	x	x	x	x	x	0	0	0	x	x	x	x	x	0	0	0	x	x	0	x	x	x	0	0

OTGSC [31:0]

Bits	Name	Description
26	ASVIE	ASVIE A Session Valid Interrupt Enable - Read/Write.
25	AVVIE	AVVIE A VBus Valid Interrupt Enable - Read/Write. Setting this bit enables the A VBus valid interrupt.
24	IDIE	IDIE USB ID Interrupt Enable - Read/Write. Setting this bit enables the USB ID interrupt.
18	ASVIS	ASVIS A Session Valid Interrupt Status - Read/Write to Clear. This bit is set when VBus has either risen above or fallen below the A session valid threshold. Software must write a one to clear this bit.
17	AVVIS	AVVIS A VBus Valid Interrupt Status - Read/Write to Clear. This bit is set when VBus has either risen above or fallen below the VBus valid threshold on an A device. Software must write a one to clear this bit.
16	IDIS	IDIS USB ID Interrupt Status - Read/Write. This bit is set when a change on the ID input has been detected. Software must write a one to clear this bit.
10	ASV	ASV A Session Valid - Read Only. Indicates VBus is above the A session valid threshold.
9	AVV	AVV A VBus Valid - Read Only. Indicates VBus is above the A VBus valid threshold.
8	ID	ID USB ID - Read Only. 0 = A device, 1 = B device

Bits	Name	Description
5	IDPU	IDPU ID Pullup - Read/Write This bit provide control over the ID pull-up resistor; 0 = off, 1 = on [default]. When this bit is 0, the ID input will not be sampled.
1	VC	VC VBUS Charge - Read/Write. Setting this bit causes the VBus line to be charged. This is used for VBus pulsing during SRP.
0	VD	VD VBUS_Discharge - Read/Write. Setting this bit causes VBus to discharge through a resistor.

OTGSC Bits

54.6.20 USBMODE (0x1A8)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																										SDIS	SLOW	ES	CM		
N/A																										RW	RW	RW	RW		
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	

USBMODE [31:0]

Bits	Name	Description
4	SDIS	SDIS Stream Disable Mode. (0 - Inactive [default]; 1 - Active) Device Mode: Setting to a '1' disables double priming on both RX and TX for low bandwidth systems. This mode ensures that when the RX and TX buffers are sufficient to contain an entire packet that the standard double buffering scheme is disabled to prevent overruns/underruns in bandwidth limited systems. Note: In High Speed Mode, all packets received are responded to with a NYET handshake when stream disable is active. Host Mode: Setting to a '1' ensures that overruns/underruns of the latency FIFO are eliminated for low bandwidth systems where the RX and TX buffers are sufficient to contain the entire packet. Enabling stream disable also has the effect of ensuring the TX latency is filled to capacity before the packet is launched onto the USB. NOTE: Time duration to pre-fill the FIFO becomes significant when stream disable is active. See TXFILLTUNING and TXTTFILLTUNING [MPH Only] to characterize the adjustments needed for the scheduler when using this feature. NOTE: The use of this feature substantially limits of the overall USB performance that can be achieved.
3	SLOM	SLOM Setup Lockout Mode. In device mode, this bit controls behavior of the setup lock mechanism. See Control Endpoint Operation Model . 0 - Setup Lockouts On (default); 1 - Setup Lockouts Off. DCD requires use of Setup Data Buffer Tripwire in USBCMD.
2	ES	ES Endian Select - Read/Write. This bit can change the byte alignment of the transfer buffers to match the host microprocessor. The bit fields in the microprocessor interface and the data structures are unaffected by the value of this bit because they are based upon the 32-bit word. Bit Meaning 0 - Little Endian [Default] 1 - Big Endian

Bits	Name	Description
1-0	CM	CM Controller Mode - R/WO. Controller mode is defaulted to the proper mode for host only and device only implementations. For those designs that contain both host & device capability, the controller defaults to an idle state and needs to be initialized to the desired operating mode after reset. For combination host/device controllers, this register can only be written once after reset. If it is necessary to switch modes, software must reset the controller by writing to the RESET bit in the USBCMD register before reprogramming this register. For OTG controller core, reset value is '00b'. 00 - Idle [Default for combination host/device] 01 - Reserved 10 - Device Controller [Default for device only controller] 11 - Host Controller [Default for host only controller]

USBMODE Bits

54.6.21 ENDPTSETUPSTAT (0x1AC)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																ENDPTSETUPSTAT															
N/A																RWC															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENDPTSETUPSTAT [31:0]

Bits	Name	Description
15-0	ENDPTSETUPSTAT	ENDPTSETUPSTAT Setup Endpoint Status. For every setup transaction that is received, a corresponding bit in this register is set to one. Software must clear or acknowledge the setup transfer by writing a one to a respective bit after it has read the setup data from Queue head. The response to a setup packet as in the order of operations and total response time is crucial to limit bus time outs while the setup lock out mechanism is engaged. This register is only used in device mode.

ENDPTSETUPSTAT Bits

54.6.22 ENDPTPRIME (0x1B0)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PETB																PERB															
RWS																RWS															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENDPTPRIME [31:0]

Bits	Name	Description
31-16	PETB	PETB Prime Endpoint Transmit Buffer - R/WS. For each endpoint a corresponding bit is used to request that a buffer is prepared for a transmit operation in order to respond to a USB IN/INTERRUPT transaction. Software should write a one to the corresponding bit when posting a new transfer descriptor to an endpoint queue head. Hardware automatically uses this bit to begin parsing for a new transfer descriptor from the queue head and prepare a transmit buffer. Hardware clears this bit when the associated endpoint(s) is (are) successfully primed. NOTE: These bits are momentarily set by hardware during hardware re-priming operations when a dTD is retired, and the dQH is updated. PETB[N] - Endpoint #N, N is in 0..7

Bits	Name	Description
15-0	PERB	PERB Prime Endpoint Receive Buffer - R/WS. For each endpoint, a corresponding bit is used to request a buffer prepare for a receive operation for when a USB host initiates a USB OUT transaction. Software should write a one to the corresponding bit whenever posting a new transfer descriptor to an endpoint queue head. Hardware automatically uses this bit to begin parsing for a new transfer descriptor from the queue head and prepare a receive buffer. Hardware clears this bit when the associated endpoint(s) is (are) successfully primed. NOTE: These bits are momentarily set by hardware during hardware re-priming operations when a dTD is retired, and the dQH is updated. PERB[N] - Endpoint #N, N is in 0..7

ENDPTPRIME Bits

54.6.23 ENDPTFLUSH (0x1B4)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FETB																FERB															
RWS																RWS															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENDPTFLUSH [31:0]

Bits	Name	Description
31-16	FETB	FETB Flush Endpoint Transmit Buffer - R/WS. Writing one to a bit(s) in this register causes the associated endpoint(s) to clear any primed buffers. If a packet is in progress for one of the associated endpoints, then that transfer continues until completion. Hardware clears this register after the endpoint flush operation is successful. FETB[N] - Endpoint #N, N is in 0..7

Bits	Name	Description
15-0	FERB	FERB Flush Endpoint Receive Buffer - R/WS. Writing one to a bit(s) causes the associated endpoint(s) to clear any primed buffers. If a packet is in progress for one of the associated endpoints, then that transfer continues until completion. Hardware clears this register after the endpoint flush operation is successful. FERB[N] - Endpoint #N, N is in 0..7

ENDPTFLUSH Bits

54.6.24 ENDPTSTAT (0x1B8)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETBR																ERBR															
RO																RO															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENDPTSTAT [31:0]

Bits	Name	Description
31-16	ETBR	ETBR Endpoint Transmit Buffer Ready – Read Only. One bit for each endpoint indicates status of the respective endpoint buffer. This bit is set to one by the hardware as a response to receiving a command from a corresponding bit in the ENDPTPRIME register. There is always a delay between setting a bit in the ENDPTPRIME register and endpoint indicating ready. This delay time varies based upon the current USB traffic and the number of bits set in the ENDPRIME register. Buffer ready is cleared by USB reset, by the USB DMA system, or through the ENDPTFLUSH register. NOTE: These bits are momentarily cleared by hardware during hardware endpoint re-priming operations when a dTD is retired, and the dQH is updated. ETBR[N] - Endpoint #N, N is in 0..7

Bits	Name	Description
15-0	ERBR	ERBR Endpoint Receive Buffer Ready – Read Only. One bit for each endpoint indicates status of the respective endpoint buffer. This bit is set to a one by the hardware as a response to receiving a command from a corresponding bit in the ENDPRIME register. There is always a delay between setting a bit in the ENDPRIME register and endpoint indicating ready. This delay time varies based upon the current USB traffic and the number of bits set in the ENDPRIME register. Buffer ready is cleared by USB reset, by the USB DMA system, or through the ENDPTFLUSH register. NOTE: These bits are momentarily cleared by hardware during hardware endpoint re-priming operations when a dTD is retired, and the dQH is updated. ERBR[N] - Endpoint #N, N is in 0..7

ENDPTSTAT Bits

54.6.25 ENDPTCOMPLETE (0x1BC)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETCE																ERCE															
RWC																RWC															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENDPTCOMPLETE [31:0]

Bits	Name	Description
31-16	ETCE	ETCE Endpoint Transmit Complete Event - R/WC. Each bit indicates a transmit event (IN/INTERRUPT) occurred, and software should read the corresponding endpoint queue to determine the endpoint status. If the corresponding IOC bit is set in the Transfer Descriptor, then this bit is set simultaneously with the USBINT . Writing one clears the corresponding bit in this register. ETCE[N] - Endpoint #N, N is in 0..7

Bits	Name	Description
15-0	ERCE	ERCE Endpoint Receive Complete Event - RW/C. Each bit indicates a received event (OUT/SETUP) occurred and software should read the corresponding endpoint queue to determine the transfer status. If the corresponding IOC bit is set in the Transfer Descriptor, then this bit is set simultaneously with the USBINT . Writing one clears the corresponding bit in this register. ERCE[N] - Endpoint #N, N is in 0..7

ENDPTCOMPLETE Bits

54.6.26 ENDPTCTRL (0x1C0 + 0x4 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								TXE	TXR	RSVD		TXT		RSVD	TXS	RSVD								RXE	RXR	RSVD		RXT		RSVD	RXS
N/A								RW	WS	N/A		RW		N/A	RW	N/A								RW	WS	N/A		RW		N/A	RW
x	x	x	x	x	x	x	x	0	0	x	x	0	0	x	0	x	x	x	x	x	x	x	x	0	0	x	x	0	0	x	0

ENDPTCTRL [31:0]

Bits	Name	Description
23	TXE	TXE TX Endpoint Enable 0 Disabled [Default] 1 Enabled An Endpoint should be enabled only after it has been configured.
22	TXR	TXR TX Data Toggle Reset (WS) Write 1 - Reset PID Sequence Whenever a configuration event is received for this Endpoint, software must write a one to this bit in order to synchronize the data PID's between the Host and device.
19-18	TXT	TXT TX Endpoint Type - Read/Write 00 Control 01 Isochronous 10 Bulk 11 Interrupt

Bits	Name	Description
16	TXS	TXS TX Endpoint Stall - Read/Write 0 End Point OK 1 End Point Stalled This bit will be cleared automatically upon receipt of a SETUP request if this Endpoint is configured as a Control Endpoint and this bit will continue to be cleared by hardware until the associated ENDPTSETUPSTAT bit is cleared. Software can write a one to this bit to force the endpoint to return a STALL handshake to the Host. This control will continue to STALL until this bit is either cleared by software or automatically cleared as above for control endpoints. NOTE: [CONTROL ENDPOINT TYPES ONLY]: there is a slight delay (50 clocks max) between the ENDPTSETUPSTAT begin cleared and hardware continuing to clear this bit. In most systems, it is unlikely the DCD software will observe this delay. However, should the DCD observe that the stall bit is not set after writing a one to it then follow this procedure: continually write this stall bit until it is set or until a new setup has been received by checking the associated endptsetupstat Bit.
7	RXE	RXE RX Endpoint Enable 0 Disabled [Default] 1 Enabled An Endpoint should be enabled only after it has been configured.
6	RXR	RXR RX Data Toggle Reset (WS) Write 1 - Reset PID Sequence Whenever a configuration event is received for this Endpoint, software must write a one to this bit in order to synchronize the data PID's between the host and device.
3-2	RXT	RXT RX Endpoint Type - Read/Write 00 Control 01 Isochronous 10 Bulk 11 Interrupt

Bits	Name	Description
0	RXS	RXS RX Endpoint Stall - Read/Write 0 End Point OK. [Default] 1 End Point Stalled This bit is set automatically upon receipt of a SETUP request if this Endpoint is configured as a Control Endpoint and this bit will continue to be cleared by hardware until the associated ENDPTSETUPSTAT bit is cleared. Software can write a one to this bit to force the endpoint to return a STALL handshake to the Host. This control will continue to STALL until this bit is either cleared by software or automatically cleared as above for control endpoints. NOTE: [CONTROL ENDPOINT TYPES ONLY]: there is a slight delay (50 clocks max) between the ENDPTSETUPSTAT begin cleared and hardware continuing to clear this bit. In most systems, it is unlikely the DCD software will observe this delay. However, should the DCD observe that the stall bit is not set after writing a one to it then follow this procedure: continually write this stall bit until it is set or until a new setup has been received by checking the associated endptsetupstat Bit.

ENDPTCTRL Bits

54.6.27 OTG_CTRL0 (0x200)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD						OTG_WKDPDMCHG_EN	RSVD					AUTORESUME_EN	RSVD	OTG_VBUS_WAKEUP_EN	OTG_ID_WAKEUP_EN	RSVD		OTG_VBUS_SOURCE_SEL	OTG_UTMI_SUSPENDM_SW	OTG_UTMI_RESET_SW	OTG_WAKEUP_INT_ENABLE	OTG_POWER_MASK	OTG_OVER_CUR_POL	OTG_OVER_CUR_DIS	RSVD		SER_MODE_SUSPEND_EN	RSVD			
N/A						RW	N/A					RW	N/A	RW	RW	N/A		RW	RW	RW	RW	RW	RW	RW	N/A		RW	N/A			
x	x	x	x	x	x	0	x	x	x	x	x	0	x	0	0	x	x	0	0	0	0	0	0	0	x	x	0	x	x	x	x

OTG_CTRL0 [31:0]

Bits	Name	Description
25	OTG_WKDPDMC HG_EN	
19	AUTORESUME_E N	
17	OTG_VBUS_WAK EUP_EN	
16	OTG_ID_WAKEU P_EN	
13	OTG_VBUS_SOU RCE_SEL	
12	OTG_UTMI_SUSP ENDM_SW	default 0 for naneng usbphy
11	OTG_UTMI_RESE T_SW	default 1 for naneng usbphy
10	OTG_WAKEUP_I NT_ENABLE	
9	OTG_POWER_MA SK	
8	OTG_OVER_CUR _POL	
7	OTG_OVER_CUR _DIS	
4	SER_MODE_SUS PEND_EN	for naneng usbphy, only switch to serial mode when suspend

OTG_CTRL0 Bits

54.6.28 PHY_CTRL0 (0x210)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD						GPIO_ID_SEL_N	RSVD										ID_DIG_OVERRIDE	SESS_VALID_OVERRIDE	VBUS_VALID_OVERRIDE	OP_MODE_SUSPENDM_ENJ	RSVD							ID_DIG_OVERRIDE_EN	SESS_VALID_OVERRIDE_EN	VBUS_VALID_OVERRIDE_EN	
N/A						RW	N/A										RW	RW	RW	RW	N/A							RW	RW	RW	
x	x	x	x	x	x	0	x	x	x	x	x	x	x	x	x	x	0	0	0	0	x	x	x	x	x	x	x	x	0	0	0

PHY_CTRL0 [31:0]

Bits	Name	Description
25	GPIO_ID_SEL_N	
14	ID_DIG_OVERRID E	
13	SESS_VALID_OV ERRIDE	
12	VBUS_VALID_OV ERRIDE	
11	OP_MODE_SUSP ENDM_ENJ	set op_mode to 2'b01 in suspend, for naneng usbphy
2	ID_DIG_OVERRID E_EN	
1	SESS_VALID_OV ERRIDE_EN	
0	VBUS_VALID_OV ERRIDE_EN	

PHY_CTRL0 Bits

54.6.29 PHY_CTRL1 (0x214)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD											UTMI_CFG_RST_N	RSVD																UTMI_OTG_SUSPENDM	RSVD		
N/A											RW	N/A																RW	N/A		
x	x	x	x	x	x	x	x	x	x	x	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	x	

PHY_CTRL1 [31:0]

Bits	Name	Description
20	UTMI_CFG_RST_N	
1	UTMI_OTG_SUSP ENDM	OTG suspend, not utmi_suspendm. setting this bit also enable host high-speed disconnect detection logic(check DP/DM voltage at end of SOF, to determine whether 50Ohm are still there on DP/DM) should clear this bit before entering into suspend state(setting portsc.phcd), avoid unwanted glitch on vbus/sess_vld/sess_end

Bits	Name	Description

PHY_CTRL1 Bits

54.6.30 TOP_STATUS (0x220)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WAKEUP_INT_STATUS	RSVD																														
RW	N/A																														
0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	

TOP_STATUS [31:0]

Bits	Name	Description
31	WAKEUP_INT_STATUS	

TOP_STATUS Bits

54.6.31 PHY_STATUS (0x224)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0							
UTMI_CLK_VALID	RSVD																								LINE_STATE		HOST_DISCONNECT		ID_DIG		RSVD		UTMI_SESS_VALID		RSVD		VBUS_VALID	
	N/A																								RW		RW		RW		N/A		RW		N/A		RW	
	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	x	0	x	0							

PHY_STATUS [31:0]

Bits	Name	Description
31	UTMI_CLK_VALID	
7-6	LINE_STATE	
5	HOST_DISCONNECT	
4	ID_DIG	

Bits	Name	Description
2	UTMI_SESS_VALI D	
0	VBUS_VALID	

PHY_STATUS Bits

55 Ethernet Controller ENET

Ethernet is a computer local area network (LAN) technology. The IEEE 802.3 standard, developed by the IEEE organization, specifies the technical standards for Ethernet, including physical layer connections, electrical signals, and medium access control layer protocols.

The Ethernet controller is used to connect to the physical layer, send and receive data on an Ethernet network. The CPU can control the ENET via the APB bus. The ENET has its own built-in DMA and can access system memory via the AXI bus to achieve data reception and transmission. The physical layer (PHY) interfaces supported by ENET are MII (not supported before 6E00), RMII, and RGMII.

55.1 Feature Summary

This chapter introduces the main features of ENET:

- Supports MII (10M/100M), RMII (10M/100M), and RGMII (10M/100M/1000M).
- Supports precise clock control (IEEE 1588-2008 protocol).
- Supports Energy Efficient Ethernet (EEE), IEEE 802.3az-2010.
- Supports PPS output, which can be used for system synchronization.
- Built-in DMA, enabling data exchange between ENET and system memory via AXI.
- Supports MDIO for configuring external PHYs.
- In RGMII mode, the transmit clock delay can be configured at the PHY side or the ENET side.

55.2 ENET Module Structure

The structure of this module is shown in the figure below. The top-level interfaces include: an AXI bus interface for DMA data transmission and reception; an APB bus interface for configuring ENET module registers; MII, RMII, or GMII interfaces for data exchange with the external PHY; and an MDIO interface for configuring the PHY chip.

During data transmission, data is sent from system memory to a dedicated memory (TX_FIFO) by the TX_DMA engine within the DMA sub-controller, then passed through the MAC sub-controller and sent to the external PHY via the media interface.

During data reception, data received through the MII interface is passed through the MAC sub-controller and sent to a dedicated memory (RX_FIFO). The RX_DMA engine of the DMA sub-controller then transfers the received data to system memory.

55.3 MAC Data Packet Description

55.3.1 Ethernet ii Ethernet Frame Structure

Ethernet ii frames, also known as Ethernet V2 frames, are the most common type of Ethernet frames in today's LANs.

55.3.1.1 Frame Format

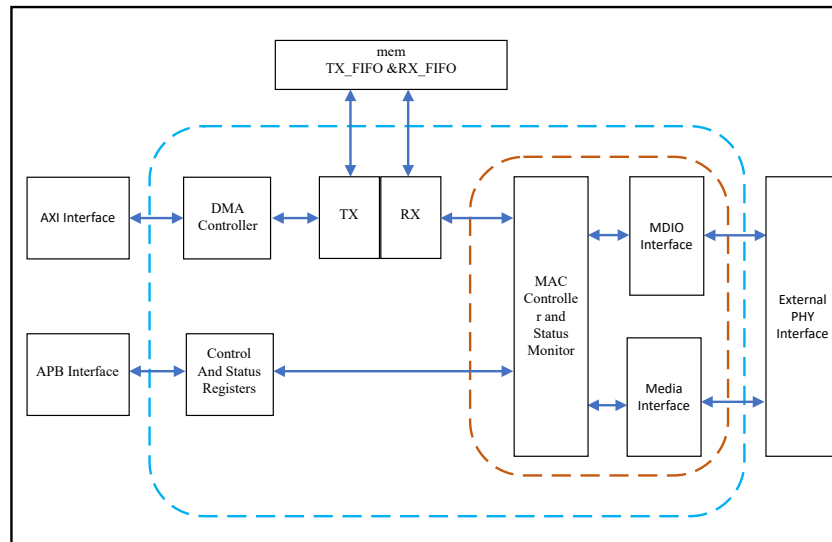


Figure 86: ENET Module Structure Diagram

Format	Preamble	DMAC	SMAC	Type	Data	FCS
Field Name	Preamble	Destination MAC Address	Source MAC Address	Type	Data Payload	Checksum
Bytes Occupied	8	6	6	2	46-1500	4

Table 207: Ethernet ii Frame Format

55.3.1.2 Description of Each Field

Preamble: Used for data synchronization; 8 fixed values of 0b10101010.

Destination MAC Address: 6 bytes for IPv4; this field identifies the frame's receiver.

Source MAC Address: 6 bytes for IPv4; this field identifies the frame's sender.

Type: Protocol type. 0x0800 indicates IPv4, 0x08DD indicates IPv6. For other details, please refer to relevant IEEE specifications.

Data Payload: The data field, representing the frame's payload (may include padding). The minimum length of the data field must be 46 bytes to ensure the frame length is at least 64 bytes. This means that even to transmit 1 byte of information, a 46-byte data field must be used. If the information placed in this field is less than 46 bytes, the rest of the field must also be padded. The maximum length of the data field is 1500 bytes. If the payload length is not an integer number of bytes, padding fields must be inserted to ensure the data frame length is an integer number of bytes.

Checksum: Also known as the Frame Check Sequence (FCS), it provides the receiver with a method to determine if a transmission error has occurred. If an error is detected, the frame is discarded. FCS is a general term; specific FCS can be further subdivided into various error-checking methods. In Ethernet frames, FCS typically uses Cyclical Redundancy Check (CRC).

55.3.2 IEEE 802.3 Type

55.3.2.1 Frame Format

Format	Preamble	SFD	DMAC	SMAC	Type	Data	FCS
Field Name	Preamble	Start Frame Delimiter	Destination MAC Address	Source MAC Address	Type	Data Payload	Checksum
Bytes Occupied	7	1	6	6	2	46-1500	4

Table 208: 802.3 Basic Frame Format

Format	Preamble	SFD	DMAC	SMAC	Tag	Type	Data	FCS
Field Name	Preamble	Start Frame Delimiter	Destination MAC Address	Source MAC Address	Tag	Type	Data Payload	Checksum
Bytes Occupied	7	1	6	6	4	2	42-1500	4

Table 209: 802.3 Frame Format with Tag

55.3.2.2 Description of Each Field

Tag Field: Divided into TPID and TCI parts. The IEEE 802.1Q protocol specifies that the TPID value is fixed at 0x8100. If a device that does not support 802.1Q receives such a frame, it will be discarded. TCI consists of three parts: Frame Priority (use priority), with values from 0 to 7; MAC Format (canonical), where 1 indicates classic format, otherwise non-classic format; and VLAN ID.

The meanings of other fields are the same as described in the 802.3 frame format above.

55.4 ENET Module Port Signal Description

55.4.1 MII Interface

Signal Name	I/O	Signal Description
TXCLK	I	Transmit Data Clock Line. This is a continuous clock providing timing reference for the MAC layer to transmit TXEN, TXD, and TXER signals to the PHY. In MII interface, TXCLK is provided by the PHY.
TXD0	O	Transmit Data Line
TXD1	O	Transmit Data Line
TXD2	O	Transmit Data Line
TXD3	O	Transmit Data Line

Signal Name	I/O	Signal Description
TXER	O	Transmit Error Signal Line. When this line is active, the transmit data is invalid.
TXEN	O	Transmit Enable. When transmitting data, this signal line is set high, indicating the transmit data is valid.
RXCLK	I	RXCLK is a continuous clock providing timing reference for the RXDV, RXD, and RXER signals transmitted from the PHY to the MAC layer. RXCLK is provided by the PHY.
RXD0	I	Receive Data Line
RXD1	I	Receive Data Line
RXD2	I	Receive Data Line
RXD3	I	Receive Data Line
RXER	I	Receive Error Signal Line. When this line is active, the receive data is invalid.
RXDV	I	Receive Data Valid Signal Line, driven by the PHY. Data on RXD is synchronized with RXCLK; RXDV must encompass the frame, starting no later than the Start Frame Delimiter (SFD) and excluding any End of Frame delimiter.
CRS	I	Carrier Sense Signal Line
COL	I	Collision Detection Signal Line
MDC	O	Management Data Clock Line
MDIO	I/O	Management Data Line, used for configuring the external PHY chip.

Table 210: MII Interface Signal Description

55.4.2 RMII Interface

Signal Name	I/O	Signal Description
REFCLK	I	Can be an external crystal input as the reference clock, or use the clock generated by the internal PLL of the chip.
TXD0	O	Transmit Data Line
TXD1	O	Transmit Data Line
TXEN	O	Transmit Enable. When transmitting data, this signal line is set high, indicating the transmit data is valid.
RXD0	I	Receive Data Line
RXD1	I	Receive Data Line
CRSDV	I	Combined from the RXDV and CRS signals of the MII interface. When the medium is not idle, CRSDV is given asynchronously with REFCLK.
MDC	O	Management Data Clock Line
MDIO	I/O	Management Data Line, used for configuring the external PHY chip.

Signal Name	I/O	Signal Description

Table 211: RMII Interface Signal Description

55.4.3 RGMII Interface

Signal Name	I/O	Signal Description
TXCLK	O	Transmit Data Clock
TXEN	O	Transmit Enable. When transmitting data, this signal line is set high, indicating the transmit data is valid.
TXD0	O	Transmit Data Line
TXD1	O	Transmit Data Line
TXD2	O	Transmit Data Line
TXD3	O	Transmit Data Line
RXCLK	I	Receive Data Clock
RXD0	I	Receive Data Line
RXD1	I	Receive Data Line
RXD2	I	Receive Data Line
RXD3	I	Receive Data Line
RXEN	I	Receive Enable
CRS	I	Carrier Sense
COL	O	Collision Detection Signal Line
MDC	O	Management Data Clock Line
MDIO	I/O	Management Data Line, used for configuring the external PHY chip.

Table 212: RGMII Interface Signal Description

55.4.4 MDIO Interface Introduction

The MAC controller accesses the PHY chip via the MDIO interface to configure the PHY chip or obtain its operating status.

The MDIO interface field supported by the MAC controller consists of a Start of Frame, Opcode, PHY Device Address, TA (Turn Around), and DATA fields. For the DATA field, it is driven by the MAC controller during write operations and by the PHY chip during read operations. The following describes the fields of MDIO write and read operation data packets.

Field	PRE	ST	OP	PHYADD	ERGAD	TA	DATA
Read Operation	32 bits of 1	'b01	'b10	PHY Chip Address	Read Register Address	0bZ0	Data

Field	PRE	ST	OP	PHYADD	ERGAD	TA	DATA
Write Operation	32 bits of 1	'b01	'b01	PHY Chip Address	Read Register Address	0b10	Data

Table 213: MDIO Interface Data Packet Field Description

- PRE: Preamble. Before sending each frame, the MAC controller continuously sends 32 MDC cycles of high level on MDIO, while outputting 32 clock cycles on MDC.
- ST: Start of Frame, 2' b01 indicates the start of frame, occupying two bits.
- OP: Opcode, occupying two bits. 2' b10 indicates read, 2' b01 indicates write.
- PHYADD: PHY Address. Occupies 5 bits, so one MDIO can manage up to 32 PHYs.
- ERGAD: Register Address. There are 32 register addresses in total. The protocol defines the first 16 register addresses in detail. Address 0 is the control register, address 1 is the status register, and the other 14 addresses are extended registers.
- TA (Turn around): TA is the idle time between address transmission and data transmission.
- DATA: Data Field, occupies 16 bits. During write operations, the MAC uses it to send write data. During read operations, the external PHY chip uses it to send read data.
- IDLE: Idle Field. When no MDIO frame is being sent, the MDIO interface outputs high impedance (externally pulled up, seen as high level on the bus; this field is not listed in the table above).

55.5 ENET Functional Description

55.5.1 Built-in DMA Controller Features

This module has a built-in DMA with independent transmit engine (TX_DMA), receive engine (RX_DMA), and independent control and status registers. When transmitting Ethernet packets, the transmit engine transfers data from system memory to the TX_FIFO, then sends it out. When receiving Ethernet packets, the receive engine transfers packets from the RX_FIFO to system memory. The DMA control module can efficiently transfer data between the FIFO and system memory at high speed without any CPU intervention. Additionally, the DMA controller can be programmed to generate CPU interrupts upon transmission completion, reception completion, or other specified conditions. The DMA controller can cache received data in other memories and transfer data from other memories to the MAC controller, but these operations require pointers pointing to the data transfer locations. The DMA controller internally has two pointers: one for describing the receive data buffer address, and one for describing the transmit data address. Among these, the base address pointer for receive data is set in the register (0x100c DMA_RX_DESC_LISTADDR), and the base address pointer for transmit data is set in the register (0x1010 DMA_TX_DESC_LISTADDR).

55.5.1.1 Burst Access

The built-in DMA controller performs burst transfers via the AXI bus interface. When the [FB] bit in the register (0x1000 DMA_BUS_MODE) is set to 1, the DMA burst transfer data length is fixed. The maximum burst transfer length for the DMA controller is set in the [PBL] and [RPBL] bits of the register (0x1000 DMA_BUS_MODE). TX_DMA will only initiate data transfer when the space in the TX_FIFO can accommodate the burst frame data or the number of bytes at the end of the Ethernet frame transmission. Then TX_DMA indicates the start address

and the required number of transfers to the AXI bus. RX_DMA will only initiate data transfer when there is enough data in the RX_FIFO for the configured burst access, or when the RX_FIFO detects the end of a frame (frame length is less than the configured burst transfer length). RX_DMA will indicate the start address and the required number of transfers to the AXI bus.

55.5.1.2 Error Access

For any data transfer initiated by the DMA controller, if the slave returns an error response, the corresponding DMA will stop all operations and update the Bus Error [EB] bit (0x1014 DMA_STATUS[EB]) and the Fatal Bus Error FBI bit (0x1014 DMA_STATUS[FBI]) in the DMA Status Register. After a DMA data transfer error occurs, the DMA controller can only resume operation after a soft reset or hard reset of the peripheral and reinitialization of the DMA.

55.5.2 DMA Controller Receive Function

When the MAC detects an SFD on the media receive interface, it initiates the receive operation. The MAC sub-controller removes the preamble and start frame delimiter before continuing to process the frame. The address field is checked for filtering, and the FCS field is used to verify the frame's CRC. If the frame fails the address filter, it is discarded.

When the MAC controller receives an Ethernet frame, it first removes the preamble and start frame delimiter. If the MAC controller receives a start frame delimiter, it transmits the data following the delimiter to the destination address. If IEEE 1588 timestamping is enabled, when the media interface detects the start frame delimiter of any frame, the RPR system will take a snapshot of the system time. Unless the MAC filters and discards this frame, this timestamp will be transmitted to the application.

Setting the [ACS] bit in the register (0x0000 MACCFG[ACS]) allows automatic removal of PAD or CRC fields when the received data length is less than 1536 bytes. The MAC controller sends data to the RX_FIFO, then discards the FCS field. If this setting is not enabled, for frames with a length of approximately 1536 bytes or more, the MAC controller will transmit all received Ethernet frames to the RX_FIFO.

At the end of Ethernet frame reception, the MAC outputs the receive status, which is sent via RX_DMA to the receive descriptor RDES0 for application reference and query.

55.5.2.1 DMA Receive Error

If the RX_FIFO becomes full before the MAC receives the end-of-frame (EOF, i.e., FCS) field, an overflow is declared and the entire frame is discarded. Simultaneously, the counter indicated by the ONFCNTOVF bit (0x1020 DMA_MISS_OVF_CNT) in the DMA Frame Loss Statistics Register will increment. Due to overflow, the status will indicate that this is a partial frame. If the FEF bit or FUF bit (DMA_OP_MODE[FEF] or DMA_OP_MODE[FUF]) in the DMA Operation Mode Register (0x1018 DMA_OP_MODE) is enabled, the RX_FIFO can filter error frames and undersized frames. In threshold mode, when reading the preamble and start frame delimiter from the RX_FIFO, the basic information of the current frame can be obtained. By comparing with the basic settings of the receive descriptor, it can determine if the frame is an error frame and discard the entire error frame. In store-and-forward mode, all error frames can be filtered and discarded before data transfer.

55.5.2.2 DMA Receive CRC Handling

The MAC controller will calculate the 32-bit CRC for the received frame (including the destination MAC address field to the FCS field) and check for any CRC errors in the received data frame. Regardless of whether automatic CRC/PAD removal is enabled (controlled by setting the ACS bit in the MAC Control Register (0x0000 MACCFG[ACS])), the MAC will receive the entire frame to calculate the CRC of the received frame and feed the result back to the application via the CRE bit in the receive descriptor RDES0.

55.5.2.3 DMA Receive Process Suspend

If a new receive data frame arrives while the receive process is suspended, the RX_DMA will re-fetch the current descriptor from system memory. If the descriptor is now owned by the RX_DMA, the receive process will re-enter the running state and start receiving the frame. If the descriptor is still owned by the host, by default, the RX_DMA will discard the current frame at the top of the RX_FIFO and increment the lost frame counter. If multiple frames are stored in the RX_FIFO, this process will be repeated. If the [DFF] bit in the DMA Operation Mode Register (0x1018 DMA_OP_MODE) is set to 1, discarding or flushing the frame at the top of the RX_FIFO can be avoided. In this case, the receive process sets the receive buffer to an unavailable state and returns to the suspended state.

55.5.3 DMA Controller Transmit Function

During continuous frame transmission, the handling timing of the second frame by TX_DMA differs depending on the setting of the [OSF] bit in the DMA Operation Mode Register (0x1018 DMA_OP_MODE). When the OSF bit is set to 1, after completing the transmission of the first frame, the transmit process immediately polls the transmit descriptor list for the second frame. If the second frame is valid, the transmit process sends the second frame before writing the status information of the first frame. This method, where the transmit process can acquire two frames simultaneously without closing the status descriptor of the first frame, is called OSF mode; otherwise, it is non-OSF mode.

In non-OSF mode (i.e., default mode), the DMA transmit operation sequence is as follows:

1. After setting up the corresponding Ethernet frame in the data buffer, the user configures the transmit descriptor (TDES0-TDES3) and sets the OWN bit to 1.
2. After the STT bit in the DMA Operation Mode Register (0x1018 DMA_OP_MODE[ST]) is set to 1, TX_DMA immediately enters the running state.
3. In the running state, TX_DMA polls the transmit descriptor list for frames that need to be transmitted. Once polling starts, TX_DMA continues in a continuous ring or linked order of descriptors. If TX_DMA detects a descriptor marked as CPU-owned (TDES0.OWN=0), or an error condition occurs, it suspends transmission and sets the Transmit Buffer Unavailable bit (the [RU] bit in the DMA Status Register (0x1014 DMA_STATUS[RU]) and the [AIS] bit (0x1014 DMA_STATUS[AIS]) to 1. The transmit engine then proceeds to step 9.
4. If the obtained descriptor is marked as TX_DMA-owned (TDES0.OWN=1), then TX_DMA decodes the transmit data buffer address based on the obtained descriptor.
5. TX_DMA fetches transmit data from system memory and transmits this data.
6. If the Ethernet frame is stored in data buffers of multiple descriptors, TX_DMA will close the intermediate descriptor and fetch the next descriptor. Steps 3, 4, and 5 are repeated until the Ethernet frame data transmission

is complete.

7. After frame transmission is complete, status information is written to the respective status bits (TDES0) of that transmit descriptor. In this step, the OWN bit is cleared to 0, making the descriptor host-owned.

8. After frame transmission is complete, when the Last Descriptor bit (TDES0.TLS) of its last descriptor is set to 1, the TI bit in the DMA Status Register (0x1014 DMA_STATUS[TI]) will be set to 1. Subsequently, the TX_DMA engine returns to step 3.

9. In the suspended state, TX_DMA will attempt to re-fetch the descriptor (return to step 3) upon receiving a transmit poll request and clear the underflow interrupt status bit.

In OSF mode, the DMA transmit operation sequence is as follows:

1. The operation of TX_DMA is the same as steps 1 6 of TX_DMA in OSF=0 mode.
2. TX_DMA can fetch the next descriptor without closing the last descriptor of the previous frame.
3. If TX_DMA owns the required descriptor, it decodes the transmit buffer address in that descriptor. If TX_DMA does not own the descriptor, it enters suspend mode and jumps to step 7.
4. TX_DMA fetches the transmit frame from system memory and transmits it until the frame data transmission ends. If the frame is split across multiple descriptors, intermediate descriptors are closed simultaneously.
5. TX_DMA waits for the transmit status of the previous frame. When the status is available, TX_DMA writes the status to the corresponding TDES0, thereby closing the descriptor.
6. TX_DMA fetches the next descriptor, then continues to step 3 (when status is normal). If the previous transmit status indicates an underflow error, TX_DMA enters suspend mode (step 7).
7. In suspend mode, if TX_DMA receives a suspend status, it writes the status to the corresponding TDES0. Afterwards, the related interrupt is set to 1 and it returns to suspend mode.
8. TX_DMA will only exit suspend mode and enter the running state (go to step 1 or step 2, depending on the suspend status) upon receiving a transmit poll request (by writing to the DMA Transmit Poll Register (DMA_TX_POLL_DEMAND)).

55.5.3.1 DMA Transmit Frame Processing

The data expected by TX_DMA buffer contains the complete Ethernet frame (excluding preamble, padding bytes, and FCS field). The destination MAC address, source MAC address, and Type/Length fields belong to valid data. If the transmit descriptor indicates that the MAC must disable CRC or PAD insertion, the buffer must contain the complete Ethernet frame (excluding preamble), including CRC bytes. Frame data can be chained and can span multiple buffers.

55.5.4 MAC Control Transmit Function

The TX_DMA engine controls the transmit transaction, moving the Ethernet packet data from a pre-set system memory address to the TX_FIFO, which is then sent by the MAC controller. After transmission is complete, the transmit status is fed back to the TX_DMA controller.

After the transmit application is started, the time at which the MAC controller sends data packets to the media interface is presetable and can be configured via the [IFG] bit in the register (0x0000 MACCFG).

If a normal collision occurs during transmission (in half-duplex mode), the MAC sub-controller will validate the

transmit status, then accept and discard all subsequent data until the next preamble and start frame delimiter are received. Upon detecting a retry request from the MAC (in the status), the same frame should be retransmitted from the preamble and start frame delimiter.

If data is not continuously provided during transmission, the MAC will issue an underflow status. During normal frame transmission, if the MAC controller receives a preamble and start frame delimiter without having obtained the FCS field of the previous frame, it will ignore that preamble and start frame delimiter and treat the new frame as a continuation of the previous frame.

The DMA transmit controller delivers data in two operational modes:

First, Threshold Mode:

In this mode, data is ready and forwarded to the MAC sub-controller as long as the number of bytes in the TX_FIFO exceeds the configured threshold (or the end of frame is written before exceeding the threshold). This threshold can be configured via the TTC bit in the register (0x1018 DMA_OP_MODE).

Second, Store-and-Forward Mode:

In this mode, frames are only delivered to the MAC core after a complete frame is stored in the TX_FIFO. If the TX_FIFO size is smaller than the Ethernet frame to be sent, the frame is delivered to the MAC core when the TX_FIFO is nearly full. The entire contents of the TX_FIFO can be flushed by setting the [FTF] bit in the register (0x1018 DMA_OP_MODE) to 1. This bit is automatically cleared by hardware, and the FIFO pointer is initialized to the default state. If this bit is set to 1 while a frame is being transferred to the MAC core, the transfer stops, the TX_FIFO is considered empty, the MAC transmitter will experience an underflow event, and the corresponding status word will be forwarded to TX_DMA.

Transmitted packets comply with IEEE 802.3/802.z rules. Transmission has the following characteristics:

- Generates preamble and start frame delimiter.
- Generates jam signal in half-duplex mode.
- Supports jabber timeout.
- Controls flow in half-duplex mode.
- Generates transmit frame status.
- Timestamp snapshot control logic supports IEEE 1588 protocol.

When requesting to send a new Ethernet data frame, the MAC controller first sends the preamble and start frame delimiter. The PR is 7 bytes of "0b10101010", and the start frame delimiter is one byte of "0b10101011". In half-duplex mode, the MAC controller uses a deferral mechanism for flow control (backpressure). When the upper-layer application requests to stop receiving frames, if transmit flow control is enabled, the MAC will send a 32-byte jam signal whenever it detects a receiving frame. This causes a collision and makes the remote station back off. The application requests flow control by setting the FCB_BPA bit in the MAC Flow Control Register (0x0018 FLOWCTRL[FCB_BPA]) to 1. If the application requests to send a frame, it will be sent according to schedule even if backpressure is active. If the backpressure function remains active for a long time (more than 16 consecutive collision events occur), the remote station will abort transmission due to excessive collisions.

55.5.4.1 TX_FIFO Flush

Setting the DMA Operation Mode Register (0x1018 DMA_OP_MODE[FTF]) to 1 performs a TX_FIFO flush operation. This operation is executed immediately, even if the TX_FIFO is currently performing a transfer oper-

ation. Performing a TX_FIFO flush operation resets the corresponding pointers to their initial state, causing an underflow event, and the current transfer operation is terminated.

55.5.4.2 Collision Transmission

When the MAC controller transmits a frame in half-duplex mode, a collision event may occur on the MAC line media interface. After a transmission collision event, the MAC indicates a retry attempt before the frame transmission ends. Retransmission is achieved by popping the frame from the FIFO again. After more than 96 bytes (or 548 bytes in 1000 Mbps mode) are popped to the MAC, the FIFO controller releases that space, making it available for DMA to push more data.

55.5.4.3 Transmit Frame Processing: CRC and PAD Generation

When the data frame (Destination MAC Address + Source MAC Address + TYPE + User Data DATA) obtained by the MAC controller is less than 60 bytes, by setting the PAD bit in the transmit descriptor TDES0 to reset, zero padding (PAD) is performed at the end of the frame data to make the data frame length meet the IEEE 802.3 minimum length requirement of 46 bytes. This can also be disabled, performing no padding operation.

The MAC controller also calculates the Cyclic Redundancy Check (CRC) for the Frame Check Sequence (FCS) field. Based on the settings of the DCRC and CRCR bits in the transmit descriptor TDES0, the calculated CRC value can be chosen to be inserted or replaced into the data being transmitted. If the MAC is preset not to append the CRC value to the end of the Ethernet frame, the calculated CRC is not sent. However, when the DPAD bit in TDES0 is 0, frames less than 60 bytes (Destination MAC Address + Source MAC Address + TYPE + User Data DATA) will have padding PAD appended after the data, and the CRC result will also be automatically appended to the end of the padded frame.

CRC calculation includes all bytes of the frame data except the preamble and start frame delimiter. The Ethernet frame CRC32 generating polynomial is 0x04C11DB7, and this polynomial is used for all 32-bit CRC calculations in the Ethernet module, as shown in the following equation:

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1 \quad (11)$$

55.5.4.4 Source MAC Address Processing

If the data from the application does not contain the source address field, the MAC controller can insert this field. If it contains the source address field, it can choose to replace that field with the value set in the MAC Filter Address Register 0 (0x0040 MAC_ADDR_0_HIGH) and other registers. The specific handling method for the SMAC field can be controlled via the [SARC] bit in the MAC Control Register (0x0000 MACCFG) or the SAIC bit in descriptor TDES1.

55.5.4.5 VLAN Insert, Replace, and Delete Processing

The MAC controller can perform insertion, replacement, and deletion processing on the VLAN identifier field of the data frame. This is controlled by the [VLC] bit in the VLAN Tag Insert/Replace Register (0x0584 VLAN_TAG_INC_RPL).

55.5.5 MAC Control Receive Function

The MAC controller caches received data in the RX_FIFO. When the received data meets certain quantity conditions, it instructs RX_DMA to perform data transfer. RX_DMA can then initiate pre-configured burst transfers to the AXI interface.

RX_DMA transfers data from the RX_FIFO to the pre-set system memory in two operational modes: threshold mode and store-and-forward mode.

First, Threshold Mode:

In threshold mode, when the number of bytes received by the RX_FIFO is greater than the configured receive threshold (configured using the [RTC] bit in the DMA Operation Mode Register (0x1018 DMA_OP_MODE[RTC])) or a complete data packet, data transfer is initiated and continues until the entire data packet is transferred. After the FCS field transfer is complete, the receive status is returned to the DMA Status Register. In this mode, some error frames are not discarded because the error status is received at the end of the frame, by which time data has already been transferred from the RX_FIFO to the pre-set system memory.

Second, Store-and-Forward Mode:

In store-and-forward mode, frames can only be read out after they are completely written to the RX_FIFO. In this mode, all error frames are discarded (if configured to do so), so only valid frames are read out and forwarded to the application.

55.5.5.1 Receive Frame Processing

When the MAC detects a start frame delimiter on the interface, it initiates the receive operation. The MAC controller removes the preamble and start frame delimiter before continuing to process the frame. The frame's address field is checked for filtering, and the FCS field is used to verify the frame's CRC. If the frame fails the address filter, it is discarded.

If the TYPE field value of the received frame is less than 0x600 and the MAC is set to automatically remove CRC/PAD, the MAC will send frame data (amount not exceeding that specified in the TYPE field) to the RX_FIFO, then start discarding bytes (including the FCS field). If the TYPE field is greater than or equal to 0x600, the MAC sends all received Ethernet frame data to the RX_FIFO regardless of the CRC removal option setting.

55.5.6 MAC Controller Frame Filtering

The MAC controller supports destination MAC address filtering, source MAC address filtering, and VLAN TAG filtering when receiving frames. All received frames undergo destination MAC address and source MAC address checks.

55.5.6.1 Destination MAC Address Exact Filtering

The MAC controller can perform exact filtering based on the source address field of the received frame. By default, the MAC controller compares the SMAC field (source MAC address) with the value set in the MAC Address Register. This can be controlled by setting the [SA] bit in the MAC Filter Address Register (0x0048 MAC_ADDR_0_HIGH[SA]) to 1. If the SAIF bit in the Address Filter Control Register (0x0004 MACFF[SAIF]) is set to 1, the MAC sub-controller will discard frames that fail SMAC filtering. Otherwise, the result of SA filtering is given via the status bits in the receive status word (see receive descriptor word RDES0).

55.5.6.2 DMAC Address Filtering

- DMAC Address Filtering in Unicast If exact filtering is selected (controlled by setting the [HUC] bit in the Address Filter Control Register (0x0004 MACFF[HU]) to 0), the MAC compares all 48 bits of the unicast destination address with the pre-set MAC filter address to determine a match. If the above [HU] bit is set to 1, the MAC controller will mask the destination MAC address field, thereby achieving filtering.
- DMAC Address Filtering in Multicast If the [HMC] bit in the Address Filter Control Register (0x0004 MACFF[HMC]) is set to 1, the MAC controller will filter based on the pre-set multicast address. If the [HMC] bit is reset, the MAC controller will perform exact filtering on unicast addresses.
- DMAC Address Filtering in Broadcast In default mode, the MAC controller does not filter any broadcast frames. However, setting the [DBF] bit in the Address Filter Control Register (0x0004 MACFF[DBF]) to 1 configures the MAC controller to reject all broadcast frames, causing the MAC controller to discard any broadcast frame.

55.5.6.3 Filtering of TAG Identifier in Ethernet Packets with VLAN Flag

When the received frame is a VLAN frame, the MAC can enable filtering of the VLAN identifier field by setting the [VTFE] bit in the Address Filter Control Register (0x0004 MACFF[VTFE]). The comparison value for filtering is set in the [VL] bit of the VLAN Tag Register (0x001c VLAN_TAG[VL]). Simultaneously, the [ETV] bit in the VLAN Tag Register (0x001c VLAN_TAG[ETV]) can be set to choose whether to compare the entire VLAN identifier field (16 bits) or only the lower 12-bit identifier for filtering.

55.5.7 PTP Sub-Controller Functional Description

The IEEE 1588 standard defines a protocol that supports precise clock synchronization in measurement and control systems using network communication, local computing, and distributed object technologies. This protocol is applicable to systems that communicate using local area networks (including but not limited to Ethernet) that support multicast message delivery. The protocol is used to synchronize heterogeneous systems containing clocks whose inherent accuracy, resolution, and stability vary. The protocol supports system-level synchronization accuracy in the sub-second range and requires minimal network and local clock computational resources. This message-based protocol is called the Precision Time Protocol (PTP). PTP is transmitted via UDP/IP. Systems or networks are classified as master and slave nodes for distributing timing/clock information. The protocol synchronizes the slave node with the master node by exchanging PTP messages, as shown in the figure below.

The PTP process is as follows:

1. The master node broadcasts a PTP Sync message to all its nodes. The Sync message contains the reference time information of the master node. The time the message leaves the master node system is t1.
2. The slave node receives the Sync message and captures the accurate time t2 using its reference timing.
3. The master node subsequently sends a Follow_up message containing t1 information to the slave node for later use.
4. The slave node sends a Delay_Req message to the master node, marking the accurate time t3 when this frame leaves the slave node.
5. The master node receives the message and captures the accurate time t4 when the message enters its

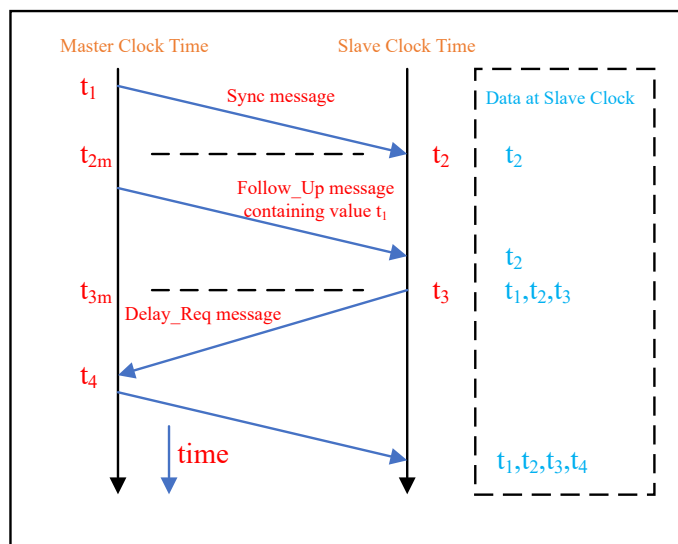


Figure 87: Network Time Synchronization Diagram

system.

6. The master node sends the t_4 information to the slave node using a Delay_Resp message.

7. The slave node uses the four values t_1 , t_2 , t_3 , and t_4 to synchronize its local reference timing with the master node's reference timing.

Although most protocols are implemented in software above the UDP layer, hardware is required to capture the precise time when specific PTP packets enter or leave the Ethernet port.

55.5.7.1 Reference Clock Source

According to the IEEE 1588 specification definition, to obtain a timestamp snapshot, the MAC controller requires a 64-bit format reference time (split into two 32-bit parts: the upper 32 bits represent seconds, and the lower 32 bits represent sub-seconds). The PTP reference clock is used internally to generate reference time (also called system time) and to capture timestamps.

55.5.7.2 PTP Controller Message Types

After the timestamp function is enabled, timestamp snapshots for which message or which type of message can be controlled via the [SNAPTYPSEL], [TSMSTRENA], and [TSEVNTENA] bits in the Timestamp Control Register (0x0700 TS_CTRL[SNAPTYPSEL], 0x0700 TS_CTRL[TSMSTRENA], and 0x0700 TS_CTRL[TSEVNTENA]). The correspondence is shown in the table below.

SNAPTYPSEL	TSMSTRENA	TSEVNTENA	PTP Type
00	x	0	SYNC, Follow_Up, Delay_Req, Delay_Resp
00	0	1	SYNC
00	1	1	Delay_Req
01	x	0	SYNC, Follow_Up, Delay_Req, Delay_Resp, Pdelay_Req, Pdelay_Resp, Pdelay_Resp_Follow_Up

SNAPTYPSEL	TSMSTRENA	TSEVNTENA	PTP Type
01	0	1	SYNC, Pdelay_Req, Pdelay_Resp
01	1	1	Delay_Req, Pdelay_Req, Pdelay_Resp
10	x	x	SYNC, Delay_Req
11	x	x	Pdelay_Req, Pdelay_Resp

55.5.7.3 Receive Characteristics

When the IEEE 1588 timestamp function is enabled, the Ethernet MAC captures timestamps for all frames received on the interface. The MAC controller provides the timestamp when the frame receive process is complete.

The captured timestamp is returned to the application in the same manner as the frame status is provided. The timestamp is sent back within the corresponding receive descriptor along with the frame's receive status. The 64-bit timestamp information is written back to the RDES6 and RDES7 fields, where RDES6 holds the 32 least significant bits of the timestamp, and RDES7 holds the 32 most significant bits of the timestamp.

55.5.7.4 Transmit Characteristics

When the IEEE 1588 timestamp function is enabled, a timestamp is captured when the start frame delimiter is output on the transmit interface. For frames requiring timestamp capture, each transmit frame can be marked to indicate whether it is necessary to capture that frame's timestamp. PTP frames can be identified without processing the transmit frame; frame control can be performed via control bits in the transmit descriptor. The captured timestamp is returned to the application in the same manner as the frame status is provided. The timestamp is sent back within the corresponding transmit descriptor along with the frame's transmit status, thereby automatically associating the timestamp with a specific PTP frame. The 64-bit timestamp information is written back to the TDES6 and TDES7 fields, where TDES6 holds the 32 least significant bits of the timestamp, and TDES7 holds the 32 most significant bits of the timestamp.

55.5.7.5 System Time Calibration

The 64-bit PTP time is updated based on the PTP reference clock. This PTP time can be used as a clock source to take snapshots (timestamps) of Ethernet frames sent or received on the transmission interface. The system time timer can be initialized or calibrated using a coarse calibration method or a fine calibration method.

The system time counter can be initialized or corrected using the coarse correction method. In this method, the initial value or offset value is written to the System Time - Seconds Update Register and the System Time - Nanoseconds Update Register (0x0710 SYST_SEC_UPD, 0x0714 SYST_NSEC_UPD). For initialization, the value in this register (initial value) is written to the system time counter. For system time calibration, the value in this register (offset value) is added to or subtracted from the system time.

In the fine correction method, the reference clock frequency relative to the master clock (as defined in IEEE 1588) is corrected over a period of time, rather than within one clock as in coarse correction. This helps maintain linear time and does not introduce drastic changes (or large jitter) in the reference time between PTP sync message intervals. In this method, an accumulator sums the content of the addend, as shown in the figure below. The arithmetic carry generated by the accumulator is used as a pulse to increment the system time counter. The

accumulator and addend are 32-bit registers. Here, the accumulator functions as a high-precision frequency multiplier or divider.

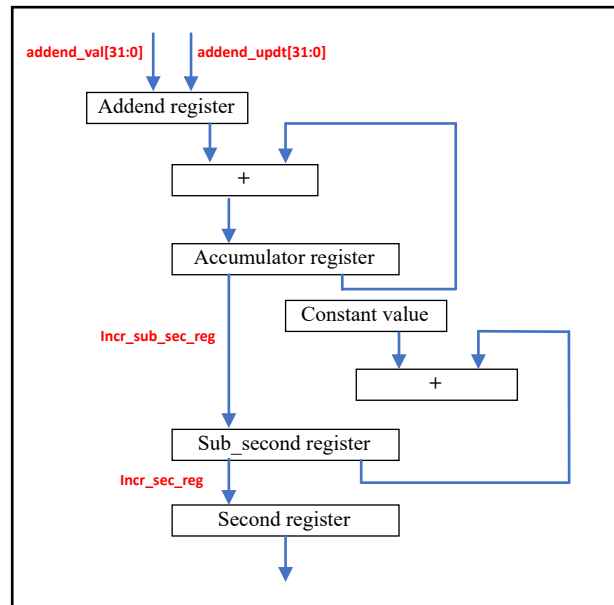


Figure 88: Fine Time Correction System Time Schematic

The system time update logic requires a 100MHz clock frequency to achieve 10ns precision. If the reference clock frequency is 66MHz, the ratio (division factor) of the reference clock frequency to the required clock frequency is 66MHz to 50MHz (value 1.32). The default addend value to be set in the Timestamp Addend Register (0x0718 TS_ADDEND) is the result of 2^{32} divided by 1.32, equivalent to 0xC1F07C1F. If the reference clock drifts down to 65MHz, the div

The value used to increment the Sub-second Increment Register (0x0704 SUB_SEC_INCR) is set to 0x0d43. This allows the system time precision to reach 10ns (in other words, the increment step time is 10 ns).

Software must calculate the frequency offset based on synchronization messages and update the Timestamp Addend Register (0x0718 TS_ADDEND) accordingly. First, use the FreqCompensationValue0 in the Timestamp Addend Register (0x0718 TS_ADDEND) to set the slave clock. The formula for this value is as follows:

$$\text{FreqCompensationValue0} = \frac{2^{32}}{\text{FreqDivisionRatio}} \quad (12)$$

If it is initially assumed that MasterToSlaveDelay is the same for consecutive sync messages, the following algorithm must be applied. After several synchronization cycles, the frequency will lock. The slave clock can then determine the precise value of MasterToSlaveDelay and re-synchronize with the master clock using the new value.

The algorithm is as follows:

1. At MasterSyncTime(n), the master clock sends a Sync message to the slave clock. The slave clock receives the message at its local clock time SlaveClockTime(n) and calculates using the formula:

$$\text{MasterClockTime}(n) = \text{MasterSyncTime}(n) + \text{MasterToSlaveDelay}(n)$$

2. The master clock count MasterClockCount(n) for the current sync cycle is calculated as:

MasterClockCount(n) = MasterClockTime(n) - MasterClockTime(n-1) (assuming MasterToSlaveDelay is the same for sync cycles n and n-1)

3. The slave clock count SlaveClockCount(n) for the current sync cycle is calculated as:

$$\text{SlaveClockCount}(n) = \text{SlaveClockTime}(n) - \text{SlaveClockTime}(n-1)$$

4. The difference between master and slave clock counts SlaveClockCount(n) for the current sync cycle is calculated as:

$$\text{ClockDiffCount}(n) = \text{MasterClockCount}(n) - \text{SlaveClockCount}(n)$$

5. The slave clock's division factor FreqScaleFactor(n) is calculated as:

$$\text{FreqScaleFactor}(n) = (\text{MasterClockCount}(n) + \text{ClockDiffCount}(n)) / \text{SlaveClockCount}(n)$$

6. The frequency compensation value FreqCompensationValue(n) for the addend register is calculated as:

$$\text{FreqCompensationValue}(n) = \text{FreqScaleFactor}(n) \times \text{FreqCompensationValue}(n-1)$$

Theoretically, this algorithm can achieve lock within one sync cycle. However, because network propagation delays and operating conditions constantly change, multiple cycles may be required. The algorithm is self-calibrating. If for some reason the slave clock initially set by the master is incorrect, the algorithm will take more sync cycles to calibrate it.

55.5.8 ENET Interrupt Description

The MAC controller supports interrupts for DMA normal events, DMA abnormal events, MCC events, etc. For details, please refer to the corresponding interrupt registers.

Interrupts summarize the above times into a single interrupt event output to the interrupt module through interrupt enable or mask control. Therefore, when the system responds to an interrupt, it needs to read the corresponding status information to determine the specific interrupt event.

55.5.9 ENET Configuration Steps Description

55.5.9.1 MII Interface Transmission

1. Configure the source clock frequency as needed.
2. Configure the ENET0_PHY_INF_SEL bit in Control Register 2 (i.e., 0x3008 CTRL2[ENET0_PHY_INF_SEL]) to select the MII interface type.
3. Select options as needed. This step must not be performed before step 2.
4. Set the [SWR] bit in the DMA Bus Mode Register (0x1000 DMA_BUS_MODE[SWR]) to 1 to perform a bus software reset, and wait for the software reset to complete.
5. Select the burst length in the DMA Bus Mode Register (0x1000 DMA_BUS_MODE[SWR]) as needed.
6. Configure the AXI bus burst length in the AXI Bus Mode Register (0x1028 DMA_AXI_MODE) as needed.
7. Write the Ethernet packet to the pre-determined system memory as needed, then write the data base address to the DMA Transmit Descriptor List Address Register (0x1010 DMA_TX_DESC_LIST_ADDR).
8. Configure the AXI bus transmit transfer threshold in the [TTC] bit of the DMA Operation Mode Register (0x1018 DMA_OP_MODE) as needed.
9. Configure the interrupt type in the DMA Interrupt Enable Register (0x101c DMA_INTR_EN) as needed.
10. In the MAC Configuration Register (0x0000 MACCFG), select source address insertion or replacement, interface speed, and port operation mode as needed.

11. Set the [GB] bit in the GMII Address Register (0x0010 GMII_ADDR) to 1.
12. Write the MAC address to be sent to the MAC Address 0 High Register (0x0040 MAC_ADDR_0_HIGH) and the MAC Address 0 Low Register (0x0044 MAC_ADDR_0_LOW) as needed.
13. Configure the GMII Data Register (0x0014 GMII_DATA) and the GMII Address Register (0x0010 GMII_ADDR) as needed.
14. Wait for the [GB] bit in the GMII Address Register (0x0010 GMII_ADDR) to become 0.
15. Configure the [ST] bit in the DMA Operation Mode Register (0x1018 DMA_OP_MODE) to 1 to enable frame transmission start, and configure the [TE] bit in the MAC Configuration Register (0x0000 MACCFG) to 1. Wait for transmission to complete and enter the pre-set DMA interrupt to check the transmit status.

55.5.9.2 MII Interface Reception

Please refer to the configuration steps in the previous subsection for MII interface configuration. Adjust the transmit configuration to receive configuration as needed.

55.5.10 Descriptor Details

55.5.10.1 Enhanced Transmit Descriptor

	31								0
TDES0	OWN	Ctrl [30:26]	T T S E	Ctrl [24:18]	T T S S	Status [16:7]	Ctrl/Status [6:3]	Status [2:0]	
TDES1	Ctrl [31:29]		Buffer 2 Byte Count [28:16]			R E S	Buffer 1 Byte Count [12:0]		
TDES2	Buffer 1 Address [31:0]								
TDES3	Buffer 2 Address [31:0] or Next Descriptor Address [31:0]								
TDES4	Reserved								
TDES5	Reserved								
TDES6	Transmit Timestamp Low [31:0]								
TDES7	Transmit Timestamp High [31:0]								

Figure 89: Transmit Descriptor

BIT	Field	Description
31	OWN	0: This descriptor is owned by the CPU. 1: This descriptor is owned by the DMA. DMA clears this bit to 0 upon completing frame transmission or after all buffers allocated by the descriptor have been read. The OWN bit of the first descriptor of a frame should also be set to 1 after all subsequent descriptors belonging to the same frame are set.
30	IC	Interrupt on Completion When this bit is set to 1, it indicates that after the current frame transmission is complete, the TI bit in the DMA Status Register (0x1014 DMA_STATUS[TI]) will be set to 1.
29	LS	Last Segment This bit is 1 when this buffer is the last buffer in the frame.
28	FS	First Segment This bit is 1 when the buffer contains the first segment of the frame.
27	DC	Disable CRC When this bit is 1, the MAC will not append CRC to the end of the transmitted frame.
26	DP	Disable PAD When this bit is 1, the MAC will not automatically add padding to frames less than 64 bytes. When 0, the MAC automatically adds padding and CRC to frames less than 64 bytes.
25	TTSE	Transmit Timestamp Enable When this bit is 1, IEEE 1588 hardware timestamping is activated for the transmit frame described by this descriptor.
24	CRCR	CRC Replacement When this bit is 1, the MAC replaces the FCS field of the frame to be transmitted with the calculated CRC value.
23:22	CIC	Checksum Insertion Control
21	TER	Transmit End of Ring This bit is 1 indicates the descriptor list has reached its final descriptor. The DMA returns to the base address of the list, forming a descriptor ring.
20	TCH	Transmit Chain This bit is 1 indicates the second address in the descriptor is the address of the next descriptor.

BIT	Field	Description
19:18	VLIC	VLAN Insertion Control • 00: No processing for VLAN frames. • 01: Delete the VLAN tag and type field before transmission. • 10: Insert the VLAN identifier from the VLAN Tag Insert/Replace Register (0x00584 VLAN_TAG_INC_RPL) into the transmit frame. • 11: Replace the original VLAN identifier in the transmit frame with the VLAN tag value from the VLAN Tag Transmit Control Register (ENET_MACVLTTTR).
17	TTSS	Transmit Timestamp Status This bit indicates a timestamp has been captured for the current transmit frame. When set to 1, TDES6 and TDES7 hold the captured timestamp value for the transmit frame.
16	IHE	IP Header Error This bit is 1 indicates the MAC transmitter detected an error in the IP data header.
15	ES	Error Summary This bit is set to 1 if any one of the bits B[16], B[14], B[13], B[12], B[11], B[10], B[9], B[8], B[2], B[1] in this descriptor is set to 1.
14	JT	Jabber Timeout Error This bit is 1 indicates the MAC transmitter experienced a Jabber timeout.
13	FF	Frame Flushed This bit is 1 indicates DMA has flushed the frame as per the software flush command issued by the CPU.
12	IPE	IP Payload Error This bit is 1 indicates the Checksum Offload Engine (COE) detected an error in the TCP, UDP, or ICMP payload and did not update the Checksum field in the original frame.
11	LOC	Loss of Carrier This bit is 1 indicates loss of carrier during frame transmission, i.e., the ENET_MII_CRS signal was invalid for one or more transmit clock cycles during frame transmission.
10	NC	No Carrier This bit is 1 indicates carrier sense was not triggered by the PHY during transmission.
9	LC	Late Collision Error This bit is 1 indicates frame transmission was aborted due to a collision occurring after the collision window (64 byte times including preamble in MII mode).

BIT	Field	Description
8	EC	Excessive Collisions Error This bit is 1 indicates an attempt to transmit the current frame was aborted due to 16 consecutive collisions.
7	VF	VLAN Frame This bit is 1 indicates the currently transmitted frame is a VLAN frame.
6: 3	CC	Collision Count This value indicates the number of collisions that occurred before transmitting the frame.
2	ED	Excessive Deferral Error This bit is 1 indicates transmission was aborted due to excessive deferral exceeding 24288 bit times.
1	UF	Underflow Error This bit is 1 indicates the MAC aborted frame transmission because data did not arrive at the transmit buffer in time, causing an underflow error. This indicates DMA encountered an empty transmit buffer during frame transmission.
0	DB	Deferred Bit This bit is 1 indicates the MAC deferred transmission because carrier was present.

Table 215: TDES0 Detailed Field Description

BIT	Field	Description
31:29	SAIC	SA Insertion/Replacement Control • 001: Insert the address from MAC Address Register 0 as the SA address into the transmit frame. • 101: Insert the address from MAC Address Register 1 as the SA address into the transmit frame. • 010: Replace the SA field in the transmit frame with the address from MAC Address Register 0. • 110: Replace the SA field in the transmit frame with the address from MAC Address Register 1. • Others: Reserved
28:16	TBS2	Transmit Buffer 2 Size Indicates the size of the second data buffer in bytes.
15:13	Reserved	Reserved
12:0	TBS1	Transmit Buffer 1 Size Indicates the size of the first data buffer in bytes. If this field is 0, DMA ignores this buffer and uses Buffer 2 or the next descriptor.

Table 216: Detailed Field Description for TDES1

BIT	Field	Description
31:0	-	Transmit Buffer Address 1 Indicates to the DMA the physical address of Buffer 1 in memory. This field is written by software when it provides this descriptor to the DMA (TDES0[OWN]=1).

Table 217: Detailed Field Description for TDES2

BIT	Field
31:0	-

Table 218: Detailed Field Description for TDES3

55.5.10.2 Enhanced Receive Descriptor

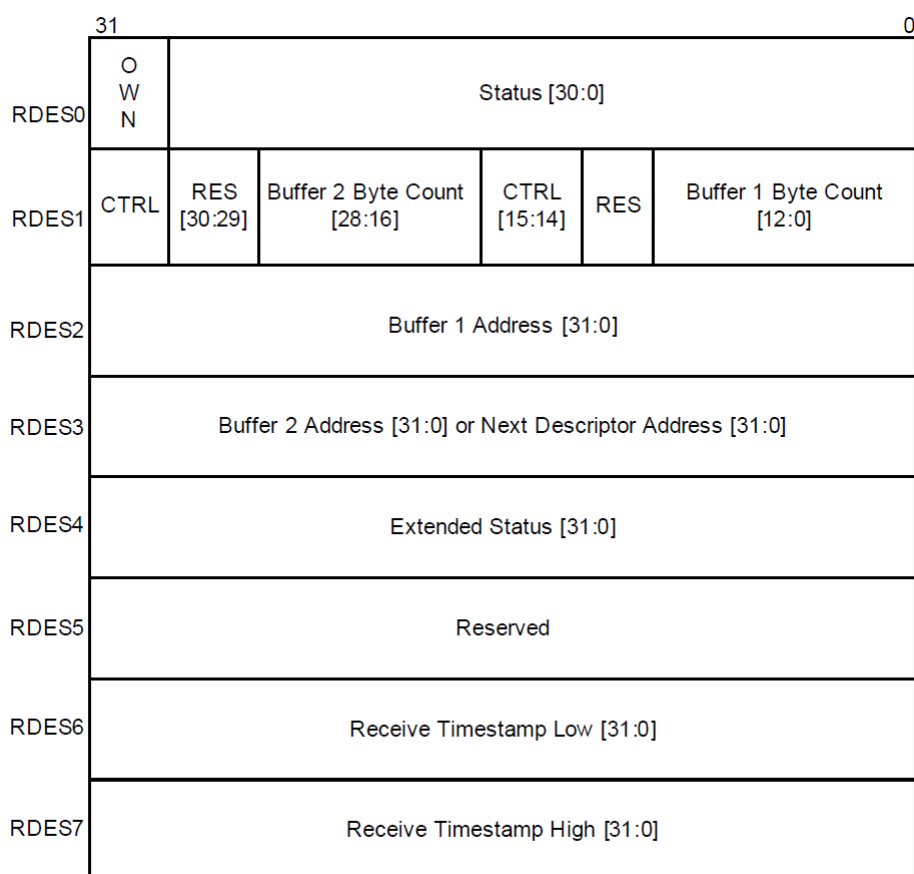


Figure 90: Receive Descriptor

BIT	Field	Description
31	OWN	Descriptor Ownership 0: This descriptor is owned by the CPU. 1: This descriptor is owned by the DMA. DMA clears this bit to 0 upon completing frame reception or when the associated buffer is full.
30	AFM	Destination Address Filter Fail Set to 1 when the frame failed the Destination Address (DA) filtering in the MAC core.
29:16	FL	Frame Length Indicates the length in bytes of the received frame transferred to host memory (including the CRC).
15	ES	Error Summary Set to 1 if any one of the following bits in this descriptor is set: B[14], B[11], B[7], B[6], B[4], B[3], B[1], or bits B[4] and B[3] in the RDES4 descriptor.
14	DE	Descriptor Error Set to 1 when a frame is truncated because it exceeds the size of the current descriptor's buffer AND the DMA does not own the next descriptor.
13	SAF	Source Address Filter Fail Set to 1 when the frame's SMAC field failed the Source Address (SA) filtering in the MAC controller.
12	LE	Length Field Error Set to 1 when the actual length of the received frame does not match the value in the Length/Type field.
11	OE	Overflow Error Set to 1 when a frame is corrupted due to a buffer overflow.
10	VLAN	VLAN Tagged Frame Set to 1 when the descriptor points to a frame that was tagged as a VLAN frame by the MAC core.
9	FS	First Descriptor of Frame Set to 1 when this descriptor points to the first buffer of the frame.
8	LS	Last Descriptor of Frame Set to 1 when this descriptor points to the last buffer of the frame.
7	ICE	IP Checksum Engine Error Set to 1 when an error is found in the IPv4 or IPv6 header. Possible reasons: 1) Ethernet Type field mismatches the IP header version field. 2) IPv4 header checksum mismatch. 3) Ethernet frame lacks the required number of IP header bytes.

BIT	Field	Description
6	LC	Late Collision Error Set to 1 when a late collision occurred while receiving the frame in half-duplex mode.
5	FT	Frame Type 0: MAC received an Ethernet frame. 1: MAC received a PTP frame.
4	RWT	Receive Watchdog Timeout Set to 1 when the receive watchdog timer expired during reception of the current frame, and the frame was truncated after the timeout.
3	RE	Receive Error Set to 1 when the PHY asserted the RX_ER signal while asserting the RX_DV signal during frame reception.
2	DE	Dribble Bit Error Set to 1 when the received frame has a non-integer multiple of bytes (odd nibble). Note: This bit is valid only in MII mode.
1	CE	CRC Error Set to 1 when a Cyclic Redundancy Check (CRC) error occurred in the received frame.
0	ESAMC	Extended Specific Address Match Control 0: The received frame passed DA filtering against MAC Address Register 0. 1: The received frame passed DA filtering against other MAC Address Registers.

Table 219: Detailed Field Description for RDES0

BIT	Field	Description
31	DIC	Disable Interrupt on Completion Set to 1 to indicate that the received frame ends in the buffer pointed to by this descriptor, and to disable generating an interrupt to the host. It also prevents the receive status bit in the DMA Status Register from being set.
30:29	Reserved	Reserved
28:16	RB2	Receive Buffer 2 Size Indicates the size of the second data buffer in bytes.
15	RER	Receive End of Ring Set to 1 to indicate this is the last descriptor in the list. The DMA will return to the base address of the descriptor list, forming a ring.

BIT	Field	Description
14	RCH	Receive Chain Set to 1 to indicate that the second address in this descriptor points to the next descriptor, not to a second buffer. In this case, the RB2 field is invalid.
13	Reserved	Reserved
12:0	RB1	Receive Buffer 1 Size Indicates the size of the first data buffer in bytes.

Table 220: Detailed Field Description for RDES1

BIT	Field	Description
31:0	-	Receive Buffer Address 1 Indicates to the DMA the physical address of Buffer 1 in memory. This field is written by software when it provides this descriptor to the DMA (RDES0[OWN]=1).

Table 221: Detailed Field Description for RDES2

BIT	Field	Description
31:0	-	Receive Buffer Address 2 Indicates to the DMA the physical address of Buffer 2 in memory. This is used when a descriptor ring structure is employed. Software writes this when providing the descriptor (RDES0[OWN]=1).

Table 222: Detailed Field Description for RDES3

BIT	Field	Description
31:28		Reserved
27:26		Layer 3 and Layer 4 Filter Number Matched 00: Filter 0 01: Filter 1 10: Filter 2 11: Filter 3 Note: This field is valid only when bit 24 or bit 25 is set high. When multiple filters match, these bits indicate the smallest matching filter number.
25		Layer 4 Filter Match Set to 1 when the received frame matches one of the enabled Layer 4 (port number) fields.

BIT	Field	Description
24		Layer 3 Filter Match Set to 1 when the received frame matches one of the enabled Layer 3 (IP address) fields.
23:21		Reserved
20:18		VLAN Tag Priority Value (User Priority) Indicates the user priority value in the VLAN tag of the received packet. Valid only when bits 16 and 17 of RDES4 are set.
17		Received Packet with AV Tag Set to 1 when a packet with an Audio/Video (AV) tag is received.
16		Received AV Packet Set to 1 when an Audio/Video (AV) packet is received.
15		Reserved
14		Timestamp Dropped Set to 1 when a timestamp snapshot was taken but then discarded due to an RX_FIFO overflow.
13		PTP Frame Type (Version) Indicates the version type of the received PTP frame. 0: IEEE1588v1 PTP frame. 1: IEEE1588v2 PTP frame.
12		PTP Frame Type (Encapsulation) Indicates the encapsulation type of the received PTP frame. 0: PTP frame over UDP-IPv4 or UDP-IPv6. 1: PTP frame over Ethernet.
11:8		PTP Message Type Indicates the type of the received PTP message. 0000: No PTP message received. 0001: SYNC (all clock types). 0010: Follow_Up (all clock types). 0011: Delay_Req (all clock types). 0100: Delay_Resp (all clock types). 0101: Pdelay_Req (in peer-to-peer transparent clock). 0110: Pdelay_Resp (in peer-to-peer transparent clock). 0111: Pdelay_Resp_Follow_Up (in peer-to-peer transparent clock). 1000: Announce. 1001: Management. 1010: Signaling. 1111: Default PTP message type. - Others: Reserved.
7		IPv6 Packet Set to 1 when an IPv6 packet is received.

BIT	Field	Description
6		IPv4 Packet Set to 1 when an IPv4 packet is received.
5		Bypass COE Engine Set to 1 to bypass the Checksum Offload Engine (COE).
4		Payload Error Indicates an IP payload error. Set to 1 in the following cases: 1) The 16-bit IP payload checksum (i.e., TCP, UDP, or ICMP checksum) calculated by the kernel does not match the corresponding checksum field in the received frame. 2) The TCP, UDP, or ICMP segment length does not match the payload length value in the IP header.
3		IP Header Error Set to 1 when the 16-bit IPv4 header checksum calculated by the kernel does not match the received checksum bytes, or when the IP header version field is inconsistent with the Ethernet Type value.
2:0		IP Payload Type Indicates the type of payload encapsulated in the IP datagram. These bits are "000" if there is an IP header error or a fragmented IP packet. 000: Unknown, or IP payload not processed. 001: UDP. 010: TCP. 011: ICMP. 1xx: Reserved.

Table 223: Detailed Field Description for RDES4

BIT	Field	Description
31:0	RTSL	Receive Timestamp Low DMA updates this field with the lower 32 bits of the timestamp captured for the corresponding received frame.

Table 224: Detailed Field Description for RDES6

BIT	Field	Description
31:0	RTSH	Receive Timestamp High DMA updates this field with the upper 32 bits of the timestamp captured for the corresponding received frame.

Table 225: Detailed Field Description for RDES7

55.6 ENET Register List

ENET0 base address: 0xF1400000



Offset	Register Name	Description	Reset Value
0x0000	MACCFG	MAC Configuration Register	0x00000000
0x0004	MACFF	MAC Frame Filter	0x00000000
0x0008	HASH_H	Hash Table High Register	0x00000000
0x000C	HASH_L	Hash Table Low Register	0x00000000
0x0010	GMII_ADDR	GMII Address Register	0x00000000
0x0014	GMII_DATA	GMII Data Register	0x00000000
0x0018	FLOWCTRL	Flow Control Register	0x00000000
0x001C	VLAN_TAG	VLAN Tag Register	0x00000000
0x0028	RWKFRMFILT	Remote Wake-Up Frame Filter Register	0x00000000
0x002C	PMT_CSR	PMT Control and Status Register	0x00000000
0x0030	LPI_CSR	LPI Control and Status Register	0x00000000
0x0034	LPI_TCR	LPI Timers Control Register	0x00000000
0x0038	INTR_STATUS	Interrupt Status Register	0x00000000
0x003C	INTR_MASK	Interrupt Mask Register	0x00000000
0x0040	MAC_ADDR_0_HIGH	MAC Address 0 High Register	0x00000000
0x0044	MAC_ADDR_0_LOW	MAC Address 0 Low Register	0x00000000
0x0048	MAC_ADDR[1][HIGH]	MAC Address High Register	0x00000000
0x004C	MAC_ADDR[1][LOW]	MAC Address Low Register	0x00000000
0x0050	MAC_ADDR[2][HIGH]	MAC Address2 High Register	0x00000000
0x0054	MAC_ADDR[2][LOW]	MAC Address2 Low Register	0x00000000
0x0058	MAC_ADDR[3][HIGH]	MAC Address3 High Register	0x00000000
0x005C	MAC_ADDR[3][LOW]	MAC Address3 Low Register	0x00000000
0x0060	MAC_ADDR[4][HIGH]	MAC Address4 High Register	0x00000000
0x0064	MAC_ADDR[4][LOW]	MAC Address4 Low Register	0x00000000
0x00D8	XMII_CSR	SGMII/RGMII/SMII Control and Status Register	0x00000000
0x00DC	WDOG_WTO	Watchdog Timeout Register	0x00000000
0x0100	MMC_CNTRL	MMC Control establishes the operating mode of MMC.	0x00000000
0x0104	MMC_INTR_RX	MMC Receive Interrupt	0x00000000
0x0108	MMC_INTR_TX	MMC Transmit Interrupt	0x00000000
0x010C	MMC_INTR_MASK_RX	MMC Receive Interrupt mask	0x00000000
0x0110	MMC_INTR_MASK_TX	MMC Transmit Interrupt Mask	0x00000000
0x0124	TX64OCTETS_GB	Number of good and bad frames transmitted with length 64 bytes, exclusive of preamble and retried frames.	0x00000000

Offset	Register Name	Description	Reset Value
0x0128	TX65TO127OCTETS_GB	Number of good and bad frames transmitted with length between 65 and 127 (inclusive) bytes, exclusive of preamble and retried frames.	0x00000000
0x012C	TX128TO255OCTETS_GB	Number of good and bad frames transmitted with length between 128 and 255 (inclusive) bytes, exclusive of preamble and retried frames.	0x00000000
0x0130	TX256TO511OCTETS_GB	Number of good and bad frames transmitted with length between 256 and 511 (inclusive) bytes, exclusive of preamble and retried frames.	0x00000000
0x0134	TX512TO1023OCTETS_GB	Number of good and bad frames transmitted with length between 512 and 1,023 (inclusive) bytes, exclusive of preamble and retried frames.	0x00000000
0x0138	TX1024TOMAXOCTETS_GB	Number of good and bad frames transmitted with length between 1,024 and maxsize (inclusive) bytes, exclusive of preamble and retried frames.	0x00000000
0x0180	RXFRAMECOUNT_GB	Number of good and bad frames received	0x00000000
0x0200	MMC_IPC_INTR_MASK_RX	MMC IPC Receive Checksum Offload Interrupt Mask maintains the mask for the interrupt generated from the receive IPC statistic counters.	0x00000000

Offset	Register Name	Description	Reset Value
0x0208	MMC_IPC_INTR_RX	MMC Receive Checksum Offload Interrupt maintains the interrupt that the receive IPC statistic counters generate. See Table 4-25 for further detail.	0x00000000
0x0210	RXIPv4_GD_FMS	Number of good IPv4 datagrams received with the TCP, UDP, or ICMP payload	0x00000000
0x0400	L3_L4_CFG[0][L3_L4_CTRL]	Layer 3 and Layer 4 Control Register	0x00000000
0x0404	L3_L4_CFG[0][L4_ADDR]	Layer 4 Address Register	0x00000000
0x0410	L3_L4_CFG[0][L3_ADDR_0]	Layer 3 Address 0 Register	0x00000000
0x0414	L3_L4_CFG[0][L3_ADDR_1]	Layer 3 Address 1 Register	0x00000000
0x0418	L3_L4_CFG[0][L3_ADDR_2]	Layer 3 Address 2 Register	0x00000000
0x041C	L3_L4_CFG[0][L3_ADDR_3]	Layer 3 Address 3 Register	0x00000000
0x0584	VLAN_TAG_INC_RPL	VLAN Tag Inclusion or Replacement Register	0x00000000
0x0588	VLAN_HASH	VLAN Hash Table Register	0x00000000
0x0700	TS_CTRL	Timestamp Control Register	0x00000000
0x0704	SUB_SEC_INCR	Sub-Second Increment Register	0x00000000
0x0708	SYST_SEC	System Time - Seconds Register	0x00000000
0x070C	SYST_NSEC	System Time - Nanoseconds Register	0x00000000
0x0710	SYST_SEC_UPD	System Time - Seconds Update Register	0x00000000
0x0714	SYST_NSEC_UPD	System Time - Nanoseconds Update Register	0x00000000
0x0718	TS_ADDEND	Timestamp Addend Register	0x00000000
0x071C	TGTTM_SEC	Target Time Seconds Register	0x00000000
0x0720	TGTTM_NSEC	Target Time Nanoseconds Register	0x00000000
0x0724	SYSTEM_H_SEC	System Time - Higher Word Seconds Register	0x00000000
0x0728	TS_STATUS	Timestamp Status Register	0x00000000
0x072C	PPS_CTRL	PPS Control Register	0x00000000
0x0730	AUX_TS_NSEC	Auxiliary Timestamp - Nanoseconds Register	0x00000000

Offset	Register Name	Description	Reset Value
0x0734	AUX_TS_SEC	Auxiliary Timestamp - Seconds Register	0x00000000
0x0760	PPS0_INTERVAL	PPS Interval Register	0x00000000
0x0764	PPS0_WIDTH	PPS Width Register	0x00000000
0x0780	PPS[1][TGTTM_SEC]	PPS Target Time Seconds Register	0x00000000
0x0784	PPS[1][TGTTM_NSEC]	PPS Target Time Nanoseconds Register	0x00000000
0x0788	PPS[1][INTERVAL]	PPS Interval Register	0x00000000
0x078C	PPS[1][WIDTH]	PPS Width Register	0x00000000
0x07A0	PPS[2][TGTTM_SEC]	PPS Target Time Seconds Register	0x00000000
0x07A4	PPS[2][TGTTM_NSEC]	PPS Target Time Nanoseconds Register	0x00000000
0x07A8	PPS[2][INTERVAL]	PPS Interval Register	0x00000000
0x07AC	PPS[2][WIDTH]	PPS Width Register	0x00000000
0x07C0	PPS[3][TGTTM_SEC]	PPS Target Time Seconds Register	0x00000000
0x07C4	PPS[3][TGTTM_NSEC]	PPS Target Time Nanoseconds Register	0x00000000
0x07C8	PPS[3][INTERVAL]	PPS Interval Register	0x00000000
0x07CC	PPS[3][WIDTH]	PPS Width Register	0x00000000
0x1000	DMA_BUS_MODE	Bus Mode Register	0x00000000
0x1004	DMA_TX_POLL_DEMAND	Transmit Poll Demand Register	0x00000000
0x1008	DMA_RX_POLL_DEMAND	Receive Poll Demand Register	0x00000000
0x100C	DMA_RX_DESC_LIST_ADDR	Receive Descriptor List Address Register	0x00000000
0x1010	DMA_TX_DESC_LIST_ADDR	Transmit Descriptor List Address Register	0x00000000
0x1014	DMA_STATUS	Status Register	0x00000000
0x1018	DMA_OP_MODE	Operation Mode Register	0x00000000
0x101C	DMA_INTR_EN	Interrupt Enable Register	0x00000000
0x1020	DMA_MISS_OVF_CNT	Missed Frame And Buffer Overflow Counter Register	0x00000000
0x1024	DMA_RX_INTR_WDOG	Receive Interrupt Watchdog Timer Register	0x00000000
0x1028	DMA_AXI_MODE	AXI Bus Mode Register	0x00000000
0x102C	DMA_BUS_STATUS	AHB or AXI Status Register	0x00000000
0x1048	DMA_CURR_HOST_TX_DESC	Current Host Transmit Descriptor Register	0x00000000

Offset	Register Name	Description	Reset Value
0x104C	DMA_CURR_HOST_RX_DESC	Current Host Receive Descriptor Register	0x00000000
0x1050	DMA_CURR_HOST_TX_BUF	Current Host Transmit Buffer Address Register	0x00000000
0x1054	DMA_CURR_HOST_RX_BUF	Current Host Receive Buffer Address Register	0x00000000
0x3000	CTRL0	Control Register 0	0x00000000
0x3008	CTRL2	Control Register 1	0x00000000

Table 226: ENET Register List

55.7 ENET Register Description

55.7.1 MACCFG (0x0)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD		SARC		TWOKPE	SFTERR	CST	TC	WD	JD	BE	JE		IFG		DCRS	PS	FES	DO	LM	DM	IPC	DR	LUD	ACS		BL	DC	TE	RE		PRELEN
N/A		RW		RW	RW	RW	RW	RW	RW	RW	RW		RW		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW		RW	RW	RW	RW		RW
x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MACCFG [31:0]

Bits	Name	Description
30-28	SARC	Source Address Insertion or Replacement Control This field controls the source address insertion or replacement for all transmitted frames. Bit 30 specifies which MAC Address register (0 or 1) is used for source address insertion or replacement based on the values of Bits [29:28]: - 2'b0x: The input signals mti_sa_ctrl_i and ati_sa_ctrl_i control the SA field generation. - 2'b10: - If Bit 30 is set to 0, the MAC inserts the content of the MAC Address 0 registers (registers 16 and 17) in the SA field of all transmitted frames. - If Bit 30 is set to 1 and the Enable MAC Address Register 1 option is selected during core configuration, the MAC inserts the content of the MAC Address 1 registers (registers 18 and 19) in the SA field of all transmitted frames. - 2'b11: - If Bit 30 is set to 0, the MAC replaces the content of the MAC Address 0 registers (registers 16 and 17) in the SA field of all transmitted frames. - If Bit 30 is set to 1 and the Enable MAC Address Register 1 option is selected during core configuration, the MAC replaces the content of the MAC Address 1 registers (registers 18 and 19) in the SA field of all transmitted frames. Note: - Changes to this field take effect only on the start of a frame. If you write this register field when a frame is being transmitted, only the subsequent frame can use the updated value, that is, the current frame does not use the updated value. - These bits are reserved and RO when the Enable SA, VLAN, and CRC Insertion on TX feature is not selected during core configuration.
27	TWOKPE	IEEE 802.3as Support for 2K Packets When set, the MAC considers all frames, with up to 2,000 bytes length, as normal packets. When Bit 20 (JE) is not set, the MAC considers all received frames of size more than 2K bytes as Giant frames. When this bit is reset and Bit 20 (JE) is not set, the MAC considers all received frames of size more than 1,518 bytes (1,522 bytes for tagged) as Giant frames. When Bit 20 is set, setting this bit has no effect on Giant Frame status.

Bits	Name	Description
26	SFTERR	SMII Force Transmit Error When set, this bit indicates to the PHY to force a transmit error in the SMII frame being transmitted. This bit is reserved if the SMII PHY port is not selected during core configuration.
25	CST	CRC Stripping for Type Frames When this bit is set, the last 4 bytes (FCS) of all frames of Ether type (Length/Type field greater than or equal to 1,536) are stripped and dropped before forwarding the frame to the application. This function is not valid when the IP Checksum Engine (Type 1) is enabled in the MAC receiver. This function is valid when Type 2 Checksum Offload Engine is enabled.
24	TC	Transmit Configuration in RGMII, SGMII, or SMII When set, this bit enables the transmission of duplex mode, link speed, and link up or down information to the PHY in the RGMII, SMII, or SGMII port. When this bit is reset, no such information is driven to the PHY. This bit is reserved (and RO) if the RGMII, SMII, or SGMII PHY port is not selected during core configuration.
23	WD	Watchdog Disable When this bit is set, the MAC disables the watchdog timer on the receiver. The MAC can receive frames of up to 16,383 bytes.
22	JD	Jabber Disable When this bit is set, the MAC disables the jabber timer on the transmitter. The MAC can transfer frames of up to 16,383 bytes. When this bit is reset, the MAC cuts off the transmitter if the application sends out more than 2,048 bytes of data (10,240 if JE is set high) during transmission.
21	BE	Frame Burst Enable When this bit is set, the MAC allows frame bursting during transmission in the GMII half-duplex mode. This bit is reserved (and RO) in the 10/100 Mbps only or full-duplex-only configurations.
20	JE	Jumbo Frame Enable When this bit is set, the MAC allows Jumbo frames of 9,018 bytes (9,022 bytes for VLAN tagged frames) without reporting a giant frame error in the receive frame status.

Bits	Name	Description
19-17	IFG	Inter-Frame Gap These bits control the minimum IFG between frames during transmission. - 000: 96 bit times - 001: 88 bit times - 010: 80 bit times - ... - 111: 40 bit times In the half-duplex mode, the minimum IFG can be configured only for 64 bit times (IFG = 100). Lower values are not considered. In the 1000-Mbps mode, the minimum IFG supported is 64 bit times (and above) in the GMAC-CORE configuration and 80 bit times (and above) in other configurations. When a JAM pattern is being transmitted because of backpressure activation, the MAC does not consider the minimum IFG.
16	DCRS	Disable Carrier Sense During Transmission When set high, this bit makes the MAC transmitter ignore the (G)MII CRS signal during frame transmission in the half-duplex mode. This request results in no errors generated because of Loss of Carrier or No Carrier during such transmission. When this bit is low, the MAC transmitter generates such errors because of Carrier Sense and can even abort the transmissions.
15	PS	Port Select This bit selects the Ethernet line speed. - 0: For 1000 Mbps operations - 1: For 10 or 100 Mbps operations In 10 or 100 Mbps operations, this bit, along with FES bit, selects the exact line speed. In the 10/100 Mbps-only (always 1) or 1000 Mbps-only (always 0) configurations, this bit is read-only with the appropriate value. In default 10/100/1000 Mbps configuration, this bit is R_W. The mac_portselect_o or mac_speed_o[1] signal reflects the value of this bit.

Bits	Name	Description
14	FES	Speed This bit selects the speed in the MII, RMII, SMII, RGMII, SGMII, or RevMII interface: - 0: 10 Mbps - 1: 100 Mbps This bit is reserved (RO) by default and is enabled only when the parameter SPEED_SELECT = Enabled. This bit generates link speed encoding when Bit 24 (TC) is set in the RGMII, SMII, or SGMII mode. This bit is always enabled for RGMII, SGMII, SMII, or RevMII interface. In configurations with RGMII, SGMII, SMII, or RevMII interface, this bit is driven as an output signal (mac_speed_o[0]) to reflect the value of this bit in the mac_speed_o signal. In configurations with RMII, MII, or GMII interface, you can optionally drive this bit as an output signal (mac_speed_o[0]) to reflect its value in the mac_speed_o signal.
13	DO	Disable Receive Own When this bit is set, the MAC disables the reception of frames when the phy_txen_o is asserted in the half-duplex mode. When this bit is reset, the MAC receives all packets that are given by the PHY while transmitting. This bit is not applicable if the MAC is operating in the full-duplex mode. This bit is reserved (RO with default value) if the MAC is configured for the full-duplex-only operation.
12	LM	Loopback Mode When this bit is set, the MAC operates in the loopback mode at GMII or MII. The (G)MII Receive clock input (clk_rx_i) is required for the loopback to work properly, because the Transmit clock is not looped-back internally.
11	DM	Duplex Mode When this bit is set, the MAC operates in the full-duplex mode where it can transmit and receive simultaneously.

Bits	Name	Description
10	IPC	Checksum Offload When this bit is set, the MAC calculates the 16-bit one's complement of the one's complement sum of all received Ethernet frame payloads. It also checks whether the IPv4 Header checksum (assumed to be bytes 25–26 or 29–30 (VLAN-tagged) of the received Ethernet frame) is correct for the received frame and gives the status in the receive status word. The MAC also appends the 16-bit checksum calculated for the IP header datagram payload (bytes after the IPv4 header) and appends it to the Ethernet frame transferred to the application (when Type 2 COE is deselected). When this bit is reset, this function is disabled. When Type 2 COE is selected, this bit, when set, enables the IPv4 header checksum checking and IPv4 or IPv6 TCP, UDP, or ICMP payload checksum checking.
9	DR	Disable Retry When this bit is set, the MAC attempts only one transmission. When a collision occurs on the GMII or MII interface, the MAC ignores the current frame transmission and reports a Frame Abort with excessive collision error in the transmit frame status. When this bit is reset, the MAC attempts retries based on the settings of the BL field (Bits [6:5]).
8	LUD	Link Up or Down This bit indicates whether the link is up or down during the transmission of configuration in the RGMII, SGMII, or SMII interface: - 0: Link Down - 1: Link Up
7	ACS	Automatic Pad or CRC Stripping When this bit is set, the MAC strips the Pad or FCS field on the incoming frames only if the value of the length field is less than 1,536 bytes. All received frames with length field greater than or equal to 1,536 bytes are passed to the application without stripping the Pad or FCS field. When this bit is reset, the MAC passes all incoming frames, without modifying them, to the Host.

Bits	Name	Description
6-5	BL	Back-Off Limit The Back-Off limit determines the random integer number (r) of slot time delays (4,096 bit times for 1000 Mbps and 512 bit times for 10/100 Mbps) for which the MAC waits before rescheduling a transmission attempt during retries after a collision. This bit is applicable only in the half-duplex mode and is reserved (RO) in the full-duplex-only configuration. - 00: $k = \min(n, 10)$ - 01: $k = \min(n, 8)$ - 10: $k = \min(n, 4)$ - 11: $k = \min(n, 1)$ where $n = \text{retransmission attempt}$. The random integer r takes the value in the range $0 \leq r < 2k$
4	DC	Deferral Check When this bit is set, the deferral check function is enabled in the MAC. The MAC issues a Frame Abort status, along with the excessive deferral error bit set in the transmit frame status, when the transmit state machine is deferred for more than 24,288 bit times in the 10 or 100 Mbps mode. If the MAC is configured for 1000 Mbps operation or if the Jumbo frame mode is enabled in the 10 or 100 Mbps mode, the threshold for deferral is 155,680 bits times. Deferral begins when the transmitter is ready to transmit, but it is prevented because of an active carrier sense signal (CRS) on GMII or MII. The defer time is not cumulative. For example, if the transmitter defers for 10,000 bit times because the CRS signal is active and then the CRS signal becomes inactive, the transmitter transmits and collision happens. Because of collision, the transmitter needs to back off and then defer again after back off completion. In such a scenario, the deferral timer is reset to 0 and it is restarted.
3	TE	Transmitter Enable When this bit is set, the transmit state machine of the MAC is enabled for transmission on the GMII or MII. When this bit is reset, the MAC transmit state machine is disabled after the completion of the transmission of the current frame, and does not transmit any further frames.

Bits	Name	Description
2	RE	Receiver Enable When this bit is set, the receiver state machine of the MAC is enabled for receiving frames from the GMII or MII. When this bit is reset, the MAC receive state machine is disabled after the completion of the reception of the current frame, and does not receive any further frames from the GMII or MII.
1-0	PRELEN	Preamble Length for Transmit frames These bits control the number of preamble bytes that are added to the beginning of every Transmit frame. The preamble reduction occurs only when the MAC is operating in the full-duplex mode. - 2'b00: 7 bytes of preamble - 2'b01: 5 bytes of preamble - 2'b10: 3 bytes of preamble - 2'b11: Reserved

MACCFG Bits

55.7.2 MACFF (0x4)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
RA	RSVD										DNTU	IPFE	RSVD				VTFE	RSVD				HPF	SAF	SAIF	PCF		DBF	PM	DAIF	HMC	HUC	PR	
RW	N/A										RW	RW	N/A				RW	N/A				RW	RW	RW	RW		RW	RW	RW	RW	RW	RW	RW
0	x	x	x	x	x	x	x	x	x	0	0	x	x	x	x	0	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0		

MACFF [31:0]

Bits	Name	Description
31	RA	Receive All When this bit is set, the MAC Receiver module passes all received frames, irrespective of whether they pass the address filter or not, to the Application. The result of the SA or DA filtering is updated (pass or fail) in the corresponding bits in the Receive Status Word. When this bit is reset, the Receiver module passes only those frames to the Application that pass the SA or DA address filter.

Bits	Name	Description
21	DNTU	Drop non-TCP/UDP over IP Frames When set, this bit enables the MAC to drop the non-TCP or UDP over IP frames. The MAC forward only those frames that are processed by the Layer 4 filter. When reset, this bit enables the MAC to forward all non-TCP or UDP over IP frames.
20	IPFE	Layer 3 and Layer 4 Filter Enable When set, this bit enables the MAC to drop frames that do not match the enabled Layer 3 and Layer 4 filters. If Layer 3 or Layer 4 filters are not enabled for matching, this bit does not have any effect. When reset, the MAC forwards all frames irrespective of the match status of the Layer 3 and Layer 4 fields.
15	VTFE	VLAN Tag Filter Enable When set, this bit enables the MAC to drop VLAN tagged frames that do not match the VLAN Tag comparison. When reset, the MAC forwards all frames irrespective of the match status of the VLAN Tag.
10	HPF	Hash or Perfect Filter When this bit is set, it configures the address filter to pass a frame if it matches either the perfect filtering or the hash filtering as set by the HMC or HUC bits. When this bit is low and the HUC or HMC bit is set, the frame is passed only if it matches the Hash filter.
9	SAF	Source Address Filter Enable When this bit is set, the MAC compares the SA field of the received frames with the values programmed in the enabled SA registers. If the comparison fails, the MAC drops the frame. When this bit is reset, the MAC forwards the received frame to the application with updated SAF bit of the Rx Status depending on the SA address comparison.
8	SAIF	SA Inverse Filtering When this bit is set, the Address Check block operates in inverse filtering mode for the SA address comparison. The frames whose SA matches the SA registers are marked as failing the SA Address filter. When this bit is reset, frames whose SA does not match the SA registers are marked as failing the SA Address filter.

Bits	Name	Description
7-6	PCF	Pass Control Frames These bits control the forwarding of all control frames (including unicast and multicast Pause frames). - 00: MAC filters all control frames from reaching the application. - 01: MAC forwards all control frames except Pause frames to application even if they fail the Address filter. - 10: MAC forwards all control frames to application even if they fail the Address Filter. - 11: MAC forwards control frames that pass the Address Filter. The following conditions should be true for the Pause frames processing: - Condition 1: The MAC is in the full-duplex mode and flow control is enabled by setting Bit 2 (RFE) of Register 6 (Flow Control Register) to 1. - Condition 2: The destination address (DA) of the received frame matches the special multicast address or the MAC Address 0 when Bit 3 (UP) of the Register 6 (Flow Control Register) is set. - Condition 3: The Type field of the received frame is 0x8808 and the OPCODE field is 0x0001. Note: This field should be set to 01 only when the Condition 1 is true, that is, the MAC is programmed to operate in the full-duplex mode and the RFE bit is enabled. Otherwise, the Pause frame filtering may be inconsistent. When Condition 1 is false, the Pause frames are considered as generic control frames. Therefore, to pass all control frames (including Pause frames) when the full-duplex mode and flow control is not enabled, you should set the PCF field to 10 or 11 (as required by the application).
5	DBF	Disable Broadcast Frames When this bit is set, the AFM module blocks all incoming broadcast frames. In addition, it overrides all other filter settings. When this bit is reset, the AFM module passes all received broadcast frames.
4	PM	Pass All Multicast When set, this bit indicates that all received frames with a multicast destination address (first bit in the destination address field is '1') are passed. When reset, filtering of multicast frame depends on HMC bit.

Bits	Name	Description
3	DAIF	DA Inverse Filtering When this bit is set, the Address Check block operates in inverse filtering mode for the DA address comparison for both unicast and multicast frames. When reset, normal filtering of frames is performed.
2	HMC	Hash Multicast When set, the MAC performs destination address filtering of received multicast frames according to the hash table. When reset, the MAC performs a perfect destination address filtering for multicast frames, that is, it compares the DA field with the values programmed in DA registers.
1	HUC	Hash Unicast When set, the MAC performs destination address filtering of unicast frames according to the hash table. When reset, the MAC performs a perfect destination address filtering for unicast frames, that is, it compares the DA field with the values programmed in DA registers.
0	PR	Promiscuous Mode When this bit is set, the Address Filter module passes all incoming frames irrespective of the destination or source address. The SA or DA Filter Fails status bits of the Receive Status Word are always cleared when PR is set.

MACFF Bits

55.7.3 HASH_H (0x8)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HTH																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

HASH_H [31:0]

Bits	Name	Description
31-0	HTH	Hash Table High This field contains the upper 32 bits of the Hash table.

HASH_H Bits

55.7.4 HASH_L (0xC)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HTL																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

HASH_L [31:0]

Bits	Name	Description
31-0	HTL	Hash Table Low This field contains the lower 32 bits of the Hash table.

HASH_L Bits

55.7.5 GMII_ADDR (0x10)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																PA					GR					CR				GW	GB
N/A																RW					RW					RW				RW	RW
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GMII_ADDR [31:0]

Bits	Name	Description
15-11	PA	Physical Layer Address This field indicates which of the 32 possible PHY devices are being accessed. For RevMII, this field gives the PHY Address of the RevMII module.
10-6	GR	GMII Register These bits select the desired GMII register in the selected PHY device. For RevMII, these bits select the desired CSR register in the RevMII Registers set.

Bits	Name	Description
5-2	CR	CSR Clock Range The CSR Clock Range selection determines the frequency of the MDC clock according to the CSR clock frequency used in your design. The CSR clock corresponding to different GMAC configurations is given in Table 9-2 on page 564. The suggested range of CSR clock frequency applicable for each value (when Bit[5] = 0) ensures that the MDC clock is approximately between the frequency range 1.0 MHz~2.5 MHz. - 0000: The CSR clock frequency is 60~100 MHz and the MDC clock frequency is CSR clock/42. - 0001: The CSR clock frequency is 100~150 MHz and the MDC clock frequency is CSR clock/62. - 0010: The CSR clock frequency is 20~35 MHz and the MDC clock frequency is CSR clock/16. - 0011: The CSR clock frequency is 35~60 MHz and the MDC clock frequency is CSR clock/26. - 0100: The CSR clock frequency is 150~250 MHz and the MDC clock frequency is CSR clock/102. - 0101: The CSR clock frequency is 250~300 MHz and the MDC clock is CSR clock/124. - 0110, 0111: Reserved When Bit 5 is set, you can achieve higher frequency of the MDC clock than the frequency limit of 2.5 MHz (specified in the IEEE Std 802.3) and program a clock divider of lower value. For example, when CSR clock is of 100 MHz frequency and you program these bits as 1010, then the resultant MDC clock is of 12.5 MHz which is outside the limit of IEEE 802.3 specified range. Program the following values only if the interfacing chips support faster MDC clocks. - 1000: CSR clock/4 - 1001: CSR clock/6 - 1010: CSR clock/8 - 1011: CSR clock/10 - 1100: CSR clock/12 - 1101: CSR clock/14 - 1110: CSR clock/16 - 1111: CSR clock/18 These bits are not used for accessing RevMII. These bits are read-only if the RevMII interface is selected as single PHY interface.

Bits	Name	Description
1	GW	GMII Write When set, this bit indicates to the PHY or RevMII that this is a Write operation using the GMII Data register. If this bit is not set, it indicates that this is a Read operation, that is, placing the data in the GMII Data register.
0	GB	GMII Busy This bit should read logic 0 before writing to Register 4 and Register 5. During a PHY or RevMII register access, the software sets this bit to 1' b1 to indicate that a Read or Write access is in progress. Register 5 is invalid until this bit is cleared by the MAC. Therefore, Register 5 (GMII Data) should be kept valid until the MAC clears this bit during a PHY Write operation. Similarly for a read operation, the contents of Register 5 are not valid until this bit is cleared. The subsequent read or write operation should happen only after the previous operation is complete. Because there is no acknowledgment from the PHY to MAC after a read or write operation is completed, there is no change in the functionality of this bit even when the PHY is not present.

GMII_ADDR Bits

55.7.6 GMII_DATA (0x14)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																GD															
N/A																RW															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GMII_DATA [31:0]

Bits	Name	Description
15-0	GD	GMII Data This field contains the 16-bit data value read from the PHY or RevMII after a Management Read operation or the 16-bit data value to be written to the PHY or RevMII before a Management Write operation.

GMII_DATA Bits

55.7.7 FLOWCTRL (0x18)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PT																RSVD								DZPQ	RSVD	PLT		UP	RFE	TFE	FCB_BPA
RW																N/A								RW	N/A	RW		RW	RW	RW	RW
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x	0	x	0	0	0	0	0	0

FLOWCTRL [31:0]

Bits	Name	Description
31-16	PT	Pause Time This field holds the value to be used in the Pause Time field in the transmit control frame. If the Pause Time bits is configured to be double-synchronized to the (G)MII clock domain, then consecutive writes to this register should be performed only after at least four clock cycles in the destination clock domain.
7	DZPQ	Disable Zero-Quanta Pause When this bit is set, it disables the automatic generation of the Zero-Quanta Pause frames on the de-assertion of the flow-control signal from the FIFO layer (MTL or external sideband flow control signal <code>sb_d_flowctrl_i/mti_flowctrl_i</code>). When this bit is reset, normal operation with automatic Zero-Quanta Pause frame generation is enabled.

Bits	Name	Description
5-4	PLT	Pause Low Threshold This field configures the threshold of the Pause timer at which the input flow control signal mti_flowctrl_i (or sbd_flowctrl_i) is checked for automatic retransmission of the Pause frame. The threshold values should be always less than the Pause Time configured in Bits[31:16]. For example, if PT = 100H (256 slot-times), and PLT = 01, then a second Pause frame is automatically transmitted if the mti_flowctrl_i signal is asserted at 228 (256 - 28) slot times after the first Pause frame is transmitted. The following list provides the threshold values for different values: - 00: The threshold is Pause time minus 4 slot times (PT - 4 slot times). - 01: The threshold is Pause time minus 28 slot times (PT - 28 slot times). - 10: The threshold is Pause time minus 144 slot times (PT - 144 slot times). - 11: The threshold is Pause time minus 256 slot times (PT - 256 slot times). The slot time is defined as the time taken to transmit 512 bits (64 bytes) on the GMII or MII interface.
3	UP	Unicast Pause Frame Detect A pause frame is processed when it has the unique multicast address specified in the IEEE Std 802.3. When this bit is set, the MAC can also detect Pause frames with unicast address of the station. This unicast address should be as specified in the MAC Address0 High Register and MAC Address0 Low Register. When this bit is reset, the MAC only detects Pause frames with unique multicast address.
2	RFE	Receive Flow Control Enable When this bit is set, the MAC decodes the received Pause frame and disables its transmitter for a specified (Pause) time. When this bit is reset, the decode function of the Pause frame is disabled.

Bits	Name	Description
1	TFE	Transmit Flow Control Enable In the full-duplex mode, when this bit is set, the MAC enables the flow control operation to transmit Pause frames. When this bit is reset, the flow control operation in the MAC is disabled, and the MAC does not transmit any Pause frames. In the half-duplex mode, when this bit is set, the MAC enables the backpressure operation. When this bit is reset, the backpressure feature is disabled.
0	FCB_BPA	Flow Control Busy or Backpressure Activate This bit initiates a Pause frame in the full-duplex mode and activates the backpressure function in the half-duplex mode if the TFE bit is set. In the full-duplex mode, this bit should be read as 1'b0 before writing to the Flow Control register. To initiate a Pause frame, the Application must set this bit to 1'b1. During a transfer of the Control Frame, this bit continues to be set to signify that a frame transmission is in progress. After the completion of Pause frame transmission, the MAC resets this bit to 1'b0. The Flow Control register should not be written to until this bit is cleared. In the half-duplex mode, when this bit is set (and TFE is set), then backpressure is asserted by the MAC. During backpressure, when the MAC receives a new frame, the transmitter starts sending a JAM pattern resulting in a collision. This control register bit is logically ORed with the mti_flowctrl_i input signal for the backpressure function. When the MAC is configured for the full-duplex mode, the BPA is automatically disabled.

FLOWCTRL Bits

55.7.8 VLAN_TAG (0x1C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD												VTIM	ESVL	VTIM	ETV	VL															
N/A												RW	RW	RW	RW	RW															
x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VLAN_TAG [31:0]

Bits	Name	Description
19	VTHM	VLAN Tag Hash Table Match Enable When set, the most significant four bits of the VLAN tag's CRC are used to index the content of Register 354 (VLAN Hash Table Register). A value of 1 in the VLAN Hash Table register, corresponding to the index, indicates that the frame matched the VLAN hash table. When Bit 16 (ETV) is set, the CRC of the 12-bit VLAN Identifier (VID) is used for comparison whereas when ETV is reset, the CRC of the 16-bit VLAN tag is used for comparison. When reset, the VLAN Hash Match operation is not performed.
18	ESVL	Enable S-VLAN When this bit is set, the MAC transmitter and receiver also consider the S-VLAN (Type = 0x88A8) frames as valid VLAN tagged frames.
17	VTIM	VLAN Tag Inverse Match Enable When set, this bit enables the VLAN Tag inverse matching. The frames that do not have matching VLAN Tag are marked as matched. When reset, this bit enables the VLAN Tag perfect matching. The frames with matched VLAN Tag are marked as matched.
16	ETV	Enable 12-Bit VLAN Tag Comparison When this bit is set, a 12-bit VLAN identifier is used for comparing and filtering instead of the complete 16-bit VLAN tag. Bits [11:0] of VLAN tag are compared with the corresponding field in the received VLAN-tagged frame. Similarly, when enabled, only 12 bits of the VLAN tag in the received frame are used for hash-based VLAN filtering. When this bit is reset, all 16 bits of the 15th and 16th bytes of the received VLAN frame are used for comparison and VLAN hash filtering.

Bits	Name	Description
15-0	VL	VLAN Tag Identifier for Receive Frames This field contains the 802.1Q VLAN tag to identify the VLAN frames and is compared to the 15th and 16th bytes of the frames being received for VLAN frames. The following list describes the bits of this field: - Bits [15:13]: User Priority - Bit 12: Canonical Format Indicator (CFI) or Drop Eligible Indicator (DEI) - Bits[11:0]: VLAN tag's VLAN Identifier (VID) field When the ETV bit is set, only the VID (Bits[11:0]) is used for comparison. If VL (VL[11:0] if ETV is set) is all zeros, the MAC does not check the fifteenth and 16th bytes for VLAN tag comparison, and declares all frames with a Type field value of 0x8100 or 0x88a8 as VLAN frames.

VLAN_TAG Bits

55.7.9 RWKFRMFILT (0x28)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																RWKFRMFILT															
																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RWKFRMFILT [31:0]

Bits	Name	Description
31-0	WKUPFRMFILT	This is the address through which the application writes or reads the remote wake-up frame filter registers (wkupfmfilter_reg). The wkupfmfilter_reg register is a pointer to eight wkupfmfilter_reg registers. The wkupfmfilter_reg register is loaded by sequentially loading the eight register values. Eight sequential writes to this address (0x0028) write all wkupfmfilter_reg registers. Similarly, eight sequential reads from this address (0x0028) read all wkupfmfilter_reg registers

RWKFRMFILT Bits

55.7.10 PMT_CSR (0x2C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RWKFILTRST	RSVD		RWKPTR					RSVD														GLBLUCAST	RSVD		RWKPRCVD	MGKPRCVD	RSVD		RWKPKTEN	MGKPKTEN	PWRDWN
RW	N/A		RW					N/A														RW	N/A		RW	RW	N/A		RW	RW	RW
0	x	x	0	0	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	x	x	0	0	x	x	0	0	0

PMT_CSR [31:0]

Bits	Name	Description
31	RWKFILTRST	Remote Wake-Up Frame Filter Register Pointer Reset When this bit is set, it resets the remote wake-up frame filter register pointer to 3' b000. It is automatically cleared after 1 clock cycle.
28-24	RWKPTR	Remote Wake-up FIFO Pointer This field gives the current value (0 to 31) of the Remote Wake-up Frame filter register pointer. When the value of this pointer is equal to 7, 15, 23 or 31, the contents of the Remote Wake-up Frame Filter Register are transferred to the clk_rx_i domain when a write occurs to that register. The maximum value of the pointer is 7, 15, 23 and 31 respectively depending on the number of Remote Wakeup Filters selected during configuration.
9	GLBLUCAST	Global Unicast When set, enables any unicast packet filtered by the MAC (DAF) address recognition to be a remote wake-up frame.
6	RWKPRCVD	Remote Wake-Up Frame Received When set, this bit indicates the power management event is generated because of the reception of a remote wake-up frame. This bit is cleared by a Read into this register.
5	MGKPRCVD	Magic Packet Received When set, this bit indicates that the power management event is generated because of the reception of a magic packet. This bit is cleared by a Read into this register.
2	RWKPKTEN	Remote Wake-Up Frame Enable When set, enables generation of a power management event because of remote wake-up frame reception.
1	MGKPKTEN	Magic Packet Enable When set, enables generation of a power management event because of magic packet reception.

Bits	Name	Description
0	PWRDWN	Power Down When set, the MAC receiver drops all received frames until it receives the expected magic packet or remote wake-up frame. This bit is then self-cleared and the power-down mode is disabled. The Software can also clear this bit before the expected magic packet or remote wake-up frame is received. The frames, received by the MAC after this bit is cleared, are forwarded to the application. This bit must only be set when the Magic Packet Enable, Global Unicast, or Remote Wake-Up Frame Enable bit is set high. Note: You can gate-off the CSR clock during the power-down mode. However, when the CSR clock is gated-off, you cannot perform any read or write operations on this register. Therefore, the Software cannot clear this bit.

PMT_CSR Bits

55.7.11 LPI_CSR (0x30)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD												LPITXA	PLSEN	PLS	LPIEN	RSVD						RLPIST	TLPIST	RSVD			RLPIEX	RLPIEN	TLPIEX	TLPIEN	
N/A												RW	RW	RW	RW	N/A						RW	RW	N/A			RW	RW	RW	RW	
x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	x	x	x	x	x	x	0	0	x	x	x	x	0	0	0	0

LPI_CSR [31:0]

Bits	Name	Description
19	LPITXA	LPI TX Automate This bit controls the behavior of the MAC when it is entering or coming out of the LPI mode on the transmit side. This bit is not functional in the GMAC-CORE configuration in which the Tx clock gating is done during the LPI mode. If the LPITXA and LPIEN bits are set to 1, the MAC enters the LPI mode only after all outstanding frames (in the core) and pending frames (in the application interface) have been transmitted. The MAC comes out of the LPI mode when the application sends any frame for transmission or the application issues a TX FIFO Flush command. In addition, the MAC automatically clears the LPIEN bit when it exits the LPI state. If TX FIFO Flush is set in Bit 20 of Register 6 (Operation Mode Register), when the MAC is in the LPI mode, the MAC exits the LPI mode. When this bit is 0, the LPIEN bit directly controls behavior of the MAC when it is entering or coming out of the LPI mode.
18	PLSEN	PHY Link Status Enable This bit enables the link status received on the RGMII, SGMII, or SMII receive paths to be used for activating the LPI LS TIMER. When set, the MAC uses the link-status bits of Register 54 (SGMII/RGMII/SMII Control and Status Register) and Bit 17 (PLS) for the LPI LS Timer trigger. When cleared, the MAC ignores the link-status bits of Register 54 and takes only the PLS bit. This bit is RO and reserved if you have not selected the RGMII, SGMII, or SMII PHY interface.
17	PLS	PHY Link Status This bit indicates the link status of the PHY. The MAC Transmitter asserts the LPI pattern only when the link status is up (okay) at least for the time indicated by the LPI LS TIMER. When set, the link is considered to be okay (up) and when reset, the link is considered to be down.
16	LPIEN	LPI Enable When set, this bit instructs the MAC Transmitter to enter the LPI state. When reset, this bit instructs the MAC to exit the LPI state and resume normal transmission. This bit is cleared when the LPITXA bit is set and the MAC exits the LPI state because of the arrival of a new packet for transmission.

Bits	Name	Description
9	RLPIST	Receive LPI State When set, this bit indicates that the MAC is receiving the LPI pattern on the GMII or MII interface.
8	TLPIST	Transmit LPI State When set, this bit indicates that the MAC is transmitting the LPI pattern on the GMII or MII interface.
3	RLPIEX	Receive LPI Exit When set, this bit indicates that the MAC Receiver has stopped receiving the LPI pattern on the GMII or MII interface, exited the LPI state, and resumed the normal reception. This bit is cleared by a read into this register. Note: This bit may not get set if the MAC stops receiving the LPI pattern for a very short duration, such as, less than 3 clock cycles of CSR clock.
2	RLPIEN	Receive LPI Entry When set, this bit indicates that the MAC Receiver has received an LPI pattern and entered the LPI state. This bit is cleared by a read into this register. Note: This bit may not get set if the MAC stops receiving the LPI pattern for a very short duration, such as, less than 3 clock cycles of CSR clock.
1	TLPIEX	Transmit LPI Exit When set, this bit indicates that the MAC transmitter has exited the LPI state after the user has cleared the LPIEN bit and the LPI TW Timer has expired. This bit is cleared by a read into this register.
0	TLPIEN	Transmit LPI Entry When set, this bit indicates that the MAC Transmitter has entered the LPI state because of the setting of the LPIEN bit. This bit is cleared by a read into this register.

LPI_CSR Bits

55.7.12 LPI_TCR (0x34)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD						LST										TWT															
N/A						RW										RW															
x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LPI_TCR [31:0]

Bits	Name	Description
25-16	LST	LPI LS TIMER This field specifies the minimum time (in milliseconds) for which the link status from the PHY should be up (OKAY) before the LPI pattern can be transmitted to the PHY. The MAC does not transmit the LPI pattern even when the LPIEN bit is set unless the LPI LS Timer reaches the programmed terminal count. The default value of the LPI LS Timer is 1000 (1 sec) as defined in the IEEE standard.
15-0	TWT	LPI TW TIMER This field specifies the minimum time (in microseconds) for which the MAC waits after it stops transmitting the LPI pattern to the PHY and before it resumes the normal transmission. The TLPIEX status bit is set after the expiry of this timer.

LPI_TCR Bits

55.7.13 INTR_STATUS (0x38)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0											
RSVD																				GPIIS	LPIIS	TSIS	RSVD	MMCRXIPIS	MMCTXIS	MMCRXIS	MMCS	PMTIS	PCSANCIS	PCSLCHGIS	RGSMIIS											
N/A																				RO	RO	RO	N/A	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO				
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	x	0	0	0	0	0	0	0	0											

INTR_STATUS [31:0]

Bits	Name	Description
11	GPIIS	GPI Interrupt Status When the GPIO feature is enabled, this bit is set when any active event (LL or LH) occurs on the GPIS field (Bits [3:0]) of Register 56 (General Purpose IO Register) and the corresponding GPIE bit is enabled. This bit is cleared on reading lane 0 (GPIS) of Register 56 (General Purpose IO Register). When the GPIO feature is not enabled, this bit is reserved.

Bits	Name	Description
10	LPIIS	LPI Interrupt Status When the Energy Efficient Ethernet feature is enabled, this bit is set for any LPI state entry or exit in the MAC Transmitter or Receiver. This bit is cleared on reading Bit 0 of Register 12 (LPI Control and Status Register). In all other modes, this bit is reserved.
9	TSIS	Timestamp Interrupt Status When the Advanced Timestamp feature is enabled, this bit is set when any of the following conditions is true: - The system time value equals or exceeds the value specified in the Target Time High and Low registers. - There is an overflow in the seconds register. - The Auxiliary snapshot trigger is asserted. This bit is cleared on reading Bit 0 of Register 458 (Timestamp Status Register).
7	MMCRXIPIS	MMC Receive Checksum Offload Interrupt Status This bit is set high when an interrupt is generated in the MMC Receive Checksum Offload Interrupt Register. This bit is cleared when all the bits in this interrupt register are cleared.
6	MMCTXIS	MMC Transmit Interrupt Status This bit is set high when an interrupt is generated in the MMC Transmit Interrupt Register. This bit is cleared when all the bits in this interrupt register are cleared.
5	MMCRXIS	MMC Receive Interrupt Status This bit is set high when an interrupt is generated in the MMC Receive Interrupt Register. This bit is cleared when all the bits in this interrupt register are cleared.
4	MMCIS	MMC Interrupt Status This bit is set high when any of the Bits [7:5] is set high and cleared only when all of these bits are low.
3	PMTIS	PMT Interrupt Status This bit is set when a magic packet or remote wake-up frame is received in the power-down mode (see Bits 5 and 6 in the PMT Control and Status Register). This bit is cleared when both Bits[6:5] are cleared because of a read operation to the PMT Control and Status register.
2	PCSANCIS	PCS Auto-Negotiation Complete This bit is set when the Auto-negotiation is completed in the TBI, RTBI, or SGMII PHY interface (Bit 5 in Register 49 (AN Status Register)). This bit is cleared when you perform a read operation to the AN Status register.

Bits	Name	Description
1	PCSLCHGIS	PCS Link Status Changed This bit is set because of any change in Link Status in the TBI, RTBI, or SGMII PHY interface (Bit 2 in Register 49 (AN Status Register)). This bit is cleared when you perform a read operation on the AN Status register.
0	RGSMIIIS	RGMII or SMII Interrupt Status This bit is set because of any change in value of the Link Status of RGMII or SMII interface (Bit 3 in Register 54 (SGMII/RGMII/SMII Control and Status Register)). This bit is cleared when you perform a read operation on the SGMII/RGMII/SMII Control and Status Register.

INTR_STATUS Bits

55.7.14 INTR_MASK (0x3C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD										LPIIM		TSIM		RSVD										PMTIM		PCSANCIM		PCSLCHGIM		RGSMIIIM	
N/A										RW		RW		N/A										RW		RW		RW		RW	
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	x	x	x	x	x	0	0	0	0

INTR_MASK [31:0]

Bits	Name	Description
10	LPIIM	LPI Interrupt Mask When set, this bit disables the assertion of the interrupt signal because of the setting of the LPI Interrupt Status bit in Register 14 (Interrupt Status Register).
9	TSIM	Timestamp Interrupt Mask When set, this bit disables the assertion of the interrupt signal because of the setting of Timestamp Interrupt Status bit in Register 14 (Interrupt Status Register).
3	PMTIM	PMT Interrupt Mask When set, this bit disables the assertion of the interrupt signal because of the setting of PMT Interrupt Status bit in Register 14 (Interrupt Status Register).

Bits	Name	Description
2	PCSANCIM	PCS AN Completion Interrupt Mask When set, this bit disables the assertion of the interrupt signal because of the setting of PCS Auto-negotiation complete bit in Register 14 (Interrupt Status Register).
1	PCSLCHGIM	PCS Link Status Interrupt Mask When set, this bit disables the assertion of the interrupt signal because of the setting of the PCS Link-status changed bit in Register 14 (Interrupt Status Register).
0	RGSMIIIM	RGMII or SMII Interrupt Mask When set, this bit disables the assertion of the interrupt signal because of the setting of the RGMII or SMII Interrupt Status bit in Register 14 (Interrupt Status Register).

INTR_MASK Bits

55.7.15 MAC_ADDR_0_HIGH (0x40)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																ADDRHI															
N/A																RW															
0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MAC_ADDR_0_HIGH [31:0]

Bits	Name	Description
31	AE	Address Enable This bit is RO. The bit value is fixed at 1.
15-0	ADDRHI	MAC Address0 [47:32] This field contains the upper 16 bits (47:32) of the first 6-byte MAC address. The MAC uses this field for filtering the received frames and inserting the MAC address in the Transmit Flow Control (Pause) Frames.

MAC_ADDR_0_HIGH Bits

55.7.16 MAC_ADDR_0_LOW (0x44)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																ADDRLO																
																RW																

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MAC_ADDR_0_LOW [31:0]

Bits	Name	Description
31-0	ADDRLO	MAC Address0 [31:0] This field contains the lower 32 bits of the first 6-byte MAC address. This is used by the MAC for filtering the received frames and inserting the MAC address in the Transmit Flow Control (Pause) Frames.

MAC_ADDR_0_LOW Bits

55.7.17 MAC_ADDR[HIGH] (0x48 + 0x8 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AE	SA	MBC						RSVD								ADDRHI															
RW	RW	RW						N/A								RW															
0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MAC_ADDR[HIGH] [31:0]

Bits	Name	Description
31	AE	Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering.
30	SA	Source Address When this bit is set, the MAC Address1[47:0] is used to compare with the SA fields of the received frame. When this bit is reset, the MAC Address1[47:0] is used to compare with the DA fields of the received frame.

Bits	Name	Description
29-24	MBC	Mask Byte Control These bits are mask control bits for comparison of each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC Address1 registers. Each bit controls the masking of the bytes as follows: - Bit 29: Register 18[15:8] - Bit 28: Register 18[7:0] - Bit 27: Register 19[31:24] - ... - Bit 24: Register 19[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
15-0	ADDRHI	MAC Address1 [47:32] This field contains the upper 16 bits (47:32) of the second 6-byte MAC address.

MAC_ADDR[HIGH] Bits

55.7.18 MAC_ADDR[LOW] (0x4C + 0x8 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRLO																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MAC_ADDR[LOW] [31:0]

Bits	Name	Description
31-0	ADDRLO	MAC Address1 [31:0] This field contains the lower 32 bits of the second 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

MAC_ADDR[LOW] Bits

55.7.19 XMII_CSR (0xD8)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																										FALSCARDET	JABTO	LNKSTS	LNKSPEED	LNKMOD	
N/A																										RW	RW	RW	RW	RW	
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0

XMII_CSR [31:0]

Bits	Name	Description
5	FALSCARDET	False Carrier Detected This bit indicates whether the SMII PHY detected false carrier (1'b1). This bit is reserved when the MAC is configured for the SGMII or RGMII PHY interface.
4	JABTO	Jabber Timeout This bit indicates whether there is jabber timeout error (1'b1) in the received frame. This bit is reserved when the MAC is configured for the SGMII or RGMII PHY interface.
3	LNKSTS	Link Status This bit indicates whether the link between the local PHY and the remote PHY is up or down. It gives the status of the link between the SGMII of MAC and the SGMII of the local PHY. The status bits are received from the local PHY during ANEG between the MAC and PHY on the SGMII link.
2-1	LNKSPEED	Link Speed This bit indicates the current speed of the link: - 00: 2.5 MHz - 01: 25 MHz - 10: 125 MHz Bit 2 is reserved when the MAC is configured for the SMII PHY interface.
0	LNKMOD	Link Mode This bit indicates the current mode of operation of the link: - 1' b0: Half-duplex mode - 1' b1: Full-duplex mode

XMII_CSR Bits

55.7.20 WDOG_WTO (0xDC)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD															PWE	RSVD	WTO														

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
N/A															RW	N/A		RW													
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0

WDOG_WTO [31:0]

Bits	Name	Description
16	PWE	Programmable Watchdog Enable When this bit is set and Bit 23 (WD) of Register 0 (MAC Configuration Register) is reset, the WTO field (Bits[13:0]) is used as watchdog timeout for a received frame. When this bit is cleared, the watchdog timeout for a received frame is controlled by the setting of Bit 23 (WD) and Bit 20 (JE) in Register 0 (MAC Configuration Register).
13-0	WTO	Watchdog Timeout When Bit 16 (PWE) is set and Bit 23 (WD) of Register 0 (MAC Configuration Register) is reset, this field is used as watchdog timeout for a received frame. If the length of a received frame exceeds the value of this field, such frame is terminated and declared as an error frame. Note: When Bit 16 (PWE) is set, the value in this field should be more than 1,522 (0x05F2). Otherwise, the IEEE Std 802.3-specified valid tagged frames are declared as error frames and are dropped.

WDOG_WTO Bits

55.7.21 MMC_CNTRL (0x100)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RSVD																							UCDBC	RSVD		CNTPRSTLVL	CNTPRST	CNTPFREEZ	RSTONRD	CNTPSTOPRO	CNTPRST	
N/A																							RW	N/A		RW	RW	RW	RW	RW	RW	RW
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	x	x	0	0	0	0	0	0	

MMC_CNTRL [31:0]

Bits	Name	Description
8	UCDBC	Update MMC Counters for Dropped Broadcast Frames When set, the MAC updates all related MMC Counters for Broadcast frames that are dropped because of the setting of Bit 5 (DBF) of Register 1 (MAC Frame Filter). When reset, the MMC Counters are not updated for dropped Broadcast frames.
5	CNTPRSTLVL	Full-Half Preset When this bit is low and Bit 4 is set, all MMC counters get preset to almost-half value. All octet counters get preset to 0x7FFF_F800 (half - 2KBytes) and all frame-counters gets preset to 0x7FFF_FFF0 (half - 16). When this bit is high and Bit 4 is set, all MMC counters get preset to almost-full value. All octet counters get preset to 0xFFFF_F800 (full - 2KBytes) and all frame-counters gets preset to 0xFFFF_FFF0 (full - 16). For 16-bit counters, the almost-half preset values are 0x7800 and 0x7FF0 for the respective octet and frame counters. Similarly, the almost-full preset values for the 16-bit counters are 0xF800 and 0xFFF0.
4	CNTPRST	Counters Preset When this bit is set, all counters are initialized or preset to almost full or almost half according to Bit 5. This bit is cleared automatically after 1 clock cycle. This bit, along with Bit 5, is useful for debugging and testing the assertion of interrupts because of MMC counter becoming half-full or full.
3	CNTFREEZ	MMC Counter Freeze When this bit is set, it freezes all MMC counters to their current value. Until this bit is reset to 0, no MMC counter is updated because of any transmitted or received frame. If any MMC counter is read with the Reset on Read bit set, then that counter is also cleared in this mode.
2	RSTONRD	Reset on Read When this bit is set, the MMC counters are reset to zero after Read (self-clearing after reset). The counters are cleared when the least significant byte lane (Bits[7:0]) is read.

Bits	Name	Description
1	CNTSTOPRO	Counter Stop Rollover When this bit is set, the counter does not roll over to zero after reaching the maximum value.
0	CNTRST	Counters Reset When this bit is set, all counters are reset. This bit is cleared automatically after 1 clock cycle

MMC_CNTRL Bits

55.7.22 MMC_INTR_RX (0x104)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD						RXCTRLFIS	RXRCVERRFIS	RXWDOGFIS	RXVLANGBFIS	RXFOVFIS	RXPAUSFIS	RXORANGFIS	RXLENERFIS	RXUCGFIS	RX1024TMAXOCTGBFIS	RX512T1023OCTGBFIS	RX256T511OCTGBFIS	RX128T255OCTGBFIS	RX65T127OCTGBFIS	RX64OCTGBFIS	RXOSIZEGFIS	RXUSIZEGFIS	RXJABERFIS	RXRUNTFIS	RXALGNERFIS	RXCRCERFIS	RXMCGFIS	RXBCGFIS	RXGOCTIS	RXGBOCTIS	RXGBFRMIS
N/A						RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MMC_INTR_RX [31:0]

Bits	Name	Description
25	RXCTRLFIS	MMC Receive Control Frame Counter Interrupt Status This bit is set when the rxctrlframes_g counter reaches half of the maximum value or the maximum value.
24	RXRCVERRFIS	MMC Receive Error Frame Counter Interrupt Status This bit is set when the rxrcverror counter reaches half of the maximum value or the maximum value.
23	RXWDOGFIS	MMC Receive Watchdog Error Frame Counter Interrupt Status This bit is set when the rxwatchdog error counter reaches half of the maximum value or the maximum value.
22	RXVLANGBFIS	MMC Receive VLAN Good Bad Frame Counter Interrupt Status This bit is set when the rxvlanframes_gb counter reaches half of the maximum value or the maximum value.
21	RXFOVFIS	MMC Receive FIFO Overflow Frame Counter Interrupt Status This bit is set when the rxfifooverflow counter reaches half of the maximum value or the maximum value.
20	RXPAUSFIS	MMC Receive Pause Frame Counter Interrupt Status This bit is set when the rxpauseframes counter reaches half of the maximum value or the maximum value.

Bits	Name	Description
19	RXORANGEFIS	MMC Receive Out Of Range Error Frame Counter Interrupt Status. This bit is set when the rxoutofrangetype counter reaches half of the maximum value or the maximum value.
18	RXLENERFIS	MMC Receive Length Error Frame Counter Interrupt Status This bit is set when the rxlengtherror counter reaches half of the maximum value or the maximum value.
17	RXUCGFIS	MMC Receive Unicast Good Frame Counter Interrupt Status This bit is set when the rxunicastframes_g counter reaches half of the maximum value or the maximum value.
16	RX1024TMAXOCTGBFIS	MMC Receive 1024 to Maximum Octet Good Bad Frame Counter Interrupt Status. This bit is set when the rx1024tomaxoctets_gb counter reaches half of the maximum value or the maximum value.
15	RX512T1023OCTGBFIS	MMC Receive 512 to 1023 Octet Good Bad Frame Counter Interrupt Status This bit is set when the rx512to1023octets_gb counter reaches half of the maximum value or the maximum value.
14	RX256T511OCTGBFIS	MMC Receive 256 to 511 Octet Good Bad Frame Counter Interrupt Status This bit is set when the rx256to511octets_gb counter reaches half of the maximum value or the maximum value.
13	RX128T255OCTGBFIS	MMC Receive 128 to 255 Octet Good Bad Frame Counter Interrupt Status This bit is set when the rx128to255octets_gb counter reaches half of the maximum value or the maximum value.
12	RX65T127OCTGBFIS	MMC Receive 65 to 127 Octet Good Bad Frame Counter Interrupt Status This bit is set when the rx65to127octets_gb counter reaches half of the maximum value or the maximum value.
11	RX64OCTGBFIS	MMC Receive 64 Octet Good Bad Frame Counter Interrupt Status This bit is set when the rx64octets_gb counter reaches half of the maximum value or the maximum value.
10	RXOSIZEGFIS	MMC Receive Oversize Good Frame Counter Interrupt Status This bit is set when the rxoversize_g counter reaches half of the maximum value or the maximum value.
9	RXUSIZEGFIS	MMC Receive Undersize Good Frame Counter Interrupt Status This bit is set when the rxundersize_g counter reaches half of the maximum value or the maximum value.

Bits	Name	Description
8	RXJABERFIS	MMC Receive Jabber Error Frame Counter Interrupt Status This bit is set when the rxjabbererror counter reaches half of the maximum value or the maximum value.
7	RXRUNTFIS	MMC Receive Runt Frame Counter Interrupt Status This bit is set when the rxrunterror counter reaches half of the maximum value or the maximum value.
6	RXALGNERFIS	MMC Receive Alignment Error Frame Counter Interrupt Status This bit is set when the rxalignmenterror counter reaches half of the maximum value or the maximum value.
5	RXCRCERFIS	MMC Receive CRC Error Frame Counter Interrupt Status This bit is set when the rxcrcerror counter reaches half of the maximum value or the maximum value.
4	RXMCGFIS	MMC Receive Multicast Good Frame Counter Interrupt Status This bit is set when the rxmulticastframes_g counter reaches half of the maximum value or the maximum value.
3	RXBCGFIS	MMC Receive Broadcast Good Frame Counter Interrupt Status This bit is set when the rxbroadcastframes_g counter reaches half of the maximum value or the maximum value.
2	RXGOCTIS	MMC Receive Good Octet Counter Interrupt Status This bit is set when the rxoctetcount_g counter reaches half of the maximum value or the maximum value.
1	RXGBOCTIS	MMC Receive Good Bad Octet Counter Interrupt Status This bit is set when the rxoctetcount_gb counter reaches half of the maximum value or the maximum value.
0	RXGBFRMIS	MMC Receive Good Bad Frame Counter Interrupt Status This bit is set when the rxframecount_gb counter reaches half of the maximum value or the maximum value.

MMC_INTR_RX Bits

55.7.23 MMC_INTR_TX (0x108)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD						TXOSIZEFIS	TXVLANGFIS	TXPAUSFIS	TXEXDEFFIS	TXGFRMIS	TXGOCTIS	TXCARERFIS	TXEXCOLFIS	TXLATCOLFIS	TXDEFFIS	TXMCOLGFI	TXSCOLGFI	TXUFLOWERFIS	TXBCGBFIS	TXMCGBFIS	TXUCGBFIS	TX1024TMAXOCTGBFIS	TX512T1023OCTGBFIS	TX256T511OCTGBFIS	TX128T255OCTGBFIS	TX65T127OCTGBFIS	TX64OCTGBFIS	TXMCGFIS	TXBCGFIS	TXGBFRMIS	TXGBOCTIS
N/A						RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

MMC_INTR_TX [31:0]

Bits	Name	Description
25	TXOSIZEGFIS	MMC Transmit Oversize Good Frame Counter Interrupt Status This bit is set when the txoversize_g counter reaches half of the maximum value or the maximum value.
24	TXVLANGFIS	MMC Transmit VLAN Good Frame Counter Interrupt Status This bit is set when the txvlanframes_g counter reaches half of the maximum value or the maximum value.
23	TXPAUSFIS	MMC Transmit Pause Frame Counter Interrupt Status This bit is set when the txpauseframeserror counter reaches half of the maximum value or the maximum value.
22	TXEXDEFFIS	MMC Transmit Excessive Deferral Frame Counter Interrupt Status This bit is set when the txexcessdef counter reaches half of the maximum value or the maximum value.
21	TXGFRMIS	MMC Transmit Good Frame Counter Interrupt Status This bit is set when the txframecount_g counter reaches half of the maximum value or the maximum value.
20	TXGOCTIS	MMC Transmit Good Octet Counter Interrupt Status This bit is set when the txoctetcount_g counter reaches half of the maximum value or the maximum value.
19	TXCARERFIS	MMC Transmit Carrier Error Frame Counter Interrupt Status This bit is set when the txcarriererror counter reaches half of the maximum value or the maximum value.
18	TXEXCOLFIS	MMC Transmit Excessive Collision Frame Counter Interrupt Status This bit is set when the txexesscol counter reaches half of the maximum value or the maximum value.
17	TXLATCOLFIS	MMC Transmit Late Collision Frame Counter Interrupt Status This bit is set when the txlatecol counter reaches half of the maximum value or the maximum value.
16	TXDEFFIS	MMC Transmit Deferred Frame Counter Interrupt Status This bit is set when the txdeferred counter reaches half of the maximum value or the maximum value.
15	TXMCOLGFIS	MMC Transmit Multiple Collision Good Frame Counter Interrupt Status This bit is set when the txmulticol_g counter reaches half of the maximum value or the maximum value.

Bits	Name	Description
14	TXSCOLGFIS	MMC Transmit Single Collision Good Frame Counter Interrupt Status This bit is set when the txsinglecol_g counter reaches half of the maximum value or the maximum value.
13	TXUFLOWERFIS	MMC Transmit Underflow Error Frame Counter Interrupt Status This bit is set when the txunderflowerror counter reaches half of the maximum value or the maximum value.
12	TXBCGBFIS	MMC Transmit Broadcast Good Bad Frame Counter Interrupt Status This bit is set when the txbroadcastframes_gb counter reaches half of the maximum value or the maximum value.
11	TXMCGBFIS	MMC Transmit Multicast Good Bad Frame Counter Interrupt Status The bit is set when the txmulticastframes_gb counter reaches half of the maximum value or the maximum value.
10	TXUCGBFIS	MMC Transmit Unicast Good Bad Frame Counter Interrupt Status This bit is set when the txunicastframes_gb counter reaches half of the maximum value or the maximum value.
9	TX1024TMAXOCTGBFIS	MMC Transmit 1024 to Maximum Octet Good Bad Frame Counter Interrupt Status This bit is set when the tx1024tomaxoctets_gb counter reaches half of the maximum value or the maximum value.
8	TX512T1023OCTGBFIS	MMC Transmit 512 to 1023 Octet Good Bad Frame Counter Interrupt Status This bit is set when the tx512to1023octets_gb counter reaches half of the maximum value or the maximum value.
7	TX256T511OCTGBFIS	MMC Transmit 256 to 511 Octet Good Bad Frame Counter Interrupt Status This bit is set when the tx256to511octets_gb counter reaches half of the maximum value or the maximum value.
6	TX128T255OCTGBFIS	MMC Transmit 128 to 255 Octet Good Bad Frame Counter Interrupt Status This bit is set when the tx128to255octets_gb counter reaches half of the maximum value or the maximum value.
5	TX65T127OCTGBFIS	MMC Transmit 65 to 127 Octet Good Bad Frame Counter Interrupt Status This bit is set when the tx65to127octets_gb counter reaches half the maximum value, and also when it reaches the maximum value.

Bits	Name	Description
4	TX64OCTGBFIS	MMC Transmit 64 Octet Good Bad Frame Counter Interrupt Status This bit is set when the tx64octets_gb counter reaches half of the maximum value or the maximum value.
3	TXMCGFIS	MMC Transmit Multicast Good Frame Counter Interrupt Status This bit is set when the txmulticastframes_g counter reaches half of the maximum value or the maximum value.
2	TXBCGFIS	MMC Transmit Broadcast Good Frame Counter Interrupt Status This bit is set when the txbroadcastframes_g counter reaches half of the maximum value or the maximum value.
1	TXGBFRMIS	MMC Transmit Good Bad Frame Counter Interrupt Status This bit is set when the txframecount_gb counter reaches half of the maximum value or the maximum value.
0	TXGBOCTIS	MMC Transmit Good Bad Octet Counter Interrupt Status This bit is set when the txoctetcount_gb counter reaches half of the maximum value or the maximum value.

MMC_INTR_TX Bits

55.7.24 MMC_INTR_MASK_RX (0x10C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD						RXCTRLFIM	RXRCVERRFIM	RXWDOGFIM	RXVLANGBFIM	RXFOVFIM	RXPUSFIM	RXORANGFIM	RXLENERFIM	RXUCGFIM	RX1024TMAXOCTGBFIM	RX512T1023OCTGBFIM	RX256T511OCTGBFIM	RX128T255OCTGBFIM	RX65T127OCTGBFIM	RX64OCTGBFIM	RXOSIZEGFIM	RXUSIZEGFIM	RXJABERFIM	RXRUNTFIM	RXALGNERFIM	RXCRCERFIM	RXMCGFIM	RXBCGFIM	RXGOCTIM	RXGBOCTIM	RSVD
N/A						RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	N/A
x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x

MMC_INTR_MASK_RX [31:0]

Bits	Name	Description
25	RXCTRLFIM	MMC Receive Control Frame Counter Interrupt Mask Setting this bit masks the interrupt when the rxctrlframes_g counter reaches half of the maximum value or the maximum value.
24	RXRCVERRFIM	MMC Receive Error Frame Counter Interrupt Mask Setting this bit masks the interrupt when the rxrcverror counter reaches half of the maximum value or the maximum value.

Bits	Name	Description
23	RXWDOGFIM	MMC Receive Watchdog Error Frame Counter Interrupt Mask Setting this bit masks the interrupt when the rxwatchdog counter reaches half of the maximum value or the maximum value.
22	RXVLANGBFIM	MMC Receive VLAN Good Bad Frame Counter Interrupt Mask Setting this bit masks the interrupt when the rxvlanframes_gb counter reaches half of the maximum value or the maximum value.
21	RXFOVFIM	MMC Receive FIFO Overflow Frame Counter Interrupt Mask Setting this bit masks the interrupt when the rxfifooverflow counter reaches half of the maximum value or the maximum value.
20	RXPAUSFIM	MMC Receive Pause Frame Counter Interrupt Mask Setting this bit masks the interrupt when the rxpauseframes counter reaches half of the maximum value or the maximum value.
19	RXORANGEFIM	MMC Receive Out Of Range Error Frame Counter Interrupt Mask Setting this bit masks the interrupt when the rxoutofrangetype counter reaches half of the maximum value or the maximum value.
18	RXLENERFIM	MMC Receive Length Error Frame Counter Interrupt Mask Setting this bit masks the interrupt when the rxlengtherror counter reaches half of the maximum value or the maximum value.
17	RXUCGFIM	MMC Receive Unicast Good Frame Counter Interrupt Mask Setting this bit masks the interrupt when the rxunicastframes_g counter reaches half of the maximum value or the maximum value.
16	RX1024TMAXOCTGBFIM	MMC Receive 1024 to Maximum Octet Good Bad Frame Counter Interrupt Mask. Setting this bit masks the interrupt when the rx1024tomaxoctets_gb counter reaches half of the maximum value or the maximum value.
15	RX512T1023OCTGBFIM	MMC Receive 512 to 1023 Octet Good Bad Frame Counter Interrupt Mask Setting this bit masks the interrupt when the rx512to1023octets_gb counter reaches half of the maximum value or the maximum value.

Bits	Name	Description
14	RX256T511OCTGBFIM	MMC Receive 256 to 511 Octet Good Bad Frame Counter Interrupt Mask Setting this bit masks the interrupt when the rx256to511octets_gb counter reaches half of the maximum value or the maximum value.
13	RX128T255OCTGBFIM	MMC Receive 128 to 255 Octet Good Bad Frame Counter Interrupt Mask Setting this bit masks the interrupt when the rx128to255octets_gb counter reaches half of the maximum value or the maximum value.
12	RX65T127OCTGBFIM	MMC Receive 65 to 127 Octet Good Bad Frame Counter Interrupt Mask Setting this bit masks the interrupt when the rx65to127octets_gb counter reaches half of the maximum value or the maximum value.
11	RX64OCTGBFIM	MMC Receive 64 Octet Good Bad Frame Counter Interrupt Mask Setting this bit masks the interrupt when the rx64octets_gb counter reaches half of the maximum value or the maximum value.
10	RXOSIZEGFIM	MMC Receive Oversize Good Frame Counter Interrupt Mask Setting this bit masks the interrupt when the rxoversize_g counter reaches half of the maximum value or the maximum value.
9	RXUSIZEGFIM	MMC Receive Undersize Good Frame Counter Interrupt Mask Setting this bit masks the interrupt when the rxundersize_g counter reaches half of the maximum value or the maximum value.
8	RXJABERFIM	MMC Receive Jabber Error Frame Counter Interrupt Mask Setting this bit masks the interrupt when the rxjabbererror counter reaches half of the maximum value or the maximum value.
7	RXRUNTFIM	MMC Receive Runt Frame Counter Interrupt Mask Setting this bit masks the interrupt when the rxrunterror counter reaches half of the maximum value or the maximum value.
6	RXALGNERFIM	MMC Receive Alignment Error Frame Counter Interrupt Mask Setting this bit masks the interrupt when the rxalignmenterror counter reaches half of the maximum value or the maximum value.
5	RXCRCERFIM	MMC Receive CRC Error Frame Counter Interrupt Mask Setting this bit masks the interrupt when the rxcrcerror counter reaches half of the maximum value or the maximum value.

Bits	Name	Description
4	RXMCGFIM	MMC Receive Multicast Good Frame Counter Interrupt Mask Setting this bit masks the interrupt when the rxmulticastframes_g counter reaches half of the maximum value or the maximum value.
3	RXBCGFIM	MMC Receive Broadcast Good Frame Counter Interrupt Mask Setting this bit masks the interrupt when the rxbroadcastframes_g counter reaches half of the maximum value or the maximum value.
2	RXGOCTIM	MMC Receive Good Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxoctetcount_g counter reaches half of the maximum value or the maximum value.
1	RXGBOCTIM	MMC Receive Good Bad Octet Counter Interrupt Mask. Setting this bit masks the interrupt when the rxoctetcount_gb counter reaches half of the maximum value or the maximum value.

MMC_INTR_MASK_RX Bits

55.7.25 MMC_INTR_MASK_TX (0x110)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD						TXOSIZEGFM	TXVLANGFM	TXPAUSFM	TXEXDEFFM	TXGFRMIM	TXGOCTIM	TXCARERFIM	TXEXCOLFIM	TXLATCOLFIM	TXDEFFM	TXMCOLGFM	TXSCOLGFM	TXUFLOWERFIM	TXBCGBFIM	TXMCGBFIM	TXUCGBFIM	TX1024TMAXOCTGBFIM	TX512T1023OCTGBFIM	TX256T511OCTGBFIM	TX128T255OCTGBFIM	TX65T127OCTGBFIM	TX64OCTGBFIM	TXMCGFM	TXBCGFM	TXGBFRMIM	TXGBOCTIM
N/A						RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MMC_INTR_MASK_TX [31:0]

Bits	Name	Description
25	TXOSIZEGFM	MMC Transmit Oversize Good Frame Counter Interrupt Mask Setting this bit masks the interrupt when the txoversize_g counter reaches half of the maximum value or the maximum value.
24	TXVLANGFM	MMC Transmit VLAN Good Frame Counter Interrupt Mask Setting this bit masks the interrupt when the txvlanframes_g counter reaches half of the maximum value or the maximum value.

Bits	Name	Description
23	TXPAUSFIM	MMC Transmit Pause Frame Counter Interrupt Mask Setting this bit masks the interrupt when the txpauseframes counter reaches half of the maximum value or the maximum value.
22	TXEXDEFFIM	MMC Transmit Excessive Deferral Frame Counter Interrupt Mask Setting this bit masks the interrupt when the txexcessdef counter reaches half of the maximum value or the maximum value.
21	TXGFRMIM	MMC Transmit Good Frame Counter Interrupt Mask Setting this bit masks the interrupt when the txframecount_g counter reaches half of the maximum value or the maximum value.
20	TXGOCTIM	MMC Transmit Good Octet Counter Interrupt Mask Setting this bit masks the interrupt when the txoctetcount_g counter reaches half of the maximum value or the maximum value.
19	TXCARERFIM	MMC Transmit Carrier Error Frame Counter Interrupt Mask Setting this bit masks the interrupt when the txcarriererror counter reaches half of the maximum value or the maximum value.
18	TXEXCOLFIM	MMC Transmit Excessive Collision Frame Counter Interrupt Mask Setting this bit masks the interrupt when the txexcesscol counter reaches half of the maximum value or the maximum value.
17	TXLATCOLFIM	MMC Transmit Late Collision Frame Counter Interrupt Mask Setting this bit masks the interrupt when the txlatecol counter reaches half of the maximum value or the maximum value.
16	TXDEFFIM	MMC Transmit Deferred Frame Counter Interrupt Mask Setting this bit masks the interrupt when the txdeferred counter reaches half of the maximum value or the maximum value.
15	TXMCOLGFIM	MMC Transmit Multiple Collision Good Frame Counter Interrupt Mask Setting this bit masks the interrupt when the txmulticol_g counter reaches half of the maximum value or the maximum value.
14	TXSCOLGFIM	MMC Transmit Single Collision Good Frame Counter Interrupt Mask Setting this bit masks the interrupt when the txsinglecol_g counter reaches half of the maximum value or the maximum value.

Bits	Name	Description
13	TXUFLOWERFIM	MMC Transmit Underflow Error Frame Counter Interrupt Mask Setting this bit masks the interrupt when the txunderflowerror counter reaches half of the maximum value or the maximum value.
12	TXBCGBFIM	MMC Transmit Broadcast Good Bad Frame Counter Interrupt Mask Setting this bit masks the interrupt when the txbroadcastframes_gb counter reaches half of the maximum value or the maximum value.
11	TXMCGBFIM	MMC Transmit Multicast Good Bad Frame Counter Interrupt Mask Setting this bit masks the interrupt when the txmulticastframes_gb counter reaches half of the maximum value or the maximum value.
10	TXUCGBFIM	MMC Transmit Unicast Good Bad Frame Counter Interrupt Mask Setting this bit masks the interrupt when the txunicastframes_gb counter reaches half of the maximum value or the maximum value.
9	TX1024TMAXOCTGBFIM	MMC Transmit 1024 to Maximum Octet Good Bad Frame Counter Interrupt Mask Setting this bit masks the interrupt when the tx1024tomaxoctets_gb counter reaches half of the maximum value or the maximum value.
8	TX512T1023OCTGBFIM	MMC Transmit 512 to 1023 Octet Good Bad Frame Counter Interrupt Mask Setting this bit masks the interrupt when the tx512to1023octets_gb counter reaches half of the maximum value or the maximum value.
7	TX256T511OCTGBFIM	MMC Transmit 256 to 511 Octet Good Bad Frame Counter Interrupt Mask Setting this bit masks the interrupt when the tx256to511octets_gb counter reaches half of the maximum value or the maximum value.
6	TX128T255OCTGBFIM	MMC Transmit 128 to 255 Octet Good Bad Frame Counter Interrupt Mask Setting this bit masks the interrupt when the tx128to255octets_gb counter reaches half of the maximum value or the maximum value.

Bits	Name	Description
5	TX65T127OCTGBFIM	MMC Transmit 65 to 127 Octet Good Bad Frame Counter Interrupt Mask Setting this bit masks the interrupt when the tx65to127octets_gb counter reaches half of the maximum value or the maximum value.
4	TX64OCTGBFIM	MMC Transmit 64 Octet Good Bad Frame Counter Interrupt Mask Setting this bit masks the interrupt when the tx64octets_gb counter reaches half of the maximum value or the maximum value.
3	TXMCGFIM	MMC Transmit Multicast Good Frame Counter Interrupt Mask Setting this bit masks the interrupt when the txmulticastframes_g counter reaches half of the maximum value or the maximum value.
2	TXBCGFIM	MMC Transmit Broadcast Good Frame Counter Interrupt Mask Setting this bit masks the interrupt when the txbroadcastframes_g counter reaches half of the maximum value or the maximum value.
1	TXGBFRMIM	MMC Transmit Good Bad Frame Counter Interrupt Mask Setting this bit masks the interrupt when the txframecount_gb counter reaches half of the maximum value or the maximum value.
0	TXGBOCTIM	MMC Transmit Good Bad Octet Counter Interrupt Mask Setting this bit masks the interrupt when the txoctetcount_gb counter reaches half of the maximum value or the maximum value.

MMC_INTR_MASK_TX Bits

55.7.26 TX64OCTETS_GB (0x124)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FRMCNT																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TX64OCTETS_GB [31:0]

Bits	Name	Description
31-0	FRMCNT	Number of good and bad frames transmitted with length 64 bytes, exclusive of preamble and retried frames.

Bits	Name	Description

TX64OCTETS_GB Bits

55.7.27 TX65TO127OCTETS_GB (0x128)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FRMCNT																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

TX65TO127OCTETS_GB [31:0]

Bits	Name	Description
31-0	FRMCNT	Number of good and bad frames transmitted with length between 65 and 127 (inclusive) bytes, exclusive of preamble and retried frames.

TX65TO127OCTETS_GB Bits

55.7.28 TX128TO255OCTETS_GB (0x12C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FRMCNT																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TX128TO255OCTETS_GB [31:0]

Bits	Name	Description
31-0	FRMCNT	Number of good and bad frames transmitted with length between 128 and 255 (inclusive) bytes, exclusive of preamble and retried frames.

TX128TO255OCTETS_GB Bits

55.7.29 TX256TO511OCTETS_GB (0x130)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FRMCNT																															
RW																															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TX256TO511OCTETS_GB [31:0]

Bits	Name	Description
31-0	FRMCNT	Number of good and bad frames transmitted with length between 256 and 511 (inclusive) bytes, exclusive of preamble and retried frames.

TX256TO511OCTETS_GB Bits

55.7.30 TX512TO1023OCTETS_GB (0x134)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																FRMCNT															
																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TX512TO1023OCTETS_GB [31:0]

Bits	Name	Description
31-0	FRMCNT	Number of good and bad frames transmitted with length between 512 and 1,023 (inclusive) bytes, exclusive of preamble and retried frames.

TX512TO1023OCTETS_GB Bits

55.7.31 TX1024TOMAXOCTETS_GB (0x138)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																FRMCNT															
																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TX1024TOMAXOCTETS_GB [31:0]

Bits	Name	Description
31-0	FRMCNT	Number of good and bad frames transmitted with length between 1,024 and maxsize (inclusive) bytes, exclusive of preamble and retried frames.

TX1024TOMAXOCTETS_GB Bits

55.7.32 RXFRAMECOUNT_GB (0x180)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FRMCNT																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RXFRAMECOUNT_GB [31:0]

Bits	Name	Description
31-0	FRMCNT	Number of good and bad frames received.

RXFRAMECOUNT_GB Bits

55.7.33 MMC_IPC_INTR_MASK_RX (0x200)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																RXIPV4GFIM															
N/A																RW															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0

MMC_IPC_INTR_MASK_RX [31:0]

Bits	Name	Description
0	RXIPV4GFIM	MMC Receive IPV4 Good Frame Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv4_gd_frms counter reaches half of the maximum value or the maximum value.

MMC_IPC_INTR_MASK_RX Bits

55.7.34 MMC_IPC_INTR_RX (0x208)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																RXIPV4GFIS															
N/A																RW															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0

MMC_IPC_INTR_RX [31:0]

Bits	Name	Description
0	RXIPV4GFIS	MMC Receive IPV4 Good Frame Counter Interrupt Status This bit is set when the rxipv4_gd_frms counter reaches half of the maximum value or the maximum value.

MMC_IPC_INTR_RX Bits

55.7.35 RXIPV4_GD_FMS (0x210)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FRMCNT																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RXIPV4_GD_FMS [31:0]

Bits	Name	Description
31-0	FRMCNT	Number of good IPv4 datagrams received with the TCP, UDP, or ICMP payload

RXIPV4_GD_FMS Bits

55.7.36 L3_L4_CFG[L3_L4_CTRL] (0x400 + 0x20 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD										L4DPIM0	L4DPM0	L4SPIM0	L4SPM0	RSVD	L4PEN0	L3HDBM0					L3HSBM0					L3DAIM0	L3DAM0	L3SAIM0	L3SAM0	RSVD	L3PEN0
N/A										RW	RW	RW	RW	N/A	RW	RW					RW					RW	RW	RW	RW	N/A	RW
x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	0

L3_L4_CFG[L3_L4_CTRL] [31:0]

Bits	Name	Description
21	L4DPIM0	Layer 4 Destination Port Inverse Match Enable When set, this bit indicates that the Layer 4 Destination Port number field is enabled for inverse matching. When reset, this bit indicates that the Layer 4 Destination Port number field is enabled for perfect matching. This bit is valid and applicable only when Bit 20 (L4DPM0) is set high.

Bits	Name	Description
20	L4DPM0	Layer 4 Destination Port Match Enable When set, this bit indicates that the Layer 4 Destination Port number field is enabled for matching. When reset, the MAC ignores the Layer 4 Destination Port number field for matching.
19	L4SPIM0	Layer 4 Source Port Inverse Match Enable When set, this bit indicates that the Layer 4 Source Port number field is enabled for inverse matching. When reset, this bit indicates that the Layer 4 Source Port number field is enabled for perfect matching. This bit is valid and applicable only when Bit 18 (L4SPM0) is set high.
18	L4SPM0	Layer 4 Source Port Match Enable When set, this bit indicates that the Layer 4 Source Port number field is enabled for matching. When reset, the MAC ignores the Layer 4 Source Port number field for matching.
16	L4PEN0	Layer 4 Protocol Enable When set, this bit indicates that the Source and Destination Port number fields for UDP frames are used for matching. When reset, this bit indicates that the Source and Destination Port number fields for TCP frames are used for matching. The Layer 4 matching is done only when either L4SPM0 or L4DPM0 bit is set high.
15-11	L3HDBM0	Layer 3 IP DA Higher Bits Match IPv4 Frames: This field contains the number of higher bits of IP Destination Address that are matched in the IPv4 frames. The following list describes the values of this field: - 0: No bits are masked. - 1: LSb[0] is masked. - 2: Two LSbs [1:0] are masked. - ... - 31: All bits except MSb are masked. IPv6 Frames: Bits [12:11] of this field correspond to Bits [6:5] of L3HSBM0, which indicate the number of lower bits of IP Source or Destination Address that are masked in the IPv6 frames. The following list describes the concatenated values of the L3HDBM0[1:0] and L3HSBM0 bits: - 0: No bits are masked. - 1: LSb[0] is masked. - 2: Two LSbs [1:0] are masked. - ... - 127: All bits except MSb are masked. This field is valid and applicable only if L3DAM0 or L3SAM0 is set high.

Bits	Name	Description
10-6	L3HSBM0	Layer 3 IP SA Higher Bits Match IPv4 Frames: This field contains the number of lower bits of IP Source Address that are masked for matching in the IPv4 frames. The following list describes the values of this field: - 0: No bits are masked. - 1: LSb[0] is masked. - 2: Two LSbs [1:0] are masked. - ... - 31: All bits except MSb are masked. IPv6 Frames: This field contains Bits [4:0] of the field that indicates the number of higher bits of IP Source or Destination Address matched in the IPv6 frames. This field is valid and applicable only if L3DAM0 or L3SAM0 is set high.
5	L3DAIM0	Layer 3 IP DA Inverse Match Enable When set, this bit indicates that the Layer 3 IP Destination Address field is enabled for inverse matching. When reset, this bit indicates that the Layer 3 IP Destination Address field is enabled for perfect matching. This bit is valid and applicable only when Bit 4 (L3DAM0) is set high.
4	L3DAM0	Layer 3 IP DA Match Enable When set, this bit indicates that Layer 3 IP Destination Address field is enabled for matching. When reset, the MAC ignores the Layer 3 IP Destination Address field for matching. Note: When Bit 0 (L3PEN0) is set, you should set either this bit or Bit 2 (L3SAM0) because either IPv6 DA or SA can be checked for filtering.
3	L3SAIM0	Layer 3 IP SA Inverse Match Enable When set, this bit indicates that the Layer 3 IP Source Address field is enabled for inverse matching. When reset, this bit indicates that the Layer 3 IP Source Address field is enabled for perfect matching. This bit is valid and applicable only when Bit 2 (L3SAM0) is set high.
2	L3SAM0	Layer 3 IP SA Match Enable When set, this bit indicates that the Layer 3 IP Source Address field is enabled for matching. When reset, the MAC ignores the Layer 3 IP Source Address field for matching.

Bits	Name	Description
0	L3PEN0	Layer 3 Protocol Enable When set, this bit indicates that the Layer 3 IP Source or Destination Address matching is enabled for the IPv6 frames. When reset, this bit indicates that the Layer 3 IP Source or Destination Address matching is enabled for the IPv4 frames. The Layer 3 matching is done only when either L3SAM0 or L3DAM0 bit is set high.

L3_L4_CFG[L3_L4_CTRL] Bits

55.7.37 L3_L4_CFG[L4_ADDR] (0x404 + 0x20 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L4DP0																L4SP0															
RW																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

L3_L4_CFG[L4_ADDR] [31:0]

Bits	Name	Description
31-16	L4DP0	Layer 4 Destination Port Number Field When Bit 16 (L4PEN0) is reset and Bit 20 (L4DPM0) is set in Register 256 (Layer 3 and Layer 4 Control Register 0), this field contains the value to be matched with the TCP Destination Port Number field in the IPv4 or IPv6 frames. When Bit 16 (L4PEN0) and Bit 20 (L4DPM0) are set in Register 256 (Layer 3 and Layer 4 Control Register 0), this field contains the value to be matched with the UDP Destination Port Number field in the IPv4 or IPv6 frames.
15-0	L4SP0	Layer 4 Source Port Number Field When Bit 16 (L4PEN0) is reset and Bit 20 (L4DPM0) is set in Register 256 (Layer 3 and Layer 4 Control Register 0), this field contains the value to be matched with the TCP Source Port Number field in the IPv4 or IPv6 frames. When Bit 16 (L4PEN0) and Bit 20 (L4DPM0) are set in Register 256 (Layer 3 and Layer 4 Control Register 0), this field contains the value to be matched with the UDP Source Port Number field in the IPv4 or IPv6 frames.

L3_L4_CFG[L4_ADDR] Bits

55.7.38 L3_L4_CFG[L3_ADDR_0] (0x410 + 0x20 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L3A00																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

L3_L4_CFG[L3_ADDR_0] [31:0]

Bits	Name	Description
31-0	L3A00	Layer 3 Address 0 Field When Bit 0 (L3PEN0) and Bit 2 (L3SAM0) are set in Register 256 (Layer 3 and Layer 4 Control Register 0), this field contains the value to be matched with Bits [31:0] of the IP Source Address field in the IPv6 frames. When Bit 0 (L3PEN0) and Bit 4 (L3DAM0) are set in Register 256 (Layer 3 and Layer 4 Control Register 0), this field contains the value to be matched with Bits [31:0] of the IP Destination Address field in the IPv6 frames. When Bit 0 (L3PEN0) is reset and Bit 2 (L3SAM0) is set in Register 256 (Layer 3 and Layer 4 Control Register 0), this field contains the value to be matched with the IP Source Address field in the IPv4 frames.

L3_L4_CFG[L3_ADDR_0] Bits

55.7.39 L3_L4_CFG[L3_ADDR_1] (0x414 + 0x20 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L3A10																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

L3_L4_CFG[L3_ADDR_1] [31:0]

Bits	Name	Description
31-0	L3A10	Layer 3 Address 1 Field When Bit 0 (L3PEN0) and Bit 2 (L3SAM0) are set in Register 256 (Layer 3 and Layer 4 Control Register 0), this field contains the value to be matched with Bits [63:32] of the IP Source Address field in the IPv6 frames. When Bit 0 (L3PEN0) and Bit 4 (L3DAM0) are set in Register 256 (Layer 3 and Layer 4 Control Register 0), this field contains the value to be matched with Bits [63:32] of the IP Destination Address field in the IPv6 frames. When Bit 0 (L3PEN0) is reset and Bit 4 (L3DAM0) is set in Register 256 (Layer 3 and Layer 4 Control Register 0), this field contains the value to be matched with the IP Destination Address field in the IPv4 frames.

L3_L4_CFG[L3_ADDR_1] Bits

55.7.40 L3_L4_CFG[L3_ADDR_2] (0x418 + 0x20 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L3A20																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

L3_L4_CFG[L3_ADDR_2] [31:0]

Bits	Name	Description
31-0	L3A20	Layer 3 Address 2 Field When Bit 0 (L3PEN0) and Bit 2 (L3SAM0) are set in Register 256 (Layer 3 and Layer 4 Control Register 0), this field contains the value to be matched with Bits [95:64] of the IP Source Address field in the IPv6 frames. When Bit 0 (L3PEN0) and Bit 4 (L3DAM0) are set in Register 256 (Layer 3 and Layer 4 Control Register 0), this field contains value to be matched with Bits [95:64] of the IP Destination Address field in the IPv6 frames. When Bit 0 (L3PEN0) is reset in Register 256 (Layer 3 and Layer 4 Control Register 0), this register is not used.

L3_L4_CFG[L3_ADDR_2] Bits

55.7.41 L3_L4_CFG[L3_ADDR_3] (0x41C + 0x20 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L3A30																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

L3_L4_CFG[L3_ADDR_3] [31:0]

Bits	Name	Description
31-0	L3A30	Layer 3 Address 3 Field When Bit 0 (L3PEN0) and Bit 2 (L3SAM0) are set in Register 256 (Layer 3 and Layer 4 Control Register 0), this field contains the value to be matched with Bits [127:96] of the IP Source Address field in the IPv6 frames. When Bit 0 (L3PEN0) and Bit 4 (L3DAM0) are set in Register 256 (Layer 3 and Layer 4 Control Register 0), this field contains the value to be matched with Bits [127:96] of the IP Destination Address field in the IPv6 frames. When Bit 0 (L3PEN0) is reset in Register 256 (Layer 3 and Layer 4 Control Register 0), this register is not used.

L3_L4_CFG[L3_ADDR_3] Bits

55.7.42 VLAN_TAG_INC_RPL (0x584)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD												CSVL	VLP	VLC	VLT																
N/A												RW	RW	RW	RW																
x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

VLAN_TAG_INC_RPL [31:0]

Bits	Name	Description
19	CSVL	C-VLAN or S-VLAN When this bit is set, S-VLAN type (0x88A8) is inserted or replaced in the 13th and 14th bytes of transmitted frames. When this bit is reset, C-VLAN type (0x8100) is inserted or replaced in the transmitted frames.
18	VLP	VLAN Priority Control When this bit is set, the control Bits [17:16] are used for VLAN deletion, insertion, or replacement. When this bit is reset, the mti_vlan_ctrl_i control input is used, and Bits [17:16] are ignored.

Bits	Name	Description
17-16	VLC	VLAN Tag Control in Transmit Frames - 2' b00: No VLAN tag deletion, insertion, or replacement - 2' b01: VLAN tag deletion The MAC removes the VLAN type (bytes 13 and 14) and VLAN tag (bytes 15 and 16) of all transmitted frames with VLAN tags. - 2' b10: VLAN tag insertion The MAC inserts VLT in bytes 15 and 16 of the frame after inserting the Type value (0x8100/0x88a8) in bytes 13 and 14. This operation is performed on all transmitted frames, irrespective of whether they already have a VLAN tag. - 2' b11: VLAN tag replacement The MAC replaces VLT in bytes 15 and 16 of all VLAN-type transmitted frames (Bytes 13 and 14 are 0x8100/0x88a8). Note: Changes to this field take effect only on the start of a frame. If you write this register field when a frame is being transmitted, only the subsequent frame can use the updated value, that is, the current frame does not use the updated value.
15-0	VLT	VLAN Tag for Transmit Frames This field contains the value of the VLAN tag to be inserted or replaced. The value must only be changed when the transmit lines are inactive or during the initialization phase. Bits[15:13] are the User Priority, Bit 12 is the CFI/DEI, and Bits[11:0] are the VLAN tag's VID field.

VLAN_TAG_INC_RPL Bits

55.7.43 VLAN_HASH (0x588)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																VLHT															
N/A																RW															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VLAN_HASH [31:0]

Bits	Name	Description
15-0	VLHT	VLAN Hash Table This field contains the 16-bit VLAN Hash Table.

VLAN_HASH Bits

55.7.44 TS_CTRL (0x700)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD			ATSEN3	ATSEN2	ATSEN1	ATSEN0	ATSFC	RSVD					TSENMADDR	SNAPTYPSEL		TSMSTRENA	TSEVNTENA	TSIPV4ENA	TSIPV6ENA	TSIPENA	TSVER2ENA	TSCTRLSSR	TSENALL	RSVD		TSADDRREG	TSTRIG	TSUPDT	TSINIT	TSCFUPDT	TSENA
N/A			RW	RW	RW	RW	RW	N/A					RW	RW		RW	RW	RW	RW	RW	RW	RW	RW	N/A		RW	RW	RW	RW	RW	RW
x	x	x	0	0	0	0	0	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	x	x	0	0	0	0	0	0

TS_CTRL [31:0]

Bits	Name	Description
28	ATSEN3	Auxiliary Snapshot 3 Enable This field controls capturing the Auxiliary Snapshot Trigger 3. When this bit is set, the Auxiliary snapshot of event on ptp_aux_trig_i[3] input is enabled. When this bit is reset, the events on this input are ignored. This bit is reserved when the Add IEEE 1588 Auxiliary Snapshot option is not selected during core configuration or the selected number in the Number of IEEE 1588 Auxiliary Snapshot Inputs option is less than four.
27	ATSEN2	Auxiliary Snapshot 2 Enable This field controls capturing the Auxiliary Snapshot Trigger 2. When this bit is set, the Auxiliary snapshot of event on ptp_aux_trig_i[2] input is enabled. When this bit is reset, the events on this input are ignored. This bit is reserved when the Add IEEE 1588 Auxiliary Snapshot option is not selected during core configuration or the selected number in the Number of IEEE 1588 Auxiliary Snapshot Inputs option is less than three.
26	ATSEN1	Auxiliary Snapshot 1 Enable This field controls capturing the Auxiliary Snapshot Trigger 1. When this bit is set, the Auxiliary snapshot of event on ptp_aux_trig_i[1] input is enabled. When this bit is reset, the events on this input are ignored. This bit is reserved when the Add IEEE 1588 Auxiliary Snapshot option is not selected during core configuration or the selected number in the Number of IEEE 1588 Auxiliary Snapshot Inputs option is less than two.

Bits	Name	Description
25	ATSEN0	Auxiliary Snapshot 0 Enable This field controls capturing the Auxiliary Snapshot Trigger 0. When this bit is set, the Auxiliary snapshot of event on ptp_aux_trig_i[0] input is enabled. When this bit is reset, the events on this input are ignored.
24	ATSFC	Auxiliary Snapshot FIFO Clear When set, it resets the pointers of the Auxiliary Snapshot FIFO. This bit is cleared when the pointers are reset and the FIFO is empty. When this bit is high, auxiliary snapshots get stored in the FIFO. This bit is reserved when the Add IEEE 1588 Auxiliary Snapshot option is not selected during core configuration.
18	TSENMACADDR	Enable MAC address for PTP Frame Filtering When set, the DA MAC address (that matches any MAC Address register) is used to filter the PTP frames when PTP is directly sent over Ethernet.
17-16	SNAPTYPSEL	Select PTP packets for Taking Snapshots These bits along with Bits 15 and 14 decide the set of PTP packet types for which snapshot needs to be taken.
15	TSMSTRENA	Enable Snapshot for Messages Relevant to Master When set, the snapshot is taken only for the messages relevant to the master node. Otherwise, the snapshot is taken for the messages relevant to the slave node.
14	TSEVNTENA	Enable Timestamp Snapshot for Event Messages When set, the timestamp snapshot is taken only for event messages (SYNC, Delay_Req, Pdelay_Req, or Pdelay_Resp). When reset, the snapshot is taken for all messages except Announce, Management, and Signaling.
13	TSIPV4ENA	Enable Processing of PTP Frames Sent over IPv4-UDP When set, the MAC receiver processes the PTP packets encapsulated in UDP over IPv4 packets. When this bit is clear, the MAC ignores the PTP transported over UDP-IPv4 packets. This bit is set by default.
12	TSIPV6ENA	Enable Processing of PTP Frames Sent over IPv6-UDP When set, the MAC receiver processes PTP packets encapsulated in UDP over IPv6 packets. When this bit is clear, the MAC ignores the PTP transported over UDP-IPv6 packets.
11	TSIPENA	Enable Processing of PTP over Ethernet Frames When set, the MAC receiver processes the PTP packets encapsulated directly in the Ethernet frames. When this bit is clear, the MAC ignores the PTP over Ethernet packets

Bits	Name	Description
10	TSVER2ENA	Enable PTP packet Processing for Version 2 Format When set, the PTP packets are processed using the 1588 version 2 format. Otherwise, the PTP packets are processed using the version 1 format.
9	TSCTRLSSR	Timestamp Digital or Binary Rollover Control When set, the Timestamp Low register rolls over after 0x3B9A_C9FF value (that is, 1 nanosecond accuracy) and increments the timestamp (High) seconds. When reset, the rollover value of sub-second register is 0x7FFF_FFFF. The sub-second increment has to be programmed correctly depending on the PTP reference clock frequency and the value of this bit.
8	TSENALL	Enable Timestamp for All Frames When set, the timestamp snapshot is enabled for all frames received by the MAC.
5	TSADDRREG	Addend Reg Update When set, the content of the Timestamp Addend register is updated in the PTP block for fine correction. This is cleared when the update is completed. This register bit should be zero before setting it.
4	TSTRIG	Timestamp Interrupt Trigger Enable When set, the timestamp interrupt is generated when the System Time becomes greater than the value written in the Target Time register. This bit is reset after the generation of the Timestamp Trigger Interrupt.
3	TSUPDT	Timestamp Update When set, the system time is updated (added or subtracted) with the value specified in Register 452 (System Time -Seconds Update Register) and Register 453 (System Time -Nanoseconds Update Register). This bit should be read zero before updating it. This bit is reset when the update is completed in hardware. The “Timestamp Higher Word” register (if enabled during core configuration) is not updated.

Bits	Name	Description
2	TSINIT	Timestamp Initialize When set, the system time is initialized (overwritten) with the value specified in the Register 452 (System Time –Seconds Update Register) and Register 453 (System Time –Nanoseconds Update Register). This bit should be read zero before updating it. This bit is reset when the initialization is complete. The “Timestamp Higher Word” register (if enabled during core configuration) can only be initialized.
1	TSCFUPDT	Timestamp Fine or Coarse Update When set, this bit indicates that the system times update should be done using the fine update method. When reset, it indicates the system timestamp update should be done using the Coarse method.
0	TSENA	Timestamp Enable When set, the timestamp is added for the transmit and receive frames. When disabled, timestamp is not added for the transmit and receive frames and the Timestamp Generator is also suspended. You need to initialize the Timestamp (system time) after enabling this mode. On the receive side, the MAC processes the 1588 frames only if this bit is set.

TS_CTRL Bits

55.7.45 SUB_SEC_INCR (0x704)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								SSINC							
N/A																								RW							
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	

SUB_SEC_INCR [31:0]

Bits	Name	Description
7-0	SSINC	Sub-second Increment Value The value programmed in this field is accumulated every clock cycle (of clk_ptp_i) with the contents of the sub-second register. For example, when PTP clock is 50 MHz (period is 20 ns), you should program 20 (0x14) when the System Time- Nanoseconds register has an accuracy of 1 ns [Bit 9 (TSCTRLSSR) is set in Register 448 (Timestamp Control Register)]. When TSCTRLSSR is clear, the Nanoseconds register has a resolution of ~0.465ns. In this case, you should program a value of 43 (0x2B) that is derived by 20ns/0.465.

SUB_SEC_INCR Bits

55.7.46 SYST_SEC (0x708)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS																															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SYST_SEC [31:0]

Bits	Name	Description
31-0	TSS	Timestamp Second The value in this field indicates the current value in seconds of the System Time maintained by the MAC.

SYST_SEC Bits

55.7.47 SYST_NSEC (0x70C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	SS																														
N/A	RO																														
x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SYST_NSEC [31:0]

Bits	Name	Description
30-0	TSSS	Timestamp Sub Seconds The value in this field has the sub second representation of time, with an accuracy of 0.46 ns. When Bit 9 (TSCTRLSSR) is set in Register 448 (Timestamp Control Register), each bit represents 1 ns and the maximum value is 0x3B9A_C9FF, after which it rolls-over to zero.

SYST_NSEC Bits

55.7.48 SYST_SEC_UPD (0x710)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSS																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SYST_SEC_UPD [31:0]

Bits	Name	Description
31-0	TSS	Timestamp Second The value in this field indicates the time in seconds to be initialized or added to the system time.

SYST_SEC_UPD Bits

55.7.49 SYST_NSEC_UPD (0x714)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDSUB	TSS																														
RW	RW																														
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SYST_NSEC_UPD [31:0]

Bits	Name	Description
31	ADDSUB	Add or Subtract Time When this bit is set, the time value is subtracted with the contents of the update register. When this bit is reset, the time value is added with the contents of the update register.

Bits	Name	Description
30-0	TSSS	Timestamp Sub Seconds The value in this field has the sub second representation of time, with an accuracy of 0.46 ns. When Bit 9 (TSCTRLSSR) is set in Register 448 (Timestamp Control Register), each bit represents 1 ns and the programmed value should not exceed 0x3B9A_C9FF.

SYST_NSEC_UPD Bits

55.7.50 TS_ADDEND (0x718)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSAR																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TS_ADDEND [31:0]

Bits	Name	Description
31-0	TSAR	Timestamp Addend Register This field indicates the 32-bit time value to be added to the Accumulator register to achieve time synchronization.

TS_ADDEND Bits

55.7.51 TGTTM_SEC (0x71C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSTR																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TGTTM_SEC [31:0]

Bits	Name	Description
31-0	TSTR	Target Time Seconds Register This register stores the time in seconds. When the timestamp value matches or exceeds both Target Timestamp registers, then based on Bits [6:5] of Register 459 (PPS Control Register), the MAC starts or stops the PPS signal output and generates an interrupt (if enabled).

TGTTM_SEC Bits

55.7.52 TGTTM_NSEC (0x720)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRG TBUSY	TTSLO																														
RW	RW																														
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

TGTTM_NSEC [31:0]

Bits	Name	Description
31	TRGTBUSY	Target Time Register Busy The MAC sets this bit when the PPSCMD field (Bit [3:0]) in Register 459 (PPS Control Register) is programmed to 010 or 011. Programming the PPSCMD field to 010 or 011, instructs the MAC to synchronize the Target Time Registers to the PTP clock domain. The MAC clears this bit after synchronizing the Target Time Registers to the PTP clock domain The application must not update the Target Time Registers when this bit is read as 1. Otherwise, the synchronization of the previous programmed time gets corrupted. This bit is reserved when the Enable Flexible Pulse-Per-Second Output feature is not selected.

Bits	Name	Description
30-0	TTSLO	Target Timestamp Low Register This register stores the time in (signed) nanoseconds. When the value of the timestamp matches the both Target Timestamp registers, then based on the TRGTMODSEL0 field (Bits [6:5]) in Register 459 (PPS Control Register), the MAC starts or stops the PPS signal output and generates an interrupt (if enabled). This value should not exceed 0x3B9A_C9FF when Bit 9 (TSCTRLSSR) is set in Register 448 (Timestamp Control Register). The actual start or stop time of the PPS signal output may have an error margin up to one unit of sub-second increment value.

TGTTM_NSEC Bits

55.7.53 SYSTM_H_SEC (0x724)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																TSHWR															
N/A																RW															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SYSTM_H_SEC [31:0]

Bits	Name	Description
15-0	TSHWR	Timestamp Higher Word Register This field contains the most significant 16-bits of the timestamp seconds value. This register is optional and can be selected using the Enable IEEE 1588 Higher Word Register option during core configuration. The register is directly written to initialize the value. This register is incremented when there is an overflow from the 32-bits of the System Time - Seconds register.

SYSTM_H_SEC Bits

55.7.54 TS_STATUS (0x728)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD				ATSNS			ATSSTM								ATSSTN							TSTRGTERR3	TSTARGET3	TSTRGTERR2	TSTARGET2	TSTRGTERR1	TSTARGET1	TSTRGTERR	AUXTSTRIG	TSTARGET	TSSOVF
N/A				RO			RO								RO							RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
x	x	0	0	0	0	0	0	x	x	x	x	0	0	0	0	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0

TS_STATUS [31:0]

Bits	Name	Description
29-25	ATSNS	Number of Auxiliary Timestamp Snapshots This field indicates the number of Snapshots available in the FIFO. A value equal to the selected depth of FIFO (4, 8, or 16) indicates that the Auxiliary Snapshot FIFO is full. These bits are cleared (to 00000) when the Auxiliary snapshot FIFO clear bit is set. This bit is valid only if the Add IEEE 1588 Auxiliary Snapshot option is selected during core configuration.
24	ATSSTM	Auxiliary Timestamp Snapshot Trigger Missed This bit is set when the Auxiliary timestamp snapshot FIFO is full and external trigger was set. This indicates that the latest snapshot is not stored in the FIFO. This bit is valid only if the Add IEEE 1588 Auxiliary Snapshot option is selected during core configuration.
19-16	ATSSTN	Auxiliary Timestamp Snapshot Trigger Identifier These bits identify the Auxiliary trigger inputs for which the timestamp available in the Auxiliary Snapshot Register is applicable. When more than one bit is set at the same time, it means that corresponding auxiliary triggers were sampled at the same clock. These bits are applicable only if the number of Auxiliary snapshots is more than one. One bit is assigned for each trigger as shown in the following list: - Bit 16: Auxiliary trigger 0 - Bit 17: Auxiliary trigger 1 - Bit 18: Auxiliary trigger 2 - Bit 19: Auxiliary trigger 3 The software can read this register to find the triggers that are set when the timestamp is taken.

Bits	Name	Description
9	TSTRGTERR3	Timestamp Target Time Error This bit is set when the target time, being programmed in Register 496 and Register 497, is already elapsed. This bit is cleared when read by the application.
8	TSTARGET3	Timestamp Target Time Reached for Target Time PPS3 When set, this bit indicates that the value of system time is greater than or equal to the value specified in Register 496 (PPS3 Target Time High Register) and Register 497 (PPS3 Target Time Low Register).
7	TSTRGTERR2	
6	TSTARGET2	
5	TSTRGTERR1	
4	TSTARGET1	
3	TSTRGTERR	
2	AUXTSTRIG	
1	TSTARGET	
0	TSSOVF	

TS_STATUS Bits

55.7.55 PPS_CTRL (0x72C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	TRGTMODSEL3		RSVD		PPSCMD3			RSVD	TRGTMODSEL2		RSVD		PPSCMD2		RSVD	TRGTMODSEL1		RSVD		PPSCMD1		RSVD	TRGTMODSEL0	PPSEN0		PPSCTRLCMD0					
N/A	RW		N/A		WO			N/A	RW		N/A		WO		N/A	RW		N/A		WO		N/A	RW	RW		RWWO					
x	0	0	x	x	0	0	0	x	0	0	x	x	0	0	0	x	0	0	x	x	0	0	0	x	0	0	0	0	0	0	0

PPS_CTRL [31:0]

Bits	Name	Description
30-29	TRGTMODSEL3	Target Time Register Mode for PPS3 Output This field indicates the Target Time registers (register 496 and 497) mode for PPS3 output signal. This field is similar to the TRGTMODSEL0 field.
26-24	PPSCMD3	Flexible PPS3 Output Control This field controls the flexible PPS3 output (ptp_pps_o[3]) signal. This field is similar to PPSCMD0[2:0] in functionality.

Bits	Name	Description
22-21	TRGTMODSEL2	Target Time Register Mode for PPS2 Output This field indicates the Target Time registers (register 488 and 489) mode for PPS2 output signal. This field is similar to the TRGTMODSEL0 field.
18-16	PPSCMD2	Flexible PPS2 Output Control This field controls the flexible PPS2 output (ptp_pps_o[2]) signal. This field is similar to PPSCMD0[2:0] in functionality.
14-13	TRGTMODSEL1	Target Time Register Mode for PPS1 Output This field indicates the Target Time registers (register 480 and 481) mode for PPS1 output signal. This field is similar to the TRGTMODSEL0 field.
10-8	PPSCMD1	Flexible PPS1 Output Control This field controls the flexible PPS1 output (ptp_pps_o[1]) signal. This field is similar to PPSCMD0[2:0] in functionality.
6-5	TRGTMODSEL0	Target Time Register Mode for PPS0 Output This field indicates the Target Time registers (register 455 and 456) mode for PPS0 output signal: - 00: Indicates that the Target Time registers are programmed only for generating the interrupt event. - 01: Reserved - 10: Indicates that the Target Time registers are programmed for generating the interrupt event and starting or stopping the generation of the PPS0 output signal. - 11: Indicates that the Target Time registers are programmed only for starting or stopping the generation of the PPS0 output signal. No interrupt is asserted.
4	PPSEN0	Flexible PPS Output Mode Enable When set low, Bits [3:0] function as PPSCTRL (backward compatible). When set high, Bits[3:0] function as PPSCMD.

Bits	Name	Description
3-0	PPSCTRLCMD0	PPSCTRL0: PPS0 Output Frequency Control This field controls the frequency of the PPS0 output (ptp_pps_o[0]) signal. The default value of PPSCTRL is 0000, and the PPS output is 1 pulse (of width clk_ptp_i) every second. For other values of PPSCTRL, the PPS output becomes a generated clock of following frequencies: - 0001: The binary rollover is 2 Hz, and the digital rollover is 1 Hz. - 0010: The binary rollover is 4 Hz, and the digital rollover is 2 Hz. - 0011: The binary rollover is 8 Hz, and the digital rollover is 4 Hz. - 0100: The binary rollover is 16 Hz, and the digital rollover is 8 Hz. - ... - 1111: The binary rollover is 32.768 KHz, and the digital rollover is 16.384 KHz. Note: In the binary rollover mode, the PPS output (ptp_pps_o) has a duty cycle of 50 percent with these frequencies. In the digital rollover mode, the PPS output frequency is an average number. The actual clock is of different frequency that gets synchronized every second. For example: - When PPSCTRL = 0001, the PPS (1 Hz) has a low period of 537 ms and a high period of 463 ms - When PPSCTRL = 0010, the PPS (2 Hz) is a sequence of: - One clock of 50 percent duty cycle and 537 ms period - Second clock of 463 ms period (268 ms low and 195 ms high) - When PPSCTRL = 0011, the PPS (4 Hz) is a sequence of: - Three clocks of 50 percent duty cycle and 268 ms period - Fourth clock of 195 ms period (134 ms low and 61 ms high) PPSCMD0: Flexible PPS0 Output Control Programming these bits with a non-zero value instructs the MAC to initiate an event. When the command is transferred or synchronized to the PTP clock domain, these bits get cleared automatically. The Software should ensure that these bits are programmed only when they are “all-zero” . 0000: No Command 0001: START Single Pulse This command generates single pulse rising at the start point defined in Target Time Registers and of a duration defined in the PPS0 Width Register. 0010: START Pulse Train This command generates the train of pulses rising at the start point defined in the Target Time Registers and of a duration defined in the PPS0 Width Register and repeated at interval defined in the PPS Interval Register. By default, the PPS pulse train is free-running unless stopped by ‘STOP Pulse train at time’ or ‘STOP Pulse Train immediately’ commands. 0011: Cancel START This command cancels the START Single Pulse and START Pulse Train commands if the system time has not crossed the programmed start time. 0100: STOP Pulse train at time This command stops the train of pulses initiated by the START Pulse Train command (PPSCMD = 0010) after the time programmed in the Target Time registers elapses. 0101: STOP Pulse Train immediately This command immediately stops the train of pulses initiated by the START Pulse Train command (PPSCMD = 0010). 0110: Cancel STOP Pulse train This command cancels the STOP pulse train at time command if the programmed stop time has not elapsed. The PPS pulse train becomes free-running on the successful execution of this command. 0111-1111: Reserved Note: These bits get cleared automatically

Bits	Name	Description

PPS_CTRL Bits

55.7.56 AUX_TS_NSEC (0x730)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	AUXTSLO																														
N/A	RO																														
x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

AUX_TS_NSEC [31:0]

Bits	Name	Description
30-0	AUXTSLO	Contains the lower 31 bits (nano-seconds field) of the auxiliary timestamp.

AUX_TS_NSEC Bits

55.7.57 AUX_TS_SEC (0x734)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AUXTSHI																															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

AUX_TS_SEC [31:0]

Bits	Name	Description
31-0	AUXTSHI	Contains the lower 32 bits of the Seconds field of the auxiliary timestamp. Note: The top of the FIFO is removed only when the last byte of AUX_TS_SEC is read.

AUX_TS_SEC Bits

55.7.58 PPS0_INTERVAL (0x760)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PPSINT																															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PPS0_INTERVAL [31:0]

Bits	Name	Description
31-0	PPSINT	PPS0 Output Signal Interval These bits store the interval between the rising edges of PPS0 signal output in terms of units of sub-second increment value. You need to program one value less than the required interval. For example, if the PTP reference clock is 50 MHz (period of 20ns), and desired interval between rising edges of PPS0 signal output is 100ns (that is, five units of sub-second increment value), then you should program value 4 (5 -1) in this register.

PPS0_INTERVAL Bits

55.7.59 PPS0_WIDTH (0x764)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PPSWIDTH																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PPS0_WIDTH [31:0]

Bits	Name	Description
31-0	PPSWIDTH	PPS0 Output Signal Width These bits store the width between the rising edge and corresponding falling edge of the PPS0 signal output in terms of units of sub-second increment value. You need to program one value less than the required interval. For example, if PTP reference clock is 50 MHz (period of 20ns), and desired width between the rising and corresponding falling edges of PPS0 signal output is 80ns (that is, four units of sub-second increment value), then you should program value 3 (4 -1) in this register.

PPS0_WIDTH Bits

55.7.60 PPS[TGTTM_SEC] (0x780 + 0x20 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSTRH1																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

PPS[TGTTM_SEC] [31:0]

Bits	Name	Description
31-0	TSTRH1	PPS1 Target Time Seconds Register This register stores the time in seconds. When the timestamp value matches or exceeds both Target Timestamp registers, then based on Bits [14:13], TRGTMODSEL1, of Register 459 (PPS Control Register), the MAC starts or stops the PPS signal output and generates an interrupt (if enabled).

PPS[TGTTM_SEC] Bits

55.7.61 PPS[TGTTM_NSEC] (0x784 + 0x20 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRGTBUSY1	TTSL1																														
RW	RW																														
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

PPS[TGTTM_NSEC] [31:0]

Bits	Name	Description
31	TRGTBUSY1	PPS1 Target Time Register Busy The MAC sets this bit when the PPSCMD1 field (Bits [10:8]) in Register 459 (PPS Control Register) is programmed to 010 or 011. Programming the PPSCMD1 field to 010 or 011 instructs the MAC to synchronize the Target Time Registers to the PTP clock domain. The MAC clears this bit after synchronizing the Target Time Registers to the PTP clock domain The application must not update the Targeers wht Time Register this bit is read as 1. Otherwise, the synchronization of the previous programmed time gets corrupted.
30-0	TTSL1	Target Time Low for PPS1 Register This register stores the time in (signed) nanoseconds. When the value of the timestamp matches the both Target Timestamp registers, then based on the TRGTMODSEL1 field (Bits [14:13]) in Register 459 (PPS Control Register), the MAC starts or stops the PPS signal output and generates an interrupt (if enabled). This value should not exceed 0x3B9A_C9FF when Bit 9 (TSCTRLSSR) is set in Register 448 (Timestamp Control Register). The actual start or stop time of the PPS signal output may have an error margin up to one unit of sub-second increment value.

PPS[TGTTM_NSEC] Bits

55.7.62 PPS[INTERVAL] (0x788 + 0x20 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PPSINT																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PPS[INTERVAL] [31:0]

Bits	Name	Description
31-0	PPSINT	PPS1 Output Signal Interval These bits store the interval between the rising edges of PPS1 signal output in terms of units of sub-second increment value. You need to program one value less than the required interval. For example, if the PTP reference clock is 50 MHz (period of 20ns), and desired interval between rising edges of PPS1 signal output is 100ns (that is, five units of sub-second increment value), then you should program value 4 (5 - 1) in this register.

PPS[INTERVAL] Bits

55.7.63 PPS[WIDTH] (0x78C + 0x20 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PPSWIDTH																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PPS[WIDTH] [31:0]

Bits	Name	Description
31-0	PPSWIDTH	PPS1 Output Signal Width These bits store the width between the rising edge and corresponding falling edge of the PPS1 signal output in terms of units of sub-second increment value. You need to program one value less than the required interval. For example, if PTP reference clock is 50 MHz (period of 20ns), and desired width between the rising and corresponding falling edges of PPS1 signal output is 80ns (that is, four units of sub-second increment value), then you should program value 3 (4 - 1) in this register.

PPS[WIDTH] Bits

55.7.64 DMA_BUS_MODE (0x1000)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
RIB	RSVD	PRWG	TXPR	MB	AAL	PBLX8	USP	RPBL							FB	PR	PBL							ATDS	DSL							DA	SWR

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RW	N/A	RW		RW	RW	RW	RW	RW	RW						RW	RW		RW						RW	RW						RW	RW
0	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

DMA_BUS_MODE [31:0]

Bits	Name	Description
31	RIB	Rebuild INCRx Burst When this bit is set high and the AHB master gets an EBT (Retry, Split, or Losing bus grant), the AHB master interface rebuilds the pending beats of any burst transfer initiated with INCRx. The AHB master interface rebuilds the beats with a combination of specified bursts with INCRx and SINGLE. By default, the AHB master interface rebuilds pending beats of an EBT with an unspecified (INCR) burst.
29-28	PRWG	Channel Priority Weights This field sets the priority weights for Channel 0 during the round-robin arbitration between the DMA channels for the system bus. - 00: The priority weight is 1. - 01: The priority weight is 2. - 10: The priority weight is 3. - 11: The priority weight is 4. This field is present in all DWC_gmac configurations except GMAC-AXI when you select the AV feature. Otherwise, this field is reserved and read-only (RO).
27	TXPR	Transmit Priority When set, this bit indicates that the transmit DMA has higher priority than the receive DMA during arbitration for the system-side bus. In the GMAC-AXI configuration, this bit is reserved and read-only (RO).
26	MB	Mixed Burst When this bit is set high and the FB bit is low, the AHB master interface starts all bursts of length more than 16 with INCR (undefined burst), whereas it reverts to fixed burst transfers (INCRx and SINGLE) for burst length of 16 and less.

Bits	Name	Description
25	AAL	Address-Aligned Beats When this bit is set high and the FB bit is equal to 1, the AHB or AXI interface generates all bursts aligned to the start address LS bits. If the FB bit is equal to 0, the first burst (accessing the start address of data buffer) is not aligned, but subsequent bursts are aligned to the address.
24	PBLX8	PBLx8 Mode When set high, this bit multiplies the programmed PBL value (Bits [22:17] and Bits[13:8]) eight times. Therefore, the DMA transfers the data in 8, 16, 32, 64, 128, and 256 beats depending on the PBL value.
23	USP	Use Separate PBL When set high, this bit configures the Rx DMA to use the value configured in Bits [22:17] as PBL. The PBL value in Bits [13:8] is applicable only to the Tx DMA operations. When reset to low, the PBL value in Bits [13:8] is applicable for both DMA engines.
22-17	RPBL	Rx DMA PBL This field indicates the maximum number of beats to be transferred in one Rx DMA transaction. This is the maximum value that is used in a single block Read or Write. The Rx DMA always attempts to burst as specified in the RPBL bit each time it starts a Burst transfer on the host bus. You can program RPBL with values of 1, 2, 4, 8, 16, and 32. Any other value results in undefined behavior. This field is valid and applicable only when USP is set high.
16	FB	Fixed Burst This bit controls whether the AHB or AXI master interface performs fixed burst transfers or not. When set, the AHB interface uses only SINGLE, INCR4, INCR8, or INCR16 during start of the normal burst transfers. When reset, the AHB or AXI interface uses SINGLE and INCR burst transfer operations.

Bits	Name	Description
15-14	PR	Priority Ratio These bits control the priority ratio in the weighted round-robin arbitration between the Rx DMA and Tx DMA. These bits are valid only when Bit 1 (DA) is reset. The priority ratio is Rx:Tx or Tx:Rx depending on whether Bit 27 (TXPR) is reset or set. - 00: The Priority Ratio is 1:1. - 01: The Priority Ratio is 2:1. - 10: The Priority Ratio is 3:1. - 11: The Priority Ratio is 4:1.
13-8	PBL	Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA transaction. This is the maximum value that is used in a single block Read or Write. The DMA always attempts to burst as specified in PBL each time it starts a Burst transfer on the host bus. PBL can be programmed with permissible values of 1, 2, 4, 8, 16, and 32. Any other value results in undefined behavior. When USP is set high, this PBL value is applicable only for Tx DMA transactions. If the number of beats to be transferred is more than 32, then perform the following steps: 1. Set the PBLx8 mode. 2. Set the PBL.

Bits	Name	Description
7	ATDS	Alternate Descriptor Size When set, the size of the alternate descriptor (described in “Alternate or Enhanced Descriptors” on page 545) increases to 32 bytes (8 DWORDS). This is required when the Advanced Timestamp feature or the IPC Full Checksum Offload Engine (Type 2) is enabled in the receiver. The enhanced descriptor is not required if the Advanced Timestamp and IPC Full Checksum Offload Engine (Type 2) features are not enabled. In such case, you can use the 16 bytes descriptor to save 4 bytes of memory. This bit is present only when you select the Alternate Descriptor feature and any one of the following features during core configuration: - Advanced Timestamp feature - IPC Full Checksum Offload Engine (Type 2) feature Otherwise, this bit is reserved and is read-only. When reset, the descriptor size reverts back to 4 DWORDs (16 bytes).
6-2	DSL	Descriptor Skip Length This bit specifies the number of Word, Dword, or Lword (depending on the 32-bit, 64-bit, or 128-bit bus) to skip between two unchained descriptors. The address skipping starts from the end of current descriptor to the start of next descriptor. When the DSL value is equal to zero, the descriptor table is taken as contiguous by the DMA in Ring mode.
1	DA	DMA Arbitration Scheme This bit specifies the arbitration scheme between the transmit and receive paths of Channel 0. - 0: Weighted round-robin with Rx:Tx or Tx:Rx The priority between the paths is according to the priority specified in Bits [15:14] (PR) and priority weights specified in Bit 27 (TXPR). - 1: Fixed priority The transmit path has priority over receive path when Bit 27 (TXPR) is set. Otherwise, receive path has priority over the transmit path.

Bits	Name	Description
0	SWR	Software Reset When this bit is set, the MAC DMA Controller resets the logic and all internal registers of the MAC. It is cleared automatically after the reset operation is complete in all of the DWC_gmac clock domains. Before reprogramming any register of the DWC_gmac, you should read a zero (0) value in this bit. Note: - The Software reset function is driven only by this bit. Bit 0 of Register 64 (Channel 1 Bus Mode Register) or Register 128 (Channel 2 Bus Mode Register) has no impact on the Software reset function. - The reset operation is completed only when all resets in all active clock domains are de-asserted. Therefore, it is essential that all PHY inputs clocks (applicable for the selected PHY interface) are present for the software reset completion. The time to complete the software reset operation depends on the frequency of the slowest active clock.

DMA_BUS_MODE Bits

55.7.65 DMA_TX_POLL_DEMAND (0x1004)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPD																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMA_TX_POLL_DEMAND [31:0]

Bits	Name	Description
31-0	TPD	Transmit Poll Demand When these bits are written with any value, the DMA reads the current descriptor to which the Register 18 (Current Host Transmit Descriptor Register) is pointing. If that descriptor is not available (owned by the Host), the transmission returns to the Suspend state and Bit 2 (TU) of Register 5 (Status Register) is asserted. If the descriptor is available, the transmission resumes.

DMA_TX_POLL_DEMAND Bits

55.7.66 DMA_RX_POLL_DEMAND (0x1008)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RPD																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMA_RX_POLL_DEMAND [31:0]

Bits	Name	Description
31-0	RPD	Receive Poll Demand When these bits are written with any value, the DMA reads the current descriptor to which the Register 19 (Current Host Receive Descriptor Register) is pointing. If that descriptor is not available (owned by the Host), the reception returns to the Suspended state and Bit 7 (RU) of Register 5 (Status Register) is asserted. If the descriptor is available, the Rx DMA returns to the active state.

DMA_RX_POLL_DEMAND Bits

55.7.67 DMA_RX_DESC_LIST_ADDR (0x100C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDESLA																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMA_RX_DESC_LIST_ADDR [31:0]

Bits	Name	Description
31-0	RDESLA	Start of Receive List This field contains the base address of the first descriptor in the Receive Descriptor list. The LSB bits (1:0, 2:0, or 3:0) for 32-bit, 64-bit, or 128-bit bus width are ignored and internally taken as all-zero by the DMA. Therefore, these LSB bits are read-only (RO).

DMA_RX_DESC_LIST_ADDR Bits

55.7.68 DMA_TX_DESC_LIST_ADDR (0x1010)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDESLA																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMA_TX_DESC_LIST_ADDR [31:0]

Bits	Name	Description
31-0	TDESLA	Start of Transmit List This field contains the base address of the first descriptor in the Transmit Descriptor list. The LSB bits (1:0, 2:0, 3:0) for 32-bit, 64-bit, or 128-bit bus width are ignored and are internally taken as all-zero by the DMA. Therefore, these LSB bits are read-only (RO).

DMA_TX_DESC_LIST_ADDR Bits

55.7.69 DMA_STATUS (0x1014)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	GLPII	TTI	GPI	GMI	GLI	EB			TS			RS			NIS	AIS	ERI	FBI	RSVD		ETI	RWT	RPS	RU	RI	UNF	OVF	TJT	TU	TPS	TI
N/A	RO	RO	RO	RO	RO	RO			RO			RO			W1C	W1C	W1C	W1C	N/A	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x	0	0	0	0	0	0	0	0	0	0	0

DMA_STATUS [31:0]

Bits	Name	Description
30	GLPII	GLPII: GMAC LPI Interrupt (for Channel 0) This bit indicates an interrupt event in the LPI logic of the MAC. To reset this bit to 1'b0, the software must read the corresponding registers in the DWC_gmac to get the exact cause of the interrupt and clear its source. Note: GLPII status is given only in Channel 0 DMA register and is applicable only when the Energy Efficient Ethernet feature is enabled. Otherwise, this bit is reserved. When this bit is high, the interrupt signal from the MAC (sbd_intr_o) is high. -or- GTMSI: GMAC TMS Interrupt (for Channel 1 and Channel 2) This bit indicates an interrupt event in the traffic manager and scheduler logic of DWC_gmac. To reset this bit, the software must read the corresponding registers (Channel Status Register) to get the exact cause of the interrupt and clear its source. Note: GTMSI status is given only in Channel 1 and Channel 2 DMA register when the AV feature is enabled and corresponding additional transmit channels are present. Otherwise, this bit is reserved. When this bit is high, the interrupt signal from the MAC (sbd_intr_o) is high.
29	TTI	Timestamp Trigger Interrupt This bit indicates an interrupt event in the Timestamp Generator block of the DWC_gmac. The software must read the corresponding registers in the DWC_gmac to get the exact cause of the interrupt and clear its source to reset this bit to 1'b0. When this bit is high, the interrupt signal from the DWC_gmac subsystem (sbd_intr_o) is high. This bit is applicable only when the IEEE 1588 Timestamp feature is enabled. Otherwise, this bit is reserved.

Bits	Name	Description
28	GPI	GMAC PMT Interrupt This bit indicates an interrupt event in the PMT module of the DWC_gmac. The software must read the PMT Control and Status Register in the MAC to get the exact cause of interrupt and clear its source to reset this bit to 1' b0. The interrupt signal from the DWC_gmac subsystem (sbd_intr_o) is high when this bit is high. This bit is applicable only when the Power Management feature is enabled. Otherwise, this bit is reserved. Note: The GPI and pmt_intr_o interrupts are generated in different clock domains.
27	GMI	GMAC MMC Interrupt This bit reflects an interrupt event in the MMC module of the DWC_gmac. The software must read the corresponding registers in the DWC_gmac to get the exact cause of the interrupt and clear the source of interrupt to make this bit as 1' b0. The interrupt signal from the DWC_gmac subsystem (sbd_intr_o) is high when this bit is high. This bit is applicable only when the MAC Management Counters (MMC) are enabled. Otherwise, this bit is reserved.

Bits	Name	Description
26	GLI	GMAC Line Interface Interrupt When set, this bit reflects any of the following interrupt events in the DWC_gmac interfaces (if present and enabled in your configuration): - PCS (TBI, RTBI, or SGMII): Link change or auto-negotiation complete event - SMII or RGMII: Link change event - General Purpose Input Status (GPIS): Any LL or LH event on the gpi_i input ports To identify the exact cause of the interrupt, the software must first read Bit 11 and Bits[2:0] of Register 14 (Interrupt Status Register) and then to clear the source of interrupt (which also clears the GLI interrupt), read any of the following corresponding registers: - PCS (TBI, RTBI, or SGMII): Register 49 (AN Status Register) - SMII or RGMII: Register 54 (SGMII/RGMII/SMII Control and Status Register) - General Purpose Input (GPI): Register 56 (General Purpose IO Register) The interrupt signal from the DWC_gmac subsystem (sbd_intr_o) is high when this bit is high.
25-23	EB	Error Bits This field indicates the type of error that caused a Bus Error, for example, error response on the AHB or AXI interface. This field is valid only when Bit 13 (FBI) is set. This field does not generate an interrupt. - 0 0 0: Error during Rx DMA Write Data Transfer - 0 1 1: Error during Tx DMA Read Data Transfer - 1 0 0: Error during Rx DMA Descriptor Write Access - 1 0 1: Error during Tx DMA Descriptor Write Access - 1 1 0: Error during Rx DMA Descriptor Read Access - 1 1 1: Error during Tx DMA Descriptor Read Access Note: 001 and 010 are reserved.

Bits	Name	Description
22-20	TS	Transmit Process State This field indicates the Transmit DMA FSM state. This field does not generate an interrupt. - 3' b000: Stopped; Reset or Stop Transmit Command issued - 3' b001: Running; Fetching Transmit Transfer Descriptor - 3' b010: Running; Waiting for status - 3' b011: Running; Reading Data from host memory buffer and queuing it to transmit buffer (Tx FIFO) - 3' b100: TIME_STAMP write state - 3' b101: Reserved for future use - 3' b110: Suspended; Transmit Descriptor Unavailable or Transmit Buffer Underflow - 3' b111: Running; Closing Transmit Descriptor
19-17	RS	Receive Process State This field indicates the Receive DMA FSM state. This field does not generate an interrupt. - 3' b000: Stopped: Reset or Stop Receive Command issued - 3' b001: Running: Fetching Receive Transfer Descriptor - 3' b010: Reserved for future use - 3' b011: Running: Waiting for receive packet - 3' b100: Suspended: Receive Descriptor Unavailable - 3' b101: Running: Closing Receive Descriptor - 3' b110: TIME_STAMP write state - 3' b111: Running: Transferring the receive packet data from receive buffer to host memory
16	NIS	Normal Interrupt Summary Normal Interrupt Summary bit value is the logical OR of the following bits when the corresponding interrupt bits are enabled in Register 7 (Interrupt Enable Register): - Register 5[0]: Transmit Interrupt - Register 5[2]: Transmit Buffer Unavailable - Register 5[6]: Receive Interrupt - Register 5[14]: Early Receive Interrupt Only unmasked bits (interrupts for which interrupt enable is set in Register 7) affect the Normal Interrupt Summary bit. This is a sticky bit and must be cleared (by writing 1 to this bit) each time a corresponding bit, which causes NIS to be set, is cleared.

Bits	Name	Description
15	AIS	Abnormal Interrupt Summary Abnormal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in Register 7 (Interrupt Enable Register): - Register 5[1]: Transmit Process Stopped - Register 5[3]: Transmit Jabber Timeout - Register 5[4]: Receive FIFO Overflow - Register 5[5]: Transmit Underflow - Register 5[7]: Receive Buffer Unavailable - Register 5[8]: Receive Process Stopped - Register 5[9]: Receive Watchdog Timeout - Register 5[10]: Early Transmit Interrupt - Register 5[13]: Fatal Bus Error Only unmasked bits affect the Abnormal Interrupt Summary bit. This is a sticky bit and must be cleared (by writing 1 to this bit) each time a corresponding bit, which causes AIS to be set, is cleared.
14	ERI	Early Receive Interrupt This bit indicates that the DMA filled the first data buffer of the packet. This bit is cleared when the software writes 1 to this bit or Bit 6 (RI) of this register is set (whichever occurs earlier).
13	FBI	Fatal Bus Error Interrupt This bit indicates that a bus error occurred, as described in Bits [25:23]. When this bit is set, the corresponding DMA engine disables all of its bus accesses.
10	ETI	Early Transmit Interrupt This bit indicates that the frame to be transmitted is fully transferred to the MTL Transmit FIFO.
9	RWT	Receive Watchdog Timeout When set, this bit indicates that the Receive Watchdog Timer expired while receiving the current frame and the current frame is truncated after the watchdog timeout.
8	RPS	Receive Process Stopped This bit is asserted when the Receive Process enters the Stopped state.

Bits	Name	Description
7	RU	Receive Buffer Unavailable This bit indicates that the host owns the Next Descriptor in the Receive List and the DMA cannot acquire it. The Receive Process is suspended. To resume processing Receive descriptors, the host should change the ownership of the descriptor and issue a Receive Poll Demand command. If no Receive Poll Demand is issued, the Receive Process resumes when the next recognized incoming frame is received. This bit is set only when the previous Receive Descriptor is owned by the DMA.
6	RI	Receive Interrupt This bit indicates that the frame reception is complete and the specific frame status information is updated in the descriptor. The reception remains in the Running state.
5	UNF	Transmit Underflow This bit indicates that the Transmit Buffer had an Underflow during frame transmission. Transmission is suspended and an Underflow Error TDES0[1] is set.
4	OVF	Receive Overflow This bit indicates that the Receive Buffer had an Overflow during frame reception. If the partial frame is transferred to the application, the overflow status is set in RDES0[11].
3	TJT	Transmit Jabber Timeout This bit indicates that the Transmit Jabber Timer expired, which happens when the frame size exceeds 2,048 (10,240 bytes when the Jumbo frame is enabled). When the Jabber Timeout occurs, the transmission process is aborted and placed in the Stopped state. This causes the Transmit Jabber Timeout TDES0[14] flag to assert.
2	TU	Transmit Buffer Unavailable This bit indicates that the host owns the Next Descriptor in the Transmit List and the DMA cannot acquire it. Transmission is suspended. Bits[22:20] explain the Transmit Process state transitions. To resume processing Transmit descriptors, the host should change the ownership of the descriptor by setting TDES0[31] and then issue a Transmit Poll Demand command.
1	TPS	Transmit Process Stopped This bit is set when the transmission is stopped.

Bits	Name	Description
0	TI	Transmit Interrupt This bit indicates that the frame transmission is complete. When transmission is complete, Bit 31 (OWN) of TDES0 is reset, and the specific frame status information is updated in the descriptor.

DMA_STATUS Bits

55.7.70 DMA_OP_MODE (0x1018)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD		DT		RSVD		RSF	DFF	RFA_2	RFD_2	TSF	FTF	RSVD		TTC			ST		RFD		RFA		EFC	FEF	FUF	DGF	RTC		OSF	SR	RSVD
N/A		RW		N/A		RW	RW	RW	RW	RW	RW	N/A		RW			RW		RW		RW		RW	RW	RW	RW	RW		RW	RW	N/A
x	x	x	0	x	x	0	0	0	0	0	0	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x

DMA_OP_MODE [31:0]

Bits	Name	Description
28	DT	Disable Dropping of TCP/IP Checksum Error Frames When this bit is set, the MAC does not drop the frames which only have errors detected by the Receive Checksum Offload engine. Such frames do not have any errors (including FCS error) in the Ethernet frame received by the MAC but have errors only in the encapsulated payload. When this bit is reset, all error frames are dropped if the FEF bit is reset. If the IPC Full Checksum Offload Engine (Type 2) is disabled, this bit is reserved (RO with value 1'b0).
25	RSF	Receive Store and Forward When this bit is set, the MTL reads a frame from the Rx FIFO only after the complete frame has been written to it, ignoring the RTC bits. When this bit is reset, the Rx FIFO operates in the cut-through mode, subject to the threshold specified by the RTC bits.
24	DFF	Disable Flushing of Received Frames When this bit is set, the Rx DMA does not flush any frames because of the unavailability of receive descriptors or buffers as it does normally when this bit is reset. (See “Receive Process Suspended” on page 83.)

Bits	Name	Description
23	RFA_2	MSB of Threshold for Activating Flow Control If the DWC_gmac is configured for an Rx FIFO size of 8 KB or more, this bit (when set) provides additional threshold levels for activating the flow control in both half-duplex and full-duplex modes. This bit (as Most Significant Bit), along with the RFA (Bits [10:9]), gives the following thresholds for activating flow control: - 100: Full minus 5 KB, that is, FULL —5 KB - 101: Full minus 6 KB, that is, FULL —6 KB - 110: Full minus 7 KB, that is, FULL —7 KB - 111: Reserved This bit is reserved (and RO) if the Rx FIFO is 4 KB or less deep.
22	RFD_2	MSB of Threshold for Deactivating Flow Control If the DWC_gmac is configured for Rx FIFO size of 8 KB or more, this bit (when set) provides additional threshold levels for deactivating the flow control in both half-duplex and full-duplex modes. This bit (as Most Significant Bit) along with the RFD (Bits [12:11]) gives the following thresholds for deactivating flow control: - 100: Full minus 5 KB, that is, FULL —5 KB - 101: Full minus 6 KB, that is, FULL —6 KB - 110: Full minus 7 KB, that is, FULL —7 KB - 111: Reserved This bit is reserved (and RO) if the Rx FIFO is 4 KB or less deep.
21	TSF	Transmit Store and Forward When this bit is set, transmission starts when a full frame resides in the MTL Transmit FIFO. When this bit is set, the TTC values specified in Bits [16:14] are ignored. This bit should be changed only when the transmission is stopped.

Bits	Name	Description
20	FTF	Flush Transmit FIFO When this bit is set, the transmit FIFO controller logic is reset to its default values and thus all data in the Tx FIFO is lost or flushed. This bit is cleared internally when the flushing operation is complete. The Operation Mode register should not be written to until this bit is cleared. The data which is already accepted by the MAC transmitter is not flushed. It is scheduled for transmission and results in underflow and runt frame transmission.
16-14	TTC	Transmit Threshold Control These bits control the threshold level of the MTL Transmit FIFO. Transmission starts when the frame size within the MTL Transmit FIFO is larger than the threshold. In addition, full frames with a length less than the threshold are also transmitted. These bits are used only when Bit 21 (TSF) is reset. - 000: 64 - 001: 128 - 010: 192 - 011: 256 - 100: 40 - 101: 32 - 110: 24 - 111: 16

Bits	Name	Description
13	ST	Start or Stop Transmission Command When this bit is set, transmission is placed in the Running state, and the DMA checks the Transmit List at the current position for a frame to be transmitted. Descriptor acquisition is attempted either from the current position in the list, which is the Transmit List Base Address set by Register 4 (Transmit Descriptor List Address Register), or from the position retained when transmission was stopped previously. If the DMA does not own the current descriptor, transmission enters the Suspended state and Bit 2 (Transmit Buffer Unavailable) of Register 5 (Status Register) is set. The Start Transmission command is effective only when transmission is stopped. If the command is issued before setting Register 4 (Transmit Descriptor List Address Register), then the DMA behavior is unpredictable. When this bit is reset, the transmission process is placed in the Stopped state after completing the transmission of the current frame. The Next Descriptor position in the Transmit List is saved, and it becomes the current position when transmission is restarted. To change the list address, you need to program Register 4 (Transmit Descriptor List Address Register) with a new value when this bit is reset. The new value is considered when this bit is set again. The stop transmission command is effective only when the transmission of the current frame is complete or the transmission is in the Suspended state.

Bits	Name	Description
12-11	RFD	Threshold for Deactivating Flow Control (in half-duplex and full-duplex modes) These bits control the threshold (Fill-level of Rx FIFO) at which the flow control is de-asserted after activation. - 00: Full minus 1 KB, that is, FULL —1 KB - 01: Full minus 2 KB, that is, FULL —2 KB - 10: Full minus 3 KB, that is, FULL —3 KB - 11: Full minus 4 KB, that is, FULL —4 KB The de-assertion is effective only after flow control is asserted. If the Rx FIFO is 8 KB or more, an additional Bit (RFD_2) is used for more threshold levels as described in Bit 22. These bits are reserved and read-only when the Rx FIFO depth is less than 4 KB.
10-9	RFA	Threshold for Activating Flow Control (in half-duplex and full-duplex modes) These bits control the threshold (Fill level of Rx FIFO) at which the flow control is activated. - 00: Full minus 1 KB, that is, FULL—1KB. - 01: Full minus 2 KB, that is, FULL—2KB. - 10: Full minus 3 KB, that is, FULL—3KB. - 11: Full minus 4 KB, that is, FULL—4KB. These values are applicable only to Rx FIFOs of 4 KB or more and when Bit 8 (EFC) is set high. If the Rx FIFO is 8 KB or more, an additional Bit (RFA_2) is used for more threshold levels as described in Bit 23. These bits are reserved and read-only when the depth of Rx FIFO is less than 4 KB. Note: When FIFO size is exactly 4 KB, although the DWC_gmac allows you to program the value of these bits to 11, the software should not program these bits to 2'b11. The value 2'b11 means flow control on FIFO empty condition
8	EFC	Enable HW Flow Control When this bit is set, the flow control signal operation based on the fill-level of Rx FIFO is enabled. When reset, the flow control operation is disabled. This bit is not used (reserved and always reset) when the Rx FIFO is less than 4 KB.

Bits	Name	Description
7	FEF	Forward Error Frames When this bit is reset, the Rx FIFO drops frames with error status (CRC error, collision error, GMII_ER, giant frame, watchdog timeout, or overflow). However, if the start byte (write) pointer of a frame is already transferred to the read controller side (in Threshold mode), then the frame is not dropped. In the GMAC-MTL configuration in which the Frame Length FIFO is also enabled during core configuration, the Rx FIFO drops the error frames if that frame's start byte is not transferred (output) on the ARI bus. When the FEF bit is set, all frames except runt error frames are forwarded to the DMA. If the Bit 25 (RSF) is set and the Rx FIFO overflows when a partial frame is written, then the frame is dropped irrespective of the FEF bit setting. However, if the Bit 25 (RSF) is reset and the Rx FIFO overflows when a partial frame is written, then a partial frame may be forwarded to the DMA. Note: When FEF bit is reset, the giant frames are dropped if the giant frame status is given in Rx Status (in Table 8-6 or Table 8-23) in the following configurations: - The IP checksum engine (Type 1) and full checksum offload engine (Type 2) are not selected. - The advanced timestamp feature is not selected but the extended status is selected. The extended status is available with the following features: - L3-L4 filter in GMAC-CORE or GMAC-MTL configurations - Full checksum offload engine (Type 2) with enhanced descriptor format in the GMAC-DMA, GMAC-AHB, or GMAC-AXI configurations.
6	FUF	Forward Undersized Good Frames When set, the Rx FIFO forwards Undersized frames (that is, frames with no Error and length less than 64 bytes) including pad-bytes and CRC When reset, the Rx FIFO drops all frames of less than 64 bytes, unless a frame is already transferred because of the lower value of Receive Threshold, for example, RTC = 01.

Bits	Name	Description
5	DGF	Drop Giant Frames When set, the MAC drops the received giant frames in the Rx FIFO, that is, frames that are larger than the computed giant frame limit. When reset, the MAC does not drop the giant frames in the Rx FIFO. Note: This bit is available in the following configurations in which the giant frame status is not provided in Rx status and giant frames are not dropped by default: - Configurations in which IP Checksum Offload (Type 1) is selected in Rx - Configurations in which the IPC Full Checksum Offload Engine (Type 2) is selected in Rx with normal descriptor format - Configurations in which the Advanced Timestamp feature is selected In all other configurations, this bit is not used (reserved and always reset).
4-3	RTC	Receive Threshold Control These two bits control the threshold level of the MTL Receive FIFO. Transfer (request) to DMA starts when the frame size within the MTL Receive FIFO is larger than the threshold. In addition, full frames with length less than the threshold are automatically transferred. The value of 11 is not applicable if the configured Receive FIFO size is 128 bytes. These bits are valid only when the RSF bit is zero, and are ignored when the RSF bit is set to 1. - 00: 64 - 01: 32 - 10: 96 - 11: 128
2	OSF	Operate on Second Frame When this bit is set, it instructs the DMA to process the second frame of the Transmit data even before the status for the first frame is obtained.

Bits	Name	Description
1	SR	Start or Stop Receive When this bit is set, the Receive process is placed in the Running state. The DMA attempts to acquire the descriptor from the Receive list and processes the incoming frames. The descriptor acquisition is attempted from the current position in the list, which is the address set by the Register 3 (Receive Descriptor List Address Register) or the position retained when the Receive process was previously stopped. If the DMA does not own the descriptor, reception is suspended and Bit 7 (Receive Buffer Unavailable) of Register 5 (Status Register) is set. The Start Receive command is effective only when the reception has stopped. If the command is issued before setting Register 3 (Receive Descriptor List Address Register), the DMA behavior is unpredictable. When this bit is cleared, the Rx DMA operation is stopped after the transfer of the current frame. The next descriptor position in the Receive list is saved and becomes the current position after the Receive process is restarted. The Stop Receive command is effective only when the Receive process is in either the Running (waiting for receive packet) or in the Suspended state.

DMA_OP_MODE Bits

55.7.71 DMA_INTR_EN (0x101C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RSVD															NIE	AIE	ERE	FBE	RSVD		ETE	RWE	RSE	RUE	RIE	UNE	OVE	TJE	TUE	TSE	TIE	
N/A															RW	RW	RW	RW	N/A		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	x	x	0	0	0	0	0	0	0	0	0	0	0	

DMA_INTR_EN [31:0]

Bits	Name	Description
16	NIE	Normal Interrupt Summary Enable When this bit is set, normal interrupt summary is enabled. When this bit is reset, normal interrupt summary is disabled. This bit enables the following interrupts in Register 5 (Status Register): - Register 5[0]: Transmit Interrupt - Register 5[2]: Transmit Buffer Unavailable - Register 5[6]: Receive Interrupt - Register 5[14]: Early Receive Interrupt
15	AIE	Abnormal Interrupt Summary Enable When this bit is set, abnormal interrupt summary is enabled. When this bit is reset, the abnormal interrupt summary is disabled. This bit enables the following interrupts in Register 5 (Status Register): - Register 5[1]: Transmit Process Stopped - Register 5[3]: Transmit Jabber Timeout - Register 5[4]: Receive Overflow - Register 5[5]: Transmit Underflow - Register 5[7]: Receive Buffer Unavailable - Register 5[8]: Receive Process Stopped - Register 5[9]: Receive Watchdog Timeout - Register 5[10]: Early Transmit Interrupt - Register 5[13]: Fatal Bus Error
14	ERE	Early Receive Interrupt Enable When this bit is set with Normal Interrupt Summary Enable (Bit 16), the Early Receive Interrupt is enabled. When this bit is reset, the Early Receive Interrupt is disabled.
13	FBE	Fatal Bus Error Enable When this bit is set with Abnormal Interrupt Summary Enable (Bit 15), the Fatal Bus Error Interrupt is enabled. When this bit is reset, the Fatal Bus Error Enable Interrupt is disabled.
10	ETE	Early Transmit Interrupt Enable When this bit is set with an Abnormal Interrupt Summary Enable (Bit 15), the Early Transmit Interrupt is enabled. When this bit is reset, the Early Transmit Interrupt is disabled.
9	RWE	Receive Watchdog Timeout Enable When this bit is set with Abnormal Interrupt Summary Enable (Bit 15), the Receive Watchdog Timeout Interrupt is enabled. When this bit is reset, the Receive Watchdog Timeout Interrupt is disabled.

Bits	Name	Description
8	RSE	Receive Stopped Enable When this bit is set with Abnormal Interrupt Summary Enable (Bit 15), the Receive Stopped Interrupt is enabled. When this bit is reset, the Receive Stopped Interrupt is disabled.
7	RUE	Receive Buffer Unavailable Enable When this bit is set with Abnormal Interrupt Summary Enable (Bit 15), the Receive Buffer Unavailable Interrupt is enabled. When this bit is reset, the Receive Buffer Unavailable Interrupt is disabled.
6	RIE	Receive Interrupt Enable When this bit is set with Normal Interrupt Summary Enable (Bit 16), the Receive Interrupt is enabled. When this bit is reset, the Receive Interrupt is disabled.
5	UNE	Underflow Interrupt Enable When this bit is set with Abnormal Interrupt Summary Enable (Bit 15), the Transmit Underflow Interrupt is enabled. When this bit is reset, the Underflow Interrupt is disabled.
4	OVE	Overflow Interrupt Enable When this bit is set with Abnormal Interrupt Summary Enable (Bit 15), the Receive Overflow Interrupt is enabled. When this bit is reset, the Overflow Interrupt is disabled.
3	TJE	Transmit Jabber Timeout Enable When this bit is set with Abnormal Interrupt Summary Enable (Bit 15), the Transmit Jabber Timeout Interrupt is enabled. When this bit is reset, the Transmit Jabber Timeout Interrupt is disabled.
2	TUE	Transmit Buffer Unavailable Enable When this bit is set with Normal Interrupt Summary Enable (Bit 16), the Transmit Buffer Unavailable Interrupt is enabled. When this bit is reset, the Transmit Buffer Unavailable Interrupt is disabled.
1	TSE	Transmit Stopped Enable When this bit is set with Abnormal Interrupt Summary Enable (Bit 15), the Transmission Stopped Interrupt is enabled. When this bit is reset, the Transmission Stopped Interrupt is disabled.
0	TIE	Transmit Interrupt Enable When this bit is set with Normal Interrupt Summary Enable (Bit 16), the Transmit Interrupt is enabled. When this bit is reset, the Transmit Interrupt is disabled.

DMA_INTR_EN Bits

55.7.72 DMA_MISS_OVF_CNT (0x1020)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD			ONFCNTOVF	OVFFRMCNT												MISCNTOVF	MISFRMCNT														
N/A			RW	RW												RW	RW														
x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMA_MISS_OVF_CNT [31:0]

Bits	Name	Description
28	ONFCNTOVF	Overflow Bit for FIFO Overflow Counter This bit is set every time the Overflow Frame Counter (Bits[27:17]) overflows, that is, the Rx FIFO overflows with the overflow frame counter at maximum value. In such a scenario, the overflow frame counter is reset to all-zeros and this bit indicates that the rollover happened.
27-17	OVFFRMCNT	Overflow Frame Counter This field indicates the number of frames missed by the application. This counter is incremented each time the MTL FIFO overflows. The counter is cleared when this register is read with mci_be_i[2] at 1' b1.
16	MISCNTOVF	Overflow Bit for Missed Frame Counter This bit is set every time Missed Frame Counter (Bits[15:0]) overflows, that is, the DMA discards an incoming frame because of the Host Receive Buffer being unavailable with the missed frame counter at maximum value. In such a scenario, the Missed frame counter is reset to all-zeros and this bit indicates that the rollover happened.
15-0	MISFRMCNT	Missed Frame Counter This field indicates the number of frames missed by the controller because of the Host Receive Buffer being unavailable. This counter is incremented each time the DMA discards an incoming frame. The counter is cleared when this register is read with mci_be_i[0] at 1' b1.

DMA_MISS_OVF_CNT Bits

55.7.73 DMA_RX_INTR_WDOG (0x1024)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								RW							
N/A																								RW							
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	

DMA_RX_INTR_WDOG [31:0]

Bits	Name	Description
7-0	RIWT	RI Watchdog Timer Count This bit indicates the number of system clock cycles multiplied by 256 for which the watchdog timer is set. The watchdog timer gets triggered with the programmed value after the Rx DMA completes the transfer of a frame for which the RI status bit is not set because of the setting in the corresponding descriptor RDES1[31]. When the watchdog timer runs out, the RI bit is set and the timer is stopped. The watchdog timer is reset when the RI bit is set high because of automatic setting of RI as per RDES1[31] of any received frame.

DMA_RX_INTR_WDOG Bits

55.7.74 DMA_AXI_MODE (0x1028)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EN_LPI	LPI_XIT_FRM	RSVD						WR_OSR_LMT				RD_OSR_LMT				RSVD		ONEKBBE	AXI_AAL	RSVD				BLN256	BLN128	BLN64	BLN32	BLN16	BLN8	BLN4	UNDEF
RW	RW	N/A						RW				RW				N/A		RW	RW	N/A				RW	RW	RW	RW	RW	RW	RW	RW
0	0	x	x	x	x	x	x	0	0	0	0	0	0	0	0	x	x	0	0	x	x	x	x	0	0	0	0	0	0	0	0

DMA_AXI_MODE [31:0]

Bits	Name	Description
31	EN_LPI	Enable Low Power Interface (LPI) When set to 1, this bit enables the LPI mode supported by the GMAC-AXI configuration and accepts the LPI request from the AXI System Clock controller. When set to 0, this bit disables the LPI mode and always denies the LPI request from the AXI System Clock controller.

Bits	Name	Description
30	LPI_XIT_FRM	Unlock on Magic Packet or Remote Wake-Up Frame When set to 1, this bit enables the GMAC-AXI to come out of the LPI mode only when the magic packet or remote wake-up frame is received. When set to 0, this bit enables the GMAC-AXI to come out of LPI mode when any frame is received.
23-20	WR_OSR_LMT	AXI Maximum Write Outstanding Request Limit This value limits the maximum outstanding request on the AXI write interface. Maximum outstanding requests = WR_OSR_LMT+1 Note: - Bit 22 is reserved if AXI_GM_MAX_WR_REQUESTS = 4. - Bit 23 bit is reserved if AXI_GM_MAX_WR_REQUESTS != 16.
19-16	RD_OSR_LMT	AXI Maximum Read Outstanding Request Limit This value limits the maximum outstanding request on the AXI read interface. Maximum outstanding requests = RD_OSR_LMT+1 Note: - Bit 18 is reserved if AXI_GM_MAX_RD_REQUESTS = 4. - Bit 19 is reserved if AXI_GM_MAX_RD_REQUESTS != 16.
13	ONEKBBE	1 KB Boundary Crossing Enable for the GMAC-AXI Master When set, the GMAC-AXI master performs burst transfers that do not cross 1 KB boundary. When reset, the GMAC-AXI master performs burst transfers that do not cross 4 KB boundary.
12	AXI_AAL	Address-Aligned Beats This bit is read-only bit and reflects the Bit 25 (AAL) of Register 0 (Bus Mode Register). When this bit is set to 1, the GMAC-AXI performs address-aligned burst transfers on both read and write channels.
7	BLLEN256	AXI Burst Length 256 When this bit is set to 1, the GMAC-AXI is allowed to select a burst length of 256 on the AXI master interface. This bit is present only when the configuration parameter AXI_BL is set to 256. Otherwise, this bit is reserved and is read-only (RO).
6	BLLEN128	AXI Burst Length 128 When this bit is set to 1, the GMAC-AXI is allowed to select a burst length of 128 on the AXI master interface. This bit is present only when the configuration parameter AXI_BL is set to 128 or more. Otherwise, this bit is reserved and is read-only (RO).

Bits	Name	Description
5	BLLEN64	AXI Burst Length 64 When this bit is set to 1, the GMAC-AXI is allowed to select a burst length of 64 on the AXI master interface. This bit is present only when the configuration parameter AXI_BL is set to 64 or more. Otherwise, this bit is reserved and is read-only (RO).
4	BLLEN32	AXI Burst Length 32 When this bit is set to 1, the GMAC-AXI is allowed to select a burst length of 32 on the AXI master interface. This bit is present only when the configuration parameter AXI_BL is set to 32 or more. Otherwise, this bit is reserved and is read-only (RO).
3	BLLEN16	AXI Burst Length 16 When this bit is set to 1 or UNDEF is set to 1, the GMAC-AXI is allowed to select a burst length of 16 on the AXI master interface.
2	BLLEN8	AXI Burst Length 8 When this bit is set to 1, the GMAC-AXI is allowed to select a burst length of 8 on the AXI master interface. Setting this bit has no effect when UNDEF is set to 1.
1	BLLEN4	AXI Burst Length 4 When this bit is set to 1, the GMAC-AXI is allowed to select a burst length of 4 on the AXI master interface. Setting this bit has no effect when UNDEF is set to 1.
0	UNDEF	AXI Undefined Burst Length This bit is read-only bit and indicates the complement (invert) value of Bit 16 (FB) in Register 0 (Bus Mode Register). - When this bit is set to 1, the GMAC-AXI is allowed to perform any burst length equal to or below the maximum allowed burst length programmed in Bits[7:3]. - When this bit is set to 0, the GMAC-AXI is allowed to perform only fixed burst lengths as indicated by BLLEN256, BLLEN128, BLLEN64, BLLEN32, BLLEN16, BLLEN8, or BLLEN4, or a burst length of 1. If UNDEF is set and none of the BLLEN bits is set, then GMAC-AXI is allowed to perform a burst length of 16.

DMA_AXI_MODE Bits

55.7.75 DMA_BUS_STATUS (0x102C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																														AXIRDSTS	AXWHSTS
N/A																														RW	RW
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0

DMA_BUS_STATUS [31:0]

Bits	Name	Description
1	AXIRDSTS	AXI Master Read Channel Status When high, it indicates that AXI master's read channel is active and transferring data.
0	AXWHSTS	AXI Master Write Channel or AHB Master Status When high, it indicates that AXI master's write channel is active and transferring data in the GMAC-AXI configuration. In the GMAC-AHB configuration, it indicates that the AHB master interface FSMs are in the non-idle state.

DMA_BUS_STATUS Bits

55.7.76 DMA_CURR_HOST_TX_DESC (0x1048)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CURTDESAPTR																															
																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMA_CURR_HOST_TX_DESC [31:0]

Bits	Name	Description
31-0	CURTDESAPTR	Host Transmit Descriptor Address Pointer Cleared on Reset. Pointer updated by the DMA during operation.

DMA_CURR_HOST_TX_DESC Bits

55.7.77 DMA_CURR_HOST_RX_DESC (0x104C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CURRDESAPTR																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMA_CURR_HOST_RX_DESC [31:0]

Bits	Name	Description
31-0	CURRDESAPTR	Host Receive Descriptor Address Pointer Cleared on Reset. Pointer updated by the DMA during operation.

DMA_CURR_HOST_RX_DESC Bits

55.7.78 DMA_CURR_HOST_TX_BUF (0x1050)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CURTBUFAPTR																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMA_CURR_HOST_TX_BUF [31:0]

Bits	Name	Description
31-0	CURTBUFAPTR	Host Transmit Buffer Address Pointer Cleared on Reset. Pointer updated by the DMA during operation.

DMA_CURR_HOST_TX_BUF Bits

55.7.79 DMA_CURR_HOST_RX_BUF (0x1054)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CURRBUFAPTR																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMA_CURR_HOST_RX_BUF [31:0]

Bits	Name	Description
31-0	CURRBUFAPTR	Host Receive Buffer Address Pointer Cleared on Reset. Pointer updated by the DMA during operation.

DMA_CURR_HOST_RX_BUF Bits

55.7.80 CTRL0 (0x3000)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
RSVD																		ENET0_RXCLK_DLY_SEL						RSVD		ENET0_TXCLK_DLY_SEL							
N/A																		RW						N/A		RW							
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	x	x	0	0	0	0	0	0		

CTRL0 [31:0]

Bits	Name	Description
13-8	ENET0_RXCLK_DLY_SEL	
5-0	ENET0_TXCLK_DLY_SEL	

CTRL0 Bits

55.7.81 CTRL2 (0x3008)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RSVD		ENET0_LPI_IRQ_EN	RSVD										ENET0_REFCLK_OE	RSVD			ENET0_PHY_INF_SEL			ENET0_FLOWCTRL	RSVD	ENET0_RMII_TXCLK_SEL	RSVD									
N/A	RW	N/A										RW	N/A			RW			RW	N/A	RW	N/A										
x	x	0	x	x	x	x	x	x	x	x	x	0	x	x	x	0	0	0	0	x	0	x	x	x	x	x	x	x	x	x	x	

CTRL2 [31:0]

Bits	Name	Description
29	ENET0_LPI_IRQ_EN	lowpower interrupt enable, for internal use only, user should use core registers for enable/disable interrupt
19	ENET0_REFCLK_OE	set to enable output 50MHz clock to rmii phy. User should set it if use soc internal clock as refclk
15-13	ENET0_PHY_INF_SEL	PHY mode select 000MII; 001RGMII; 100RMII; should be set before config IOMUX, otherwise may cause glitch for RGMII
12	ENET0_FLOWCTL	flow control request
10	ENET0_RMII_TXCLK_SEL	RMII mode output clock pad select set to use txck as RMII refclk; clr to use rxck as RMII refclk; default 0(rxck) refclk is always from pad, can use external clock from pad, or use internal clock output to pad then loopback.

CTRL2 Bits

56 Industrial Network Overview

This chapter describes the industrial network communication peripherals of this product.

56.1 EtherCat Slave Controller ESC

ESC supports the following features:

- 3 ports, supporting MII and RMII
- 8 FMMU
- 8 SyncManagers
- 60KB PDRAM
- 64-bit Distributed Clock

SYNC0 and SYNC1 of ESC serve as 2 input signals to the interconnection manager TRIGM, and 1 output signal of TRIGM is connected to the LATCH0 signal of ESC.

57 EtherCAT slave controller ESC

57.1 Overview

EtherCAT is an Ethernet-based fieldbus system developed by Beckhoff Automation and is one of the IEC61158 fieldbus standards. This product integrates the EtherCAT IP core authorized by Beckhoff and supports the following features:

- Number of ports: 3
- FMMU: 8
- SyncManager: 8
- Process data RAM (PDRAM): 60KB
- Distributed Clocks: 64-bit
- PHY interface: MII/RMII
- Process data interface (PDI): On-chip bus
- EEPROM: Support I2C plug-in and analog EEPROM
- GPIO: 64 units

EtherCAT official reference material:

[Hardware Data Sheet Section I - Technology](#)

[Hardware Data Sheet Section II - Register Description](#)

57.2 Function Description

57.2.1 EtherCAT Physical Layer Interface

ESC module supports up to 3 Ethernet PHY interfaces and supports MII interface types. It also has a built-in MII to RMII conversion module to support RMII PHY with low pin count. Setting the register ESC.PHY_CFG0[PORTn_RMII_EN] to 1 switches the PHY interface to RMII mode.

supports connecting TXCK to automatically adjust the phase of the TXD signal. The configuration register ESC.PHY_CFG0[PORTn_TX_SHIFT] allows you to manually adjust the phase offset of TXD.

provides an MDIO interface to configure Ethernet PHY.

57.2.2 EEPROM

EtherCAT slave stations require nonvolatile memory to save device configuration and description information (EtherCAT Slave Information (ESI)). The ESC module supports external EEPROM to the I2C interface, and also supports CPU Analog EEPROM (EEPROM Emulation), that is, ESI is stored in nonvolatile storage areas such as Flash, and the CPU is responsible for transporting it to the ESC for reading by the EtherCAT master station.

works in analog EEPROM mode by default, and no external EEPROM is required.

If an external EEPROM is used, the register ESC.GPR_CFG0[EEPROM_EMU] needs to be cleared.

57.2.3 Clock

[Figure 91](#) describe the clock structure of ESC

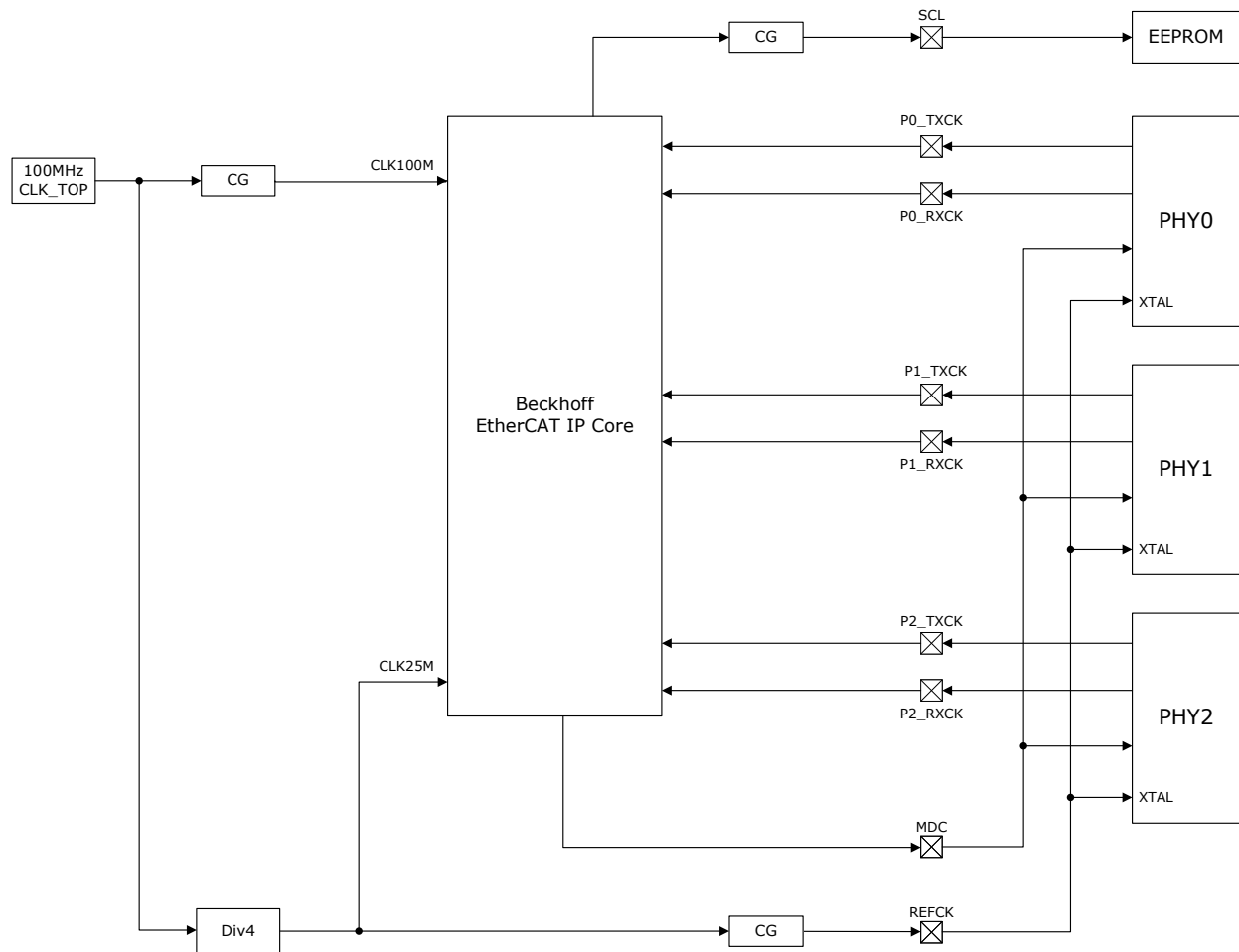


Figure 91: ESC clock structure diagram

ESC module uses the 100MHz functional clock provided by the chip clock system as a single clock source. This functional clock is divided by 4 to produce a 25 MHz clock that is output from REFCK IO and serves as the clock source input to the PHY. The

ESC module and the Ethernet PHY must use the same clock source. It is not supported for the ESC module to use the chip's internal functional clock and the PHY to use an external 25MHz crystal oscillator. The

100MHz clock has a gating function and its enable register is ESC.GPR_CFG0[CLK100_EN]. The

25MHz output clock REFCK has a gating function, and its enable register is ESC.PHY_CFG1[REFCK_25M_OE].

I2C interface clock SCL has a gating function, and its enable register is ESC.GPR_CFG1[I2C_SCLK_EN].

57.2.4 Reset

The logic circuit of the ESC module can only be hardware reset following the subsystem in which it is located. The

ESC module provides two registers ESC_RST_ECAT and ESC_RST_PDI, which can be accessed by the EtherCAT master and slave CPU respectively to trigger reset requests. As shown in [Figure 92](#), reset requests can trigger interrupts and reset of Ethernet PHY.

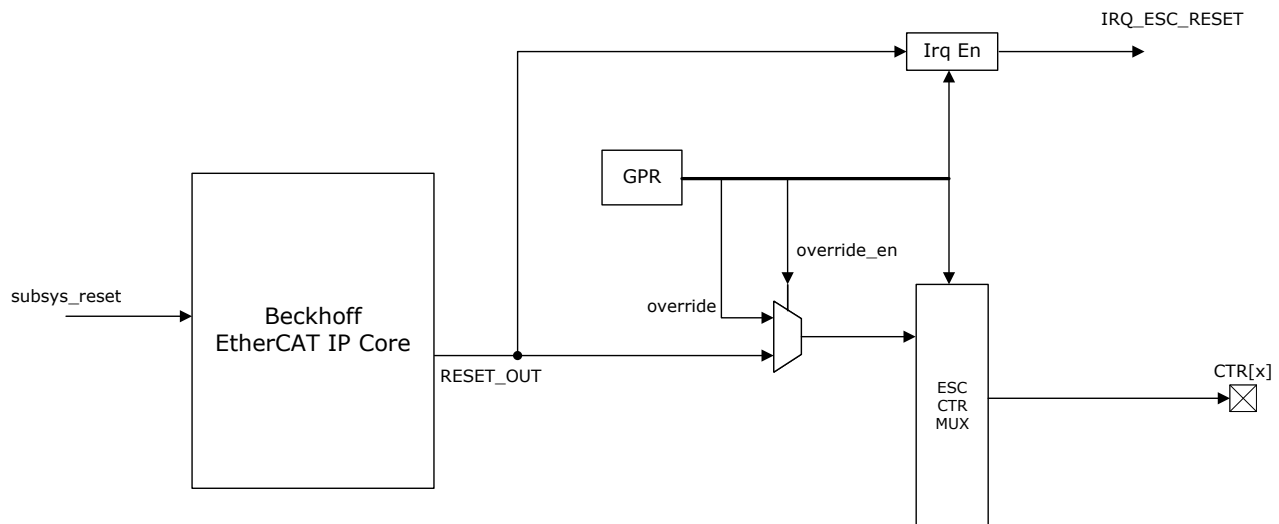


Figure 92: ESC reset function diagram

reset interrupt requests is `ESC.GPR_CFG1[RSTO_IRQ_EN]`. The reset of the

Ethernet PHY is generated by combination control. There are two control sources. One is the reset request issued by the EtherCAT IP Core, which is generated through the configuration register `ESC_RST_ECOT` or `ESC_RST_PDI`, and the other is directly controlled by the register `ESC.GPR_CFG1[RSTO_OVRD]`. In order to ensure that the Ethernet PHY remains in a reset state until ESC initialization is completed, GPR directly controls the reset of the Ethernet PHY by default, and the selection of the control source is determined by the register `ESC.GPR_CFG1[RSTO_OVRD_ENJ]`.

As shown in the figure, the reset signal of the Ethernet PHY passes through a `CTR_MUX` module before being sent to IO. It can support the PHY Reset signal to flexibly select different IO and modify the effective level. See [Subsection 57.2.6](#) for details.

57.2.5 LED and LINK status

The ESC module provides an LED RUN signal to the IO output. The

ESC module provides the LED ERR signal to the IO output. The

ESC module does not support STATE RUN LED output. The

ESC module provides a LINKACT signal for each port to the IO output. It is recommended to use this signal to drive the status indicating LEDs of each ESC port instead of using the LED signal of Ethernet PHY. The

ESC module provides an IO input signal of MII LINK for each port, which is usually connected to a certain LED signal that meets the requirements of the Ethernet PHY to realize the Link Detection function of EtherCAT. The IO inputs of MII LINK are overridden by GPR by default: registers `ESC.GPR_CFG2[NMII_LINKn_FROM_IO]` and `ESC.GPR_CFG2[NMII_LINK0_GPR]`. If the port is not in use, you must set GPR to ensure that the port is in No Link state. The

ESC module supports RXER-based Enhanced Link Detection.

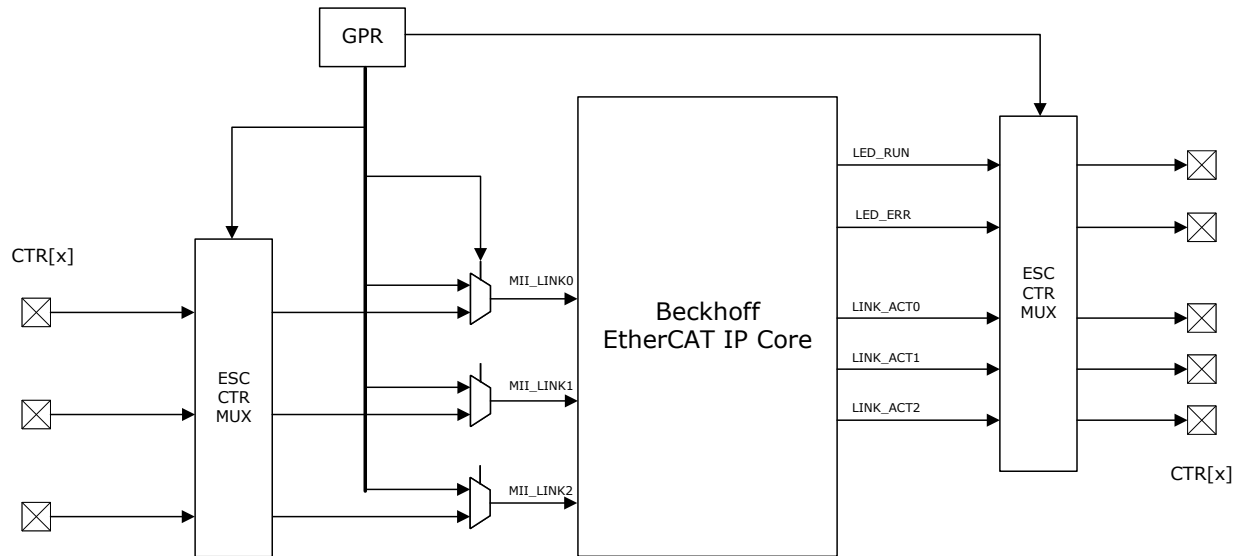


Figure 93: ESC Link function diagram

As shown in [Figure 93](#), there is a CTR MUX module between each LED and Link status signal and IO described in this section to achieve flexible mapping and level adjustment of each signal and IO. See [Subsection 57.2.6](#) for details.

57.2.6 IO Multipurpose Selector CTR MUX

As mentioned above, EtherCAT has a total of 9 non-MII interface IOs (including PHY Reset, LED, LINKACK, MII LINK). Using the CTR MUX module, these signals can be mapped to any IO from CTR[0] to CTR[8]. In this way, when only a part of the signals are used, such as only 2 or 1 EtherCAT port, you can flexibly select the IO to be used, and also reserve suitable IO for other functions. The selection of IO maps is controlled by the ESC.CTRn register.

CTR MUX also supports inverting arbitrary signals to support active-high or active-low levels such as MII LINK.

57.2.7 SYNC and LATCH

The Distributed Clocks (DC) function of EtherCAT IP Core provides 2 SYNC signals. In the ESC module, the SYNC signal supports the following functions:

- supports two IRQ interrupts for SYNC0 and SYNC1 respectively
- supports two DMA requests for SYNC0 and SYNC1 respectively
- supports output of SYNC0 and SYNC1 to IO ESC.EVTO[0] and ESC.EVTO[1] respectively
- SYNC0 and SYNC1 can control the locking of ESC's GPIO input and output signals
- supports the output of SYNC0 and SYNC1 to other modules on the chip

The enable register for triggering interrupts by SYNC0 and SYNC1 is ESC.GPR_CFG1[SYNcN_IRQ_EN]. The enable register for

SYNC0 and SYNC1 triggering DMA requests is ESC.GPR_CFG1[SYNcN_DMA_EN].

Details about how SYNC0 and SYNC1 control ESC GPIOs are described in [Subsection 57.2.9](#). See [Section 56.1](#) for details on

's SYNC0 and SYNC1 outputs to other modules on the chip.

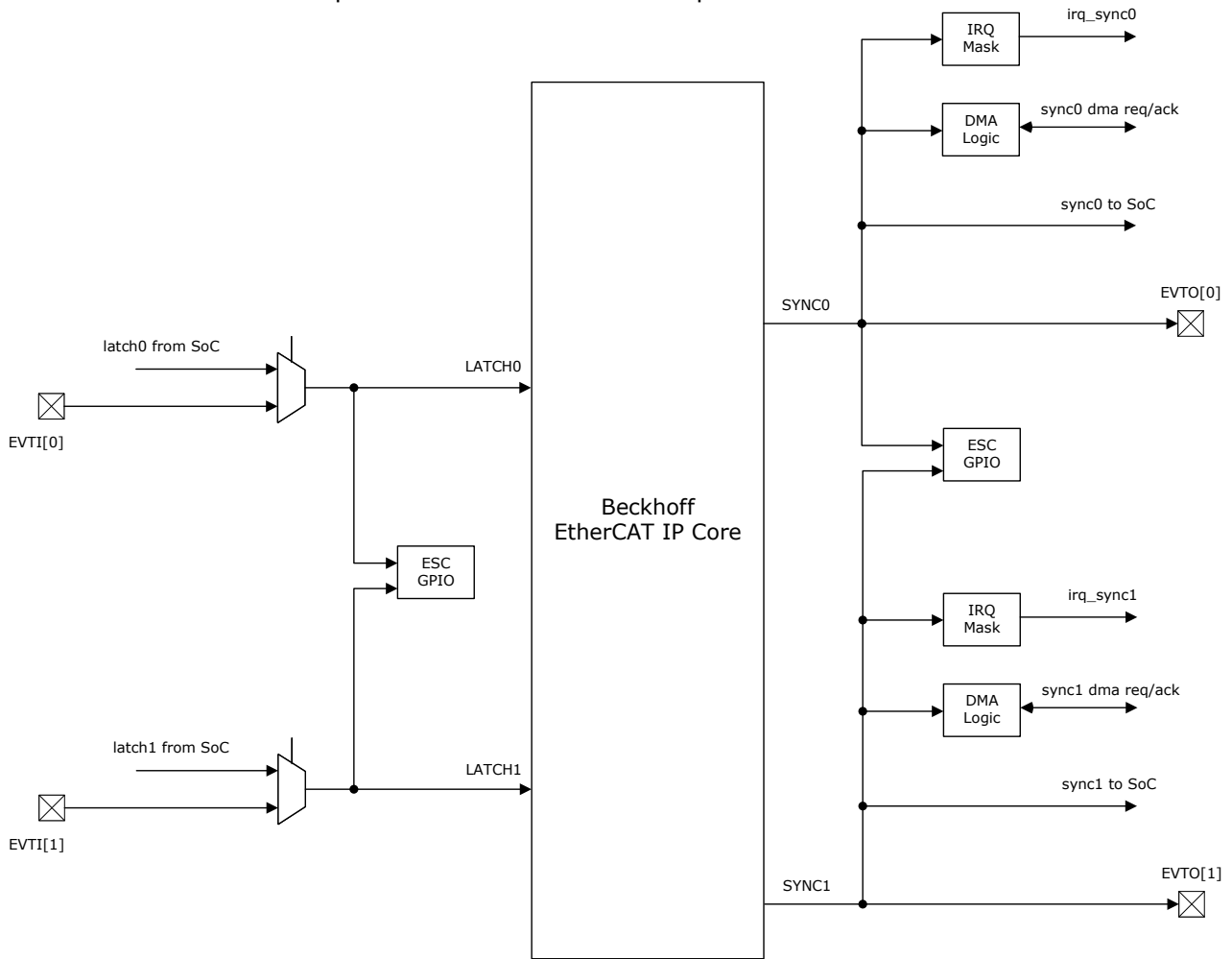


Figure 94: Functional Diagram of ESC SYNC and LATCH

The DC function of EtherCAT IP Core provides 2 LATCH signals. There are the following options for the sources of LATCH signals in the ESC module:

- from IO ESC.EVTI[0] and ESC.EVTI[1]
- From other modules in the chip

LATCH signal source selection is controlled by register ESC.GPR_CFG1[LATCHn_FROM_IO]. For details on

LATCH from other modules in the chip, please see [Section 56.1](#).

In addition, the LATCH0 and LATCH1 signals can also control the locking of the GPIO input and output signals of the ESC.

57.2.8 Interrupts and DMA

ESC module supports the following interrupts (each interrupt has a separate interrupt number):

- Interrupts for EtherCAT IP Core
- SYNC0 interrupt
- SYNC1 interrupt

- RESET interrupt

ESC module supports two interrupt requests: SYNC0 and SYNC1.

57.2.9 GPIO

The ESC module supports 64 GPIOs, as shown in Figure 95 :

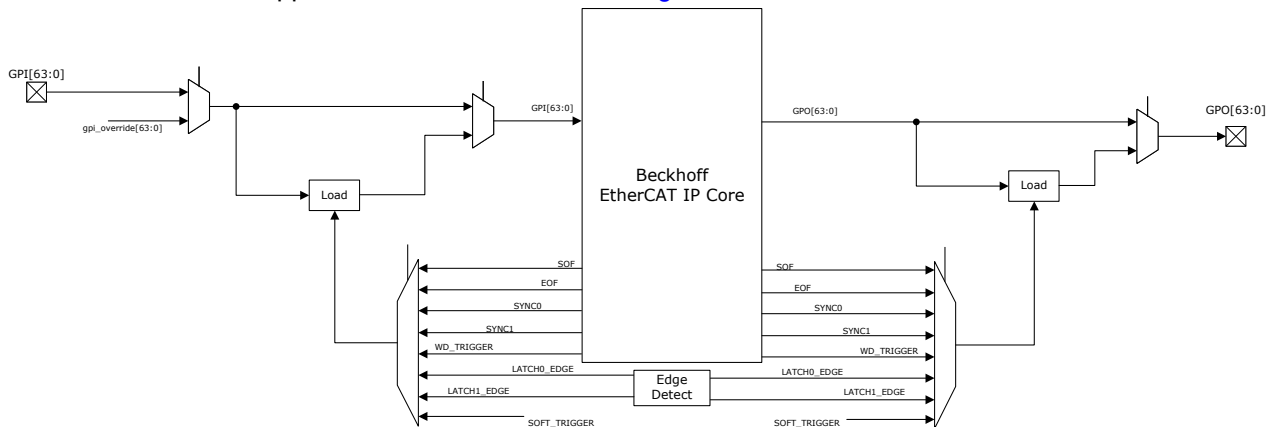


Figure 95: ESC GPIO function diagram

EtherCAT IP Core's GPO signal can directly drive the GPO IO output or output it after being latched through a Trigger event. The control of this selection is controlled by the register ESC.GPIO_CTRL[gpo_trig_en].

latch trigger events include rising or falling edges of SOF, EOF, SYNC0, SYNC1, LATCH0 and LATCH1, Watch Dog Trigger, and software write GPR operations. See register ESC.GPIO_CTRL[gpo_trig_sel] for details. The input of

IO can be connected directly or connected to the GPI port of the EtherCAT IP Core via a Trigger event latch. The control of this selection is controlled by the register ESC.GPIO_CTRL[gpi_trig_en].

latch trigger events include rising or falling edges of SOF, EOF, SYNC0, SYNC1, LATCH0 and LATCH1, Watch Dog Trigger, and software write GPR operations. See register ESC.GPIO_CTRL[gpi_trig_sel] for details.

In addition, the GPI signal can also be overridden by the GPR register. See registers ESC.GPIO_CTRL[gpi_override_en] and ESC.GPI_OVERRIDE0/1.

The trigger latch of GPIO can be configured in groups of 8 IOs. When the register ESC.GPIO_CTRL[gpo_trig_sel] or

ESC.GPIO_CTRL[gpi_trig_sel] is set to 0b1110, the registers ESC.GPI_TRIG_CFG and ESC.GPO_TRIG_CFG control the trigger latch of GPIO. This register controls the behavior of an 8-GPIO group every 4 bits. The 4-bit control information is encoded in the same way as ESC.GPIO_CTRL[gpo_trig_sel], that is, ESC.GPO_TRIG_CFG[3:0] can control the trigger latch behavior of GPO[7:0], ESC.GPO_TRIG_CFG[7:4] can control GPO[15:8], and so on.

57.3 ESC register

ESC is as follows:

ESC base address: 0xF1700000

Offset	Register Name	Description	Reset Value
0x0000	TYPE	Type of EtherCAT controller	0xA2
0x0001	REVISION	Revision of EtherCAT controller	0x00
0x0002	BUILD	Build of EtherCAT controller	0x0000
0x0004	FMMU_NUM	FMMU supported	0x08
0x0005	SYNCM_NUM	SyncManagers supported	0x08
0x0006	RAM_SIZE	RAM Size	0x3C
0x0007	PORT_DESC	Port Descriptor	0x3F
0x0008	FEATURE	ESC Feature supported	0x008C
0x0010	STATION_ADDR	Configured Station Address	0x0000
0x0012	STATION_ALS	Configured Station Alias	0x0000
0x0020	REG_WEN	Register Write Enable	0x00
0x0021	REG_WP	Register Write Protection	0x00
0x0030	ESC_WEN	ESC Write Enable	0x00
0x0031	ESC_WP	ESC Write Protection	0x00
0x0040	ESC_RST_EC	ESC Reset ECAT	0x00
0x0041	ESC_RST_PDI	ESC Reset PDI	0x00
0x0100	ESC_DL_CTRL	ESC DL Control	0x00000000
0x0108	PHYSICAL_RW_OFFSET	Physical Read/Write Offset	0x0000
0x0110	ESC_DL_STAT	ESC DL Status	0x0000
0x0120	AL_CTRL	AL Control	0x0001
0x0130	AL_STAT	AL Status	0x0001
0x0134	AL_STAT_CODE	AL Status Code	0x0000
0x0138	RUN_LED_OVRD	RUN LED Override	0x00
0x0139	ERR_LED_OVRD	ERR LED Override	0x00
0x0140	PDI_CTRL	PDI Control	0x00
0x0141	ESC_CFG	ESC Configuration	0x01
0x014E	PDI_INFO	PDI Information	0x0000
0x0150	PDI_CFG	PDI Configuration	0x84
0x0151	PDI_SL_CFG	PDI Sync/Latch[1:0] Configuration	0x00
0x0152	PDI_EXT_CFG	PDI Extended Configuration	0x0000
0x0200	ECAT_EVT_MSK	ECAT Event Mask	0x0000
0x0204	PDI_AL_EVT_MSK	PDI AL Event Mask	0x00000000
0x0210	ECAT_EVT_REQ	ECAT Event Request	0x0000
0x0220	AL_EVT_REQ	AL Event Request	0x00000000
0x0300	RX_ERR_CNT[PORT0]	RX Error Counter	0x0000
0x0302	RX_ERR_CNT[PORT1]	RX Error Counter	0x0000
0x0304	RX_ERR_CNT[PORT2]	RX Error Counter	0x0000
0x0306	RX_ERR_CNT[PORT3]	RX Error Counter	0x0000
0x0308	FWD_RX_ERR_CNT[PORT0]	Forwarded RX Error Counter	0x00

Offset	Register Name	Description	Reset Value
0x0309	FWD_RX_ERR_CNT[PORT1]	Forwarded RX Error Counter	0x00
0x030A	FWD_RX_ERR_CNT[PORT2]	Forwarded RX Error Counter	0x00
0x030B	FWD_RX_ERR_CNT[PORT3]	Forwarded RX Error Counter	0x00
0x030C	ECAT_PU_ERR_CNT	ECAT Processing Unit Error Counter	0x00
0x030D	PDI_ERR_CNT	PDI Error Counter	0x00
0x0310	LOST_LINK_CNT[PORT0]	Lost Link Counter	0x00
0x0311	LOST_LINK_CNT[PORT1]	Lost Link Counter	0x00
0x0312	LOST_LINK_CNT[PORT2]	Lost Link Counter	0x00
0x0313	LOST_LINK_CNT[PORT3]	Lost Link Counter	0x00
0x0400	WDG_DIV	Watchdog Divider	0x09C2
0x0410	WDG_TIME_PDI	Watchdog Time PDI	0x03E8
0x0420	WDG_TIME_PDAT	Watchdog Time Process Data	0x0000
0x0440	WDG_STAT_PDAT	Watchdog Status Process Data	0x0000
0x0442	WDG_CNT_PDAT	Watchdog Counter Process Data	0x00
0x0443	WDG_CNT_PDI	Watchdog Counter PDI	0x00
0x0500	EEPROM_CFG	EEPROM Configuration	0x00
0x0501	EEPROM_PDI_ACC_STAT	EEPROM PDI Access State	0x00
0x0502	EEPROM_CTRL_STAT	EEPROM Control/Status	0x0000
0x0504	EEPROM_ADDR	EEPROM Address	0x00000000
0x0508	EEPROM_DATA	EEPROM Data	0x0000000000000000
0x0510	MII_MNG_CS	MII Management Control/Status	0x0000
0x0512	PHY_ADDR	PHY Address	0x00
0x0513	PHY_REG_ADDR	PHY Register Address	0x00
0x0514	PHY_DATA	PHY Data	0x0000
0x0516	MIIM_ECAT_ACC_STAT	MII Management ECAT Access State	0x00
0x0517	MIIM_PDI_ACC_STAT	MII Management PDI Access State	0x00
0x0518	PHY_STAT[PORT0]	PHY Port	0x00
0x0519	PHY_STAT[PORT1]	PHY Port	0x00
0x051A	PHY_STAT[PORT2]	PHY Port	0x00
0x051B	PHY_STAT[PORT3]	PHY Port	0x00
0x0600	FMMU[0][LOGIC_START_ADDR]	Logical Start Address	0x00000000
0x0604	FMMU[0][LENGTH]	Length	0x0000
0x0606	FMMU[0][LOGIC_START_BIT]	Logical Start Bit	0x00
0x0607	FMMU[0][LOGIC_STOP_BIT]	Logical Stop Bit	0x00

Offset	Register Name	Description	Reset Value
0x0608	FMMU[0][PHYSICAL_START_ADDR]	Physical Start Address	0x0000
0x060A	FMMU[0][PHYSICAL_START_BIT]	Physical Start Bit	0x00
0x060B	FMMU[0][TYPE]	Type	0x00
0x060C	FMMU[0][ACTIVATE]	Activate	0x00
0x0610	FMMU[1][LOGIC_START_ADDR]	Logical Start Address	0x00000000
0x0614	FMMU[1][LENGTH]	Length	0x0000
0x0616	FMMU[1][LOGIC_START_BIT]	Logical Start Bit	0x00
0x0617	FMMU[1][LOGIC_STOP_BIT]	Logical Stop Bit	0x00
0x0618	FMMU[1][PHYSICAL_START_ADDR]	Physical Start Address	0x0000
0x061A	FMMU[1][PHYSICAL_START_BIT]	Physical Start Bit	0x00
0x061B	FMMU[1][TYPE]	Type	0x00
0x061C	FMMU[1][ACTIVATE]	Activate	0x00
0x0620	FMMU[2][LOGIC_START_ADDR]	Logical Start Address	0x00000000
0x0624	FMMU[2][LENGTH]	Length	0x0000
0x0626	FMMU[2][LOGIC_START_BIT]	Logical Start Bit	0x00
0x0627	FMMU[2][LOGIC_STOP_BIT]	Logical Stop Bit	0x00
0x0628	FMMU[2][PHYSICAL_START_ADDR]	Physical Start Address	0x0000
0x062A	FMMU[2][PHYSICAL_START_BIT]	Physical Start Bit	0x00
0x062B	FMMU[2][TYPE]	Type	0x00
0x062C	FMMU[2][ACTIVATE]	Activate	0x00
0x0630	FMMU[3][LOGIC_START_ADDR]	Logical Start Address	0x00000000
0x0634	FMMU[3][LENGTH]	Length	0x0000
0x0636	FMMU[3][LOGIC_START_BIT]	Logical Start Bit	0x00
0x0637	FMMU[3][LOGIC_STOP_BIT]	Logical Stop Bit	0x00
0x0638	FMMU[3][PHYSICAL_START_ADDR]	Physical Start Address	0x0000
0x063A	FMMU[3][PHYSICAL_START_BIT]	Physical Start Bit	0x00
0x063B	FMMU[3][TYPE]	Type	0x00
0x063C	FMMU[3][ACTIVATE]	Activate	0x00
0x0640	FMMU[4][LOGIC_START_ADDR]	Logical Start Address	0x00000000

Offset	Register Name	Description	Reset Value
0x0644	FMMU[4][LENGTH]	Length	0x0000
0x0646	FMMU[4][LOGIC_START_BIT]	Logical Start Bit	0x00
0x0647	FMMU[4][LOGIC_STOP_BIT]	Logical Stop Bit	0x00
0x0648	FMMU[4][PHYSICAL_START_ADDR]	Physical Start Address	0x0000
0x064A	FMMU[4][PHYSICAL_START_BIT]	Physical Start Bit	0x00
0x064B	FMMU[4][TYPE]	Type	0x00
0x064C	FMMU[4][ACTIVATE]	Activate	0x00
0x0650	FMMU[5][LOGIC_START_ADDR]	Logical Start Address	0x00000000
0x0654	FMMU[5][LENGTH]	Length	0x0000
0x0656	FMMU[5][LOGIC_START_BIT]	Logical Start Bit	0x00
0x0657	FMMU[5][LOGIC_STOP_BIT]	Logical Stop Bit	0x00
0x0658	FMMU[5][PHYSICAL_START_ADDR]	Physical Start Address	0x0000
0x065A	FMMU[5][PHYSICAL_START_BIT]	Physical Start Bit	0x00
0x065B	FMMU[5][TYPE]	Type	0x00
0x065C	FMMU[5][ACTIVATE]	Activate	0x00
0x0660	FMMU[6][LOGIC_START_ADDR]	Logical Start Address	0x00000000
0x0664	FMMU[6][LENGTH]	Length	0x0000
0x0666	FMMU[6][LOGIC_START_BIT]	Logical Start Bit	0x00
0x0667	FMMU[6][LOGIC_STOP_BIT]	Logical Stop Bit	0x00
0x0668	FMMU[6][PHYSICAL_START_ADDR]	Physical Start Address	0x0000
0x066A	FMMU[6][PHYSICAL_START_BIT]	Physical Start Bit	0x00
0x066B	FMMU[6][TYPE]	Type	0x00
0x066C	FMMU[6][ACTIVATE]	Activate	0x00
0x0670	FMMU[7][LOGIC_START_ADDR]	Logical Start Address	0x00000000
0x0674	FMMU[7][LENGTH]	Length	0x0000
0x0676	FMMU[7][LOGIC_START_BIT]	Logical Start Bit	0x00
0x0677	FMMU[7][LOGIC_STOP_BIT]	Logical Stop Bit	0x00
0x0678	FMMU[7][PHYSICAL_START_ADDR]	Physical Start Address	0x0000
0x067A	FMMU[7][PHYSICAL_START_BIT]	Physical Start Bit	0x00
0x067B	FMMU[7][TYPE]	Type	0x00

Offset	Register Name	Description	Reset Value
0x067C	FMMU[7][ACTIVATE]	Activate	0x00
0x0800	SYNCM[0][PHYSICAL_START_ADDR]	Physical Start Address	0x0000
0x0802	SYNCM[0][LENGTH]	Length	0x0000
0x0804	SYNCM[0][CONTROL]	Control	0x00
0x0805	SYNCM[0][STATUS]	Status	0x00
0x0806	SYNCM[0][ACTIVATE]	Activate	0x00
0x0807	SYNCM[0][PDI_CTRL]	PDI Control	0x00
0x0808	SYNCM[1][PHYSICAL_START_ADDR]	Physical Start Address	0x0000
0x080A	SYNCM[1][LENGTH]	Length	0x0000
0x080C	SYNCM[1][CONTROL]	Control	0x00
0x080D	SYNCM[1][STATUS]	Status	0x00
0x080E	SYNCM[1][ACTIVATE]	Activate	0x00
0x080F	SYNCM[1][PDI_CTRL]	PDI Control	0x00
0x0810	SYNCM[2][PHYSICAL_START_ADDR]	Physical Start Address	0x0000
0x0812	SYNCM[2][LENGTH]	Length	0x0000
0x0814	SYNCM[2][CONTROL]	Control	0x00
0x0815	SYNCM[2][STATUS]	Status	0x00
0x0816	SYNCM[2][ACTIVATE]	Activate	0x00
0x0817	SYNCM[2][PDI_CTRL]	PDI Control	0x00
0x0818	SYNCM[3][PHYSICAL_START_ADDR]	Physical Start Address	0x0000
0x081A	SYNCM[3][LENGTH]	Length	0x0000
0x081C	SYNCM[3][CONTROL]	Control	0x00
0x081D	SYNCM[3][STATUS]	Status	0x00
0x081E	SYNCM[3][ACTIVATE]	Activate	0x00
0x081F	SYNCM[3][PDI_CTRL]	PDI Control	0x00
0x0820	SYNCM[4][PHYSICAL_START_ADDR]	Physical Start Address	0x0000
0x0822	SYNCM[4][LENGTH]	Length	0x0000
0x0824	SYNCM[4][CONTROL]	Control	0x00
0x0825	SYNCM[4][STATUS]	Status	0x00
0x0826	SYNCM[4][ACTIVATE]	Activate	0x00
0x0827	SYNCM[4][PDI_CTRL]	PDI Control	0x00
0x0828	SYNCM[5][PHYSICAL_START_ADDR]	Physical Start Address	0x0000
0x082A	SYNCM[5][LENGTH]	Length	0x0000
0x082C	SYNCM[5][CONTROL]	Control	0x00

Offset	Register Name	Description	Reset Value
0x082D	SYNCM[5][STATUS]	Status	0x00
0x082E	SYNCM[5][ACTIVATE]	Activate	0x00
0x082F	SYNCM[5][PDI_CTRL]	PDI Control	0x00
0x0830	SYNCM[6][PHYSICAL_START_ADDR]	Physical Start Address	0x0000
0x0832	SYNCM[6][LENGTH]	Length	0x0000
0x0834	SYNCM[6][CONTROL]	Control	0x00
0x0835	SYNCM[6][STATUS]	Status	0x00
0x0836	SYNCM[6][ACTIVATE]	Activate	0x00
0x0837	SYNCM[6][PDI_CTRL]	PDI Control	0x00
0x0838	SYNCM[7][PHYSICAL_START_ADDR]	Physical Start Address	0x0000
0x083A	SYNCM[7][LENGTH]	Length	0x0000
0x083C	SYNCM[7][CONTROL]	Control	0x00
0x083D	SYNCM[7][STATUS]	Status	0x00
0x083E	SYNCM[7][ACTIVATE]	Activate	0x00
0x083F	SYNCM[7][PDI_CTRL]	PDI Control	0x00
0x0900	RCV_TIME[PORT0]	Receive Time	0x00000000
0x0904	RCV_TIME[PORT1]	Receive Time	0x00000000
0x0908	RCV_TIME[PORT2]	Receive Time	0x00000000
0x090C	RCV_TIME[PORT3]	Receive Time	0x00000000
0x0910	SYS_TIME	System Time	0x0000000000000000
0x0918	RCVT_ECAT_PU	Receive Time ECAT Processing Unit	0x0000000000000000
0x0920	SYS_TIME_OFFSET	System Time Offset	0x0000000000000000
0x0928	SYS_TIME_DELAY	System Time Delay	0x00000000
0x092C	SYS_TIME_DIFF	System Time Difference	0x00000000
0x0930	SPD_CNT_START	Speed Counter Start	0x0000
0x0932	SPD_CNT_DIFF	Speed Counter Diff	0x0000
0x0934	SYS_TIME_DIFF_FD	System Time Difference Filter Depth	0x00
0x0935	SPD_CNT_FD	Speed Counter Filter Depth	0x00
0x0936	RCV_TIME_LM	Receive Time Latch Mode	0x00
0x0980	CYC_UNIT_CTRL	Cyclic Unit Control	0x00
0x0981	SYNCO_ACT	SYNC Out Unit Activation	0x00
0x0982	PULSE_LEN	Pulse Length of SyncSignals	0x0000
0x0984	ACT_STAT	Activation Status	0x00
0x098E	SYNCO_STAT	SYNCO Status	0x00
0x098F	SYNC1_STAT	SYNC1 Status	0x00
0x0990	START_TIME_CO	Start Time Cyclic Operation	0x0000000000000000

Offset	Register Name	Description	Reset Value
0x0998	NXT_SYNC1_PULSE	Next SYNC1 Pulse	0x0000000000000000
0x09A0	SYNC0_CYC_TIME	SYNC0 Cycle Time	0x00000000
0x09A4	SYNC1_CYC_TIME	SYNC1 Cycle Time	0x00000000
0x09A8	LATCH0_CTRL	Latch0 Control	0x00
0x09A9	LATCH1_CTRL	Latch1 Control	0x00
0x09AE	LATCH0_STAT	Latch0 Status	0x00
0x09AF	LATCH1_STAT	Latch1 Status	0x00
0x09B0	LATCH0_TIME_PE	Latch0 Time Positive Edge	0x0000000000000000
0x09B8	LATCH0_TIME_NE	Latch0 Time Negative Edge	0x0000000000000000
0x09C0	LATCH1_TIME_PE	Latch1 Time Positive Edge	0x0000000000000000
0x09C8	LATCH1_TIME_NE	Latch1 Time Negative Edge	0x0000000000000000
0x09F0	ECAT_BUF_CET	EtherCAT Buffer Change Event Time	0x00000000
0x09F8	PDI_BUF_SET	PDI Buffer Start Event Time	0x00000000
0x09FC	PDI_BUF_CET	PDI Buffer Change Event Time	0x00000000
0x0E00	PID	Product ID	0x0000000000000000
0x0E08	VID	Vendor ID	0x0000000000000000
0x0F00	DIO_OUT_DATA	Digital I/O Output Data	0x00000000
0x0F10	GPO	General Purpose Outputs	0x0000000000000000
0x0F18	GPI	General Purpose Inputs	0x0000000000000000
0x0F80	USER_RAM_BYTE0	User Ram Byte 0	0x00
0x0F81	USER_RAM_BYTE1	User Ram Byte 1	0x00
0x0F82	USER_RAM_BYTE2	User Ram Byte 2	0x00
0x0F83	USER_RAM_BYTE3	User Ram Byte 3	0x00
0x0F84	USER_RAM_BYTE4	User Ram Byte 4	0x00
0x0F85	USER_RAM_BYTE5	User Ram Byte 5	0x00
0x0F86	USER_RAM_BYTE6	User Ram Byte 6	0x00
0x0F87	USER_RAM_BYTE7	User Ram Byte 7	0x00
0x0F88	USER_RAM_BYTE8	User Ram Byte 8	0x00
0x0F89	USER_RAM_BYTE9	User Ram Byte 9	0x00
0x0F8A	USER_RAM_BYTE10	User Ram Byte 10	0x00
0x0F8B	USER_RAM_BYTE11	User Ram Byte 11	0x00
0x0F8E	USER_RAM_BYTE14	User Ram Byte 14	0x00
0x0F8F	USER_RAM_BYTE15	User Ram Byte 15	0x00
0x0F93	USER_RAM_BYTE19	User Ram Byte 19	0x00
0x1000	PDRAM	Process Data Ram	0x00000000
0x10000	PDRAM_ALS	Process Data Ram Alias	0x00000000
0x1F000	GPR_CFG0	General Purpose Configure 0	0x00001000
0x1F004	GPR_CFG1	General Purpose Configure 1	0x00000000
0x1F008	GPR_CFG2	General Purpose Configure 2	0x00000000

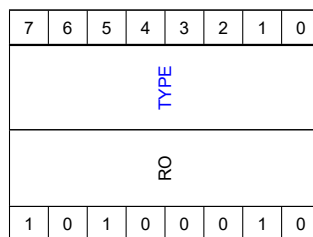
Offset	Register Name	Description	Reset Value
0x1F010	PHY_CFG0	PHY Configure 0	0x00000000
0x1F014	PHY_CFG1	PHY Configure 1	0x00000001
0x1F020	GPIO_CTRL	GPIO Output Enable	0x00000000
0x1F024	GPI_TRIG_CFG	GPI trigger config	0x00000000
0x1F028	GPO_TRIG_CFG	GPO trigger config	0x00000000
0x1F030	GPI_OVERRIDE0	GPI low word Override value	0x00000000
0x1F034	GPI_OVERRIDE1	GPI high word Override value	0x00000000
0x1F038	GPO_REG0	GPO low word read value	0x00000000
0x1F03C	GPO_REG1	GPO high word read value	0x00000000
0x1F040	GPI_REG0	GPI low word read value	0x00000000
0x1F044	GPI_REG1	GPI high word read value	0x00000000
0x1F060	GPR_STATUS	global status register	0x00000000
0x1F080	IO_CFG[CTR0]	CTR IO Configure	0x00000000
0x1F084	IO_CFG[CTR1]	CTR IO Configure	0x00000000
0x1F088	IO_CFG[CTR2]	CTR IO Configure	0x00000000
0x1F08C	IO_CFG[CTR3]	CTR IO Configure	0x00000000
0x1F090	IO_CFG[CTR4]	CTR IO Configure	0x00000000
0x1F094	IO_CFG[CTR5]	CTR IO Configure	0x00000000
0x1F098	IO_CFG[CTR6]	CTR IO Configure	0x00000000
0x1F09C	IO_CFG[CTR7]	CTR IO Configure	0x00000000
0x1F0A0	IO_CFG[CTR8]	CTR IO Configure	0x00000000

Table 227: ESC Register List

57.4 ESC register details

ESC registers is as follows:

57.4.1 TYPE (0x0)



TYPE [7:0]

Bits	Name	Description
7-0	TYPE	Controller type

TYPE Bits

57.4.2 REVISION (0x1)

7	6	5	4	3	2	1	0
X							
RO							
0	0	0	0	0	0	0	0

REVISION [7:0]

Bits	Name	Description
7-0	X	major version X

REVISION Bits

57.4.3 BUILD (0x2)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BUILD								Y				Z			
RO								RO				RO			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BUILD [15:0]

Bits	Name	Description
15-8	BUILD	
7-4	Y	minor version Y
3-0	Z	maintenance version Z

BUILD Bits

57.4.4 FMMU_NUM (0x4)

7	6	5	4	3	2	1	0
NUM							
RO							
0	0	0	0	1	0	0	0

FMMU_NUM [7:0]

Bits	Name	Description
7-0	NUM	Number of supported FMMU channels (or entities)

Bits	Name	Description

FMMU_NUM Bits

57.4.5 SYNCM_NUM (0x5)

7	6	5	4	3	2	1	0
NUM							
RO							
0	0	0	0	1	0	0	0

SYNCM_NUM [7:0]

Bits	Name	Description
7-0	NUM	Number of supported SyncManager channels (or entities)

SYNCM_NUM Bits

57.4.6 RAM_SIZE (0x6)

7	6	5	4	3	2	1	0
SIZE							
RO							
0	0	1	1	1	1	0	0

RAM_SIZE [7:0]

Bits	Name	Description
7-0	SIZE	Process Data RAM size supported in KByte

RAM_SIZE Bits

57.4.7 PORT_DESC (0x7)

7	6	5	4	3	2	1	0
PORT3		PORT2		PORT1		PORT0	
RO		RO		RO		RO	
0	0	1	1	1	1	1	1

PORT_DESC [7:0]

Bits	Name	Description
7-6	PORT3	Port configuration: 00:Not implemented 01:Not configured (SII EEPROM) 10:EBUS 11:MII/RMII/RGMII
5-4	PORT2	Port configuration: 00:Not implemented 01:Not configured (SII EEPROM) 10:EBUS 11:MII/RMII/RGMII
3-2	PORT1	Port configuration: 00:Not implemented 01:Not configured (SII EEPROM) 10:EBUS 11:MII/RMII/RGMII
1-0	PORT0	Port configuration: 00:Not implemented 01:Not configured (SII EEPROM) 10:EBUS 11:MII/RMII/RGMII

PORT_DESC Bits

57.4.8 FEATURE (0x8)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD				FFSC	RWC	LRW	EDSA	SHFE	ELDM	RSVD		DCW	DC	RSVD	FMMU
N/A				RO	RO	RO	RO	RO	RO	N/A		RO	RO	N/A	RO
x	x	x	x	0	0	0	0	1	0	x	x	1	1	x	0

FEATURE [15:0]

Bits	Name	Description
11	FFSC	Fixed FMMU/SyncManager configuration: 0:Variable configuration 1:Fixed configuration (refer to documentation of supporting ESCs)
10	RWC	EtherCAT read/write command support(BRW,APRW,FPRW): 0:Supported 1:Not supported

Bits	Name	Description
9	LRW	EtherCAT LRW command support: 0:Supported 1:Not supported
8	EDSA	Enhanced DC SYNC Activation: 0:Not available 1:Available Note:This feature refers to registers 0x981[7:3] and 0x0984
7	SHFE	Seperate Handling of FCS Errors: 0:Not supported 1:Supported, frames with wrong FCS and additional nibble will be counted separately in Forwarded RX Error Counter
6	ELDM	Enhanced Link Detection MII: 0:Not available 1:Available
3	DCW	Distributed Clocks width: 0:32 bit 1:64 bit
2	DC	Distributed Clocks: 0:Not available 1:Available
0	FMMU	FMMU Operation: 0:Bit oriented 1:Byte oriented

FEATURE Bits

57.4.9 STATION_ADDR (0x10)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR															
R0															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

STATION_ADDR [15:0]

Bits	Name	Description
15-0	ADDR	Address used for node addressing (FPRD/FPWR/FPRW/FRMW commands)

STATION_ADDR Bits

57.4.10 STATION_ALS (0x12)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR															
RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

STATION_ALS [15:0]

Bits	Name	Description
15-0	ADDR	Alias Address used for node addressing (FPRD/FPWR/FPRW/FRMW commands). The use of this alias is activated by Register DL Control Bit 0x0100[24]. NOTE:EEPROM value is only transferred into this register at first EEPROM load after power-on or reset. ESC20 exception:EEPROM value is transferred into this register after each EEPROM reload command.

STATION_ALS Bits

57.4.11 REG_WEN (0x20)

7	6	5	4	3	2	1	0
RSVD							EN
N/A							RO
x	x	x	x	x	x	x	0

REG_WEN [7:0]

Bits	Name	Description
0	EN	If register write protection is enabled, this register has to be written in the same Ethernet frame (value does not matter) before other writes to this station are allowed. This bit is self-clearing at the beginning of the next frame (SOF), or if Register Write Protection is disabled.

REG_WEN Bits

57.4.12 REG_WP (0x21)

7	6	5	4	3	2	1	0
RSVD							WP
N/A							RO
x	x	x	x	x	x	x	0

REG_WP [7:0]

Bits	Name	Description
0	WP	Register write protection: 0:Protection disabled 1:Protection enabled Registers 0x0000:0x0F7F are write-protected, except for 0x0020 and 0x0030

REG_WP Bits

57.4.13 ESC_WEN (0x30)

7	6	5	4	3	2	1	0
RSVD							EN
N/A							RO
x	x	x	x	x	x	x	0

ESC_WEN [7:0]

Bits	Name	Description
0	EN	If ESC write protection is enabled, this register has to be written in the same Ethernet frame (value does not matter) before other writes to this station are allowed. This bit is self-clearing at the beginning of the next frame (SOF), or if ESC Write Protection is disabled.

ESC_WEN Bits

57.4.14 ESC_WP (0x31)

7	6	5	4	3	2	1	0
RSVD							WP

7	6	5	4	3	2	1	0
N/A							RO
x	x	x	x	x	x	x	0

ESC_WP [7:0]

Bits	Name	Description
0	WP	Write protect: 0:Protection disabled 1:Protection enabled All areas are write-protected, except for 0x0030.

ESC_WP Bits

57.4.15 ESC_RST_ECAT (0x40)

7	6	5	4	3	2	1	0
RSVD						PR	
N/A						RO	
x	x	x	x	x	x	0	0

ESC_RST_ECAT [7:0]

Bits	Name	Description
1-0	PR	Progress of the reset procedure: 00:initial/reset state 01:after writing 0x52 ('R'), when previous state was 00 10:after writing 0x45 ('E'), when previous state was 01 11:after writing 0x53 ('S'), when previous state was 10. This value must not be observed because the ESC enters reset when this state is reached, resulting in state 00

ESC_RST_ECAT Bits

57.4.16 ESC_RST_PDI (0x41)

7	6	5	4	3	2	1	0
RST							

7	6	5	4	3	2	1	0
RW							
0	0	0	0	0	0	0	0

ESC_RST_PDI [7:0]

Bits	Name	Description
7-0	RST	A reset is asserted after writing the reset sequence 0x52 ('R'), 0x45 ('E') and 0x53 ('S') in this register with 3 consecutive commands. Any other command which does not continue the sequence by writing the next expected value will cancel the reset procedure

ESC_RST_PDI Bits

57.4.17 ESC_DL_CTRL (0x100)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RSVD							SA	RSVD				RFS			LP3		LP2		LP1		LP0		RSVD							TU	FR	
N/A							RO	N/A				RO			RO		RO		RO		RO		N/A							RO	RO	
x	x	x	x	x	x	x	0	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	x	x	x	x	x	x	0	0

ESC_DL_CTRL [31:0]

Bits	Name	Description
24	SA	Station alias: 0:Ignore Station Alias 1:Alias can be used for all configured address comm

Bits	Name	Description
18-16	RFS	RX FIFO Size (ESC delays start of forwarding until FIFO is at least half full). RX FIFO Size/RX delay reduction** : Value:EBUS:MII: 0:-50 ns -40 ns (-80 ns***) 1:-40 ns -40 ns (-80 ns***) 2:-30 ns -40 ns 3:-20 ns -40 ns 4:-10 ns no change 5:no change no change 6:no change no change 7:default default NOTE:EEPROM value is only taken over at first EEPROM load after power-on or reset
15-14	LP3	Loop Port 3: 00:Auto 01:Auto Close 10:Open 11:Closed
13-12	LP2	Loop Port 2: 00:Auto 01:Auto Close 10:Open 11:Closed
11-10	LP1	Loop Port 1: 00:Auto 01:Auto Close 10:Open 11:Closed

Bits	Name	Description
9-8	LP0	Loop Port 0: 00:Auto 01:Auto Close 10:Open 11:Closed NOTE: Loop open means sending/receiving over this port is enabled, loop closed means sending/receiving is disabled and frames are forwarded to the next open port internally. Auto:loop closed at link down, opened at link up Auto Close:loop closed at link down, opened with writing 01 again after link up (or receiving a valid Ethernet frame at the closed port) Open:loop open regardless of link state Closed:loop closed regardless of link state
1	TU	Temporary use of settings in 0x0100:0x0103[8:15]: 0:permanent use 1:use for about 1 second, then revert to previous settings
0	FR	Forwarding rule: 0:Forward non-EtherCAT frames: EtherCAT frames are processed, non-EtherCAT frames are forwarded without processing or modification. The source MAC address is not changed for any frame. 1:Destroy non-EtherCAT frames: EtherCAT frames are processed, non-EtherCAT frames are destroyed. The source MAC address is changed by the Processing Unit for every frame (SOURCE_MAC[1] is set

ESC_DL_CTRL Bits

57.4.18 PHYSICAL_RW_OFFSET (0x108)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PHYSICAL_RW_OFFSET [15:0]

Bits	Name	Description
15-0	OFFSET	This register is used for ReadWrite commands in Device Addressing mode (FPRW, APRW, BRW). The internal read address is directly taken from the offset address field of the EtherCAT datagram header, while the internal write address is calculated by adding the Physical Read/Write Offset value to the offset address field. Internal read address = ADR, internal write address = ADR + R/W-Offset

PHYSICAL_RW_OFFSET Bits

57.4.19 ESC_DL_STAT (0x110)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CP3	LP3	CP2	LP2	CP1	LP1	CP0	LP0	PLP3	PLP2	PLP1	PLP0	RSVD	ELD	WDS	EPLC
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	N/A	RO	RO	RO
0	0	0	0	0	0	0	0	0	0	0	0	x	0	0	0

ESC_DL_STAT [15:0]

Bits	Name	Description
15	CP3	Communication on Port 3: 0:No stable communication 1:Communication established
14	LP3	Loop Port 3: 0:Open 1:Closed
13	CP2	Communication on Port 2: 0:No stable communication 1:Communication established

Bits	Name	Description
12	LP2	Loop Port 2: 0:Open 1:Closed
11	CP1	Communication on Port 1: 0:No stable communication 1:Communication established
10	LP1	Loop Port 1: 0:Open 1:Closed
9	CP0	Communication on Port 0: 0:No stable communication 1:Communication established
8	LP0	Loop Port 0: 0:Open 1:Closed
7	PLP3	Physical link on Port 3: 0:No link 1:Link detected
6	PLP2	Physical link on Port 2: 0:No link 1:Link detected
5	PLP1	Physical link on Port 1: 0:No link 1:Link detected
4	PLP0	Physical link on Port 0: 0:No link 1:Link detected
2	ELD	Enhanced Link detection: 0:Deactivated for all ports 1:Activated for at least one port NOTE:EEPROM value is only transferred into this register at first EEPROM load after power-on or reset
1	WDS	PDI Watchdog Status: 0:Watchdog expired 1:Watchdog reloaded

Bits	Name	Description
0	EPLC	PDI operational/EEPROM loaded correctly: 0:EEPROM not loaded, PDI not operational (no access to Process Data RAM) 1:EEPROM loaded correctly, PDI operational (access to Process Data RAM)

ESC_DL_STAT Bits

57.4.20 AL_CTRL (0x120)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD										DI	EIA	IST			
N/A										RW	RW	RW			
x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	1

AL_CTRL [15:0]

Bits	Name	Description
5	DI	Device Identification: 0:No request 1:Device Identification request
4	EIA	Error Ind Ack: 0:No Ack of Error Ind in AL status register 1:Ack of Error Ind in AL status register
3-0	IST	Initiate State Transition of the Device State Machine: 1:Request Init State 3:Request Bootstrap State 2:Request Pre-Operational State 4:Request Safe-Operational State 8:Request Operational State

AL_CTRL Bits

57.4.21 AL_STAT (0x130)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD										DI	EI	AS			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
N/A										RW	RW	RW			
x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	1

AL_STAT [15:0]

Bits	Name	Description
5	DI	Device Identification: 0:Device Identification not valid 1:Device Identification loaded
4	EI	Error Ind: 0:Device is in State as requested or Flag cleared by command 1:Device has not entered requested State or changed State as result of a local action
3-0	AS	Actual State of the Device State Machine: 1:Init State 3:Bootstrap State 2:Pre-Operational State 4:Safe-Operational State 8:Operational State

AL_STAT Bits

57.4.22 AL_STAT_CODE (0x134)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CODE										RW					
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

AL_STAT_CODE [15:0]

Bits	Name	Description
15-0	CODE	AL Status Code

AL_STAT_CODE Bits

57.4.23 RUN_LED_OVRD (0x138)

7	6	5	4	3	2	1	0
RSVD			EN_OVRD	LED_CODE			
N/A			RW	RW			
x	x	x	0	0	0	0	0

RUN_LED_OVRD [7:0]

Bits	Name	Description
4	EN_OVRD	Enable Override: 0:Override disabled 1:Override enabled
3-0	LED_CODE	LED code: 0x0:Off 0x1:Flash 1x 0x2-0xC:Flash 2x -12x 0xD:Blinking 0xE:Flickering 0xF:On

RUN_LED_OVRD Bits

57.4.24 ERR_LED_OVRD (0x139)

7	6	5	4	3	2	1	0
RSVD			EN_OVRD	LED_CODE			
N/A			RW	RW			
x	x	x	0	0	0	0	0

ERR_LED_OVRD [7:0]

Bits	Name	Description
4	EN_OVRD	Enable Override: 0:Override disabled 1:Override enabled
3-0	LED_CODE	LED code: 0x0:Off 0x1-0xC:Flash 1x -12x 0xD:Blinking 0xE:Flickering 0xF:On

Bits	Name	Description

ERR_LED_OVRD Bits

57.4.25 PDI_CTRL (0x140)

7	6	5	4	3	2	1	0
PDI							
PDI							
0	0	0	0	0	0	0	0

PDI_CTRL [7:0]

Bits	Name	Description
7-0	PDI	Process data interface: 0x00:Interface deactivated (no PDI) 0x01:4 Digital Input 0x02:4 Digital Output 0x03:2 Digital Input and 2 Digital Output 0x04:Digital I/O 0x05:SPI Slave 0x06:Oversampling I/O 0x07:EtherCAT Bridge (port 3) 0x08:16 Bit asynchronous Microcontroller interface 0x09:8 Bit asynchronous Microcontroller interface 0x0A:16 Bit synchronous Microcontroller interface 0x0B:8 Bit synchronous Microcontroller interface 0x10:32 Digital Input and 0 Digital Output 0x11:24 Digital Input and 8 Digital Output 0x12:16 Digital Input and 16 Digital Output 0x13:8 Digital Input and 24 Digital Output 0x14:0 Digital Input and 32 Digital Output 0x80:On-chip bus Others:Reserved

PDI_CTRL Bits

57.4.26 ESC_CFG (0x141)

7	6	5	4	3	2	1	0
ELP3	ELP2	ELP1	ELP0	CDLIU	DCSOU	ELDAP	DEV_EMU
RO	RO	RO	RO	RO	RO	RO	RO
0	0	0	0	0	0	0	1

ESC_CFG [7:0]

Bits	Name	Description
7	ELP3	Enhanced Link port 3: 0:disabled (if bit 1=0) 1:enabled
6	ELP2	Enhanced Link port 2: 0:disabled (if bit 1=0) 1:enabled
5	ELP1	Enhanced Link port 1: 0:disabled (if bit 1=0) 1:enabled
4	ELP0	Enhanced Link port 0: 0:disabled (if bit 1=0) 1:enabled
3	CDLIU	Distributed Clocks Latch In Unit: 0:disabled (power saving) 1:enabled
2	DCSOU	Distributed Clocks SYNC Out Unit: 0:disabled (power saving) 1:enabled
1	ELDAP	Enhanced Link detection all ports: 0:disabled (if bits [7:4]=0) 1:enabled at all ports (overrides bits [7:4])
0	DEV_EMU	Device emulation (control of AL status): 0:AL status register has to be set by PDI 1:AL status register will be set to value written to AL control register

ESC_CFG Bits

57.4.27 PDI_INFO (0x14E)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD												PDICN	PDIA	ECLFE	PFABW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
N/A												RO	RO	RO	RO
x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

PDI_INFO [15:0]

Bits	Name	Description
3	PDICN	PDI configuration invalid: 0:PDI configuration ok 1:PDI configuration invalid
2	PDIA	PDI active: 0:PDI not active 1:PDI active
1	ECLFE	ESC configuration area loaded from EEPROM: 0:not loaded 1:loaded
0	PFABW	DI function acknowledge by write: 0:Disabled 1:Enabled

PDI_INFO Bits

57.4.28 PDI_CFG (0x150)

7	6	5	4	3	2	1	0
BUS			CLK				
RO			RO				
1	0	0	0	0	1	0	0

PDI_CFG [7:0]

Bits	Name	Description
7-5	BUS	On-chip bus: 000:Intel® Avalon® 001:AXI® 010:Xilinx® PLB v4.6 100:Xilinx OPB others:reserved

Bits	Name	Description
4-0	CLK	On-chip bus clock: 0:asynchronous 1-31:synchronous multiplication factor (N * 25 MHz)

PDI_CFG Bits

57.4.29 PDI_SL_CFG (0x151)

7	6	5	4	3	2	1	0
SYNC1_MAER	SYNC1_CFG	SYNC1_ODP	SYNC0_MAER	SYNC0_CFG	SYNC0_ODP		
RO	RO	RO	RO	RO	RO		
0	0	0	0	0	0	0	0

PDI_SL_CFG [7:0]

Bits	Name	Description
7	SYNC1_MAER	SYNC1 mapped to AL Event Request register 0x0220[3]: 0:Disabled 1:Enabled
6	SYNC1_CFG	SYNC1/LATCH1 configuration*: 0:LATCH1 input 1:SYNC1 output
5-4	SYNC1_ODP	SYNC1 output driver/polarity: 00:Push-Pull active low 01:Open Drain (active low) 10:Push-Pull active high 11:Open Source (active high)
3	SYNC0_MAER	SYNC0 mapped to AL Event Request register 0x0220[2]: 0:Disabled 1:Enabled
2	SYNC0_CFG	SYNC0/LATCH0 configuration*: 0:LATCH0 Input 1:SYNC0 Output

Bits	Name	Description
1-0	SYNC0_ODP	SYNC0 output driver/polarity: 00:Push-Pull active low 01:Open Drain (active low) 10:Push-Pull active high 11:Open Source (active high)

PDI_SL_CFG Bits

57.4.30 PDI_EXT_CFG (0x152)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD					OCBST			RSVD					RPS		
N/A					RW			N/A					RO		
x	x	x	x	x	0	0	0	x	x	x	x	x	x	0	0

PDI_EXT_CFG [15:0]

Bits	Name	Description
10-8	OCBST	On-chip bus sub-type for AXI: 000:AXI3 001:AXI4 010:AXI4 LITE others:reserved
1-0	RPS	Read prefetch size (in cycles of PDI width): 0:4 cycles 1:1 cycle (typical) 2:2 cycles 3:Reserved

PDI_EXT_CFG Bits

57.4.31 ECAT_EVT_MSK (0x200)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MASK															
RO															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ECAT_EVT_MSK [15:0]

Bits	Name	Description
15-0	MASK	ECAT Event masking of the ECAT Event Request Events for mapping into ECAT event field of EtherCAT frames: 0:Corresponding ECAT Event Request register bit is not mapped 1:Corresponding ECAT Event Request register bit is mapped

ECAT_EVT_MSK Bits

57.4.32 PDI_AL_EVT_MSK (0x204)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MASK																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

PDI_AL_EVT_MSK [31:0]

Bits	Name	Description
31-0	MASK	AL Event masking of the AL Event Request register Events for mapping to PDI IRQ signal: 0:Corresponding AL Event Request register bit is not mapped 1:Corresponding AL Event Request register bit is mapped

PDI_AL_EVT_MSK Bits

57.4.33 ECAT_EVT_REQ (0x210)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD				MV								ALS_EVT	DLS_EVT	RSVD	DCL_EVT
N/A				RO								RO	RO	N/A	RO
x	x	x	x	0	0	0	0	0	0	0	0	0	0	x	0

ECAT_EVT_REQ [15:0]

Bits	Name	Description
11-4	MV	Mirrors values of each SyncManager Status: 0:No Sync Channel 0 event 1:Sync Channel 0 event pending 0:No Sync Channel 1 event 1:Sync Channel 1 event pending ... 0:No Sync Channel 7 event 1:Sync Channel 7 event pending
3	ALS_EVT	AL Status event: 0:No change in AL Status 1:AL Status change (Bit is cleared by reading out AL Status 0x0130:0x0131 from ECAT)
2	DLS_EVT	DL Status event: 0:No change in DL Status 1:DL Status change (Bit is cleared by reading out DL Status 0x0110:0x0111 from ECAT)
0	DCL_EVT	DC Latch event: 0:No change on DC Latch Inputs 1:At least one change on DC Latch Inputs (Bit is cleared by reading DC Latch event times from ECAT for ECAT-controlled Latch Units, so that Latch 0/1 Status 0x09AE:0x09AF indicates no event)

ECAT_EVT_REQ Bits

57.4.34 AL_EVT_REQ (0x220)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								SM_INT																RSVD	WDG_PD	EE_EMU	SM_ACT	ST_DC_SYNC1	ST_DC_SYNC0	DCL_EVT	ALC_EVT
N/A								RO																N/A	RO	RO	RO	RO	RO	RO	RO
x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	0	0	0	0	0	0	0

AL_EVT_REQ [31:0]

Bits	Name	Description
23-8	SM_INT	SyncManager interrupts (SyncManager register offset 0x5, bit [0] or [1]): 0:No SyncManager 0 interrupt 1:SyncManager 0 interrupt pending 0:No SyncManager 1 interrupt 1:SyncManager 1 interrupt pending ... 0:No SyncManager 15 interrupt 1:SyncManager 15 interrupt pending
6	WDG_PD	Watchdog Process Data: 0:Has not expired 1:Has expired (Bit is cleared by reading Watchdog Status Process Data 0x0440 from PDI)
5	EE_EMU	EEPROM Emulation: 0:No command pending 1:EEPROM command pending (Bit is cleared by acknowledging the command in EEPROM Control/Status register 0x0502:0x0503[10:8] from PDI)
4	SM_ACT	SyncManager activation register (SyncManager register offset 0x6) changed: 0:No change in any SyncManager 1:At least one SyncManager changed (Bit is cleared by reading SyncManager Activation registers 0x0806 etc. from PDI)
3	ST_DC_SYNC1	State of DC SYNC1 (if register 0x0151[7]=1): (Bit is cleared by reading of SYNC1 status 0x098F from PDI, use only in Acknowledge mode)
2	ST_DC_SYNC0	State of DC SYNC0 (if register 0x0151[3]=1): (Bit is cleared by reading SYNC0 status 0x098E from PDI, use only in Acknowledge mode)

Bits	Name	Description
1	DCL_EVT	DC Latch event: 0:No change on DC Latch Inputs 1:At least one change on DC Latch Inputs (Bit is cleared by reading DC Latch event times from PDI, so that Latch 0/1 Status 0x09AE:0x09AF indicates no event. Available if Latch Unit is PDI-controlled)
0	ALC_EVT	AL Control event: 0:No AL Control Register change 1:AL Control Register has been written3 (Bit is cleared by reading AL Control register 0x0120:0x0121 from PDI)

AL_EVT_REQ Bits

57.4.35 RX_ERR_CNT (0x300 + 0x2 * n)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_ERR								IVD_FRM							
RO								RO							
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RX_ERR_CNT [15:0]

Bits	Name	Description
15-8	RX_ERR	RX Error counter of Port y (counting is stopped when 0xFF is reached).
7-0	IVD_FRM	Invalid frame counter of Port y (counting is stopped when 0xFF is reached).

RX_ERR_CNT Bits

57.4.36 FWD_RX_ERR_CNT (0x308 + 0x1 * n)

7	6	5	4	3	2	1	0
ERR_CNT							
RO							
0	0	0	0	0	0	0	0

FWD_RX_ERR_CNT [7:0]

Bits	Name	Description
7-0	ERR_CNT	Forwarded error counter of Port y (counting is stopped when 0xFF is reached).

FWD_RX_ERR_CNT Bits

57.4.37 ECAT_PU_ERR_CNT (0x30C)

7	6	5	4	3	2	1	0
CNT							
RO							
0	0	0	0	0	0	0	0

ECAT_PU_ERR_CNT [7:0]

Bits	Name	Description
7-0	CNT	ECAT Processing Unit error counter (counting is stopped when 0xFF is reached). Counts errors of frames passing the Processing Unit.

ECAT_PU_ERR_CNT Bits

57.4.38 PDI_ERR_CNT (0x30D)

7	6	5	4	3	2	1	0
CNT							
RO							
0	0	0	0	0	0	0	0

PDI_ERR_CNT [7:0]

Bits	Name	Description
7-0	CNT	PDI Error counter (counting is stopped when 0xFF is reached). Counts if a PDI access has an interface error.

PDI_ERR_CNT Bits

57.4.39 LOST_LINK_CNT (0x310 + 0x1 * n)

7	6	5	4	3	2	1	0
CNT							
RO							
0	0	0	0	0	0	0	0

LOST_LINK_CNT [7:0]

Bits	Name	Description
7-0	CNT	Lost Link counter of Port y (counting is stopped when 0xff is reached). Counts only if port is open and loop is Auto.

LOST_LINK_CNT Bits

57.4.40 WDG_DIV (0x400)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIV															
RO															
0	0	0	0	1	0	0	1	1	1	0	0	0	0	1	0

WDG_DIV [15:0]

Bits	Name	Description
15-0	DIV	Watchdog divider: Number of 25 MHz tics (minus 2) that represent the basic watchdog increment. (Default value is 100μs = 2498)

WDG_DIV Bits

57.4.41 WDG_TIME_PDI (0x410)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIME															
RO															
0	0	0	0	0	0	1	1	1	1	1	0	1	0	0	0

WDG_TIME_PDI [15:0]

Bits	Name	Description
15-0	TIME	Watchdog Time PDI:number of basic watchdog increments (Default value with Watchdog divider 100μs means 100ms Watchdog)

WDG_TIME_PDI Bits

57.4.42 WDG_TIME_PDAT (0x420)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIME															
RO															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

WDG_TIME_PDAT [15:0]

Bits	Name	Description
15-0	TIME	Watchdog Time Process Data:number of basic watchdog increments (Default value with Watchdog divider 100μs means 100ms Watchdog)

WDG_TIME_PDAT Bits

57.4.43 WDG_STAT_PDAT (0x440)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD															ST
N/A															RW
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0

WDG_STAT_PDAT [15:0]

Bits	Name	Description
0	ST	Watchdog Status of Process Data (triggered by SyncManagers) 0:Watchdog Process Data expired 1:Watchdog Process Data is active or disabled

WDG_STAT_PDAT Bits

57.4.44 WDG_CNT_PDAT (0x442)

7	6	5	4	3	2	1	0
CNT							
RO							
0	0	0	0	0	0	0	0

WDG_CNT_PDAT [7:0]

Bits	Name	Description
7-0	CNT	Watchdog Counter Process Data (counting is stopped when 0xFF is reached). Counts if Process Data Watchdog expires.

WDG_CNT_PDAT Bits

57.4.45 WDG_CNT_PDI (0x443)

7	6	5	4	3	2	1	0
CNT							
RO							
0	0	0	0	0	0	0	0

WDG_CNT_PDI [7:0]

Bits	Name	Description
7-0	CNT	Watchdog PDI counter (counting is stopped when 0xFF is reached). Counts if PDI Watchdog expires.

WDG_CNT_PDI Bits

57.4.46 EEPROM_CFG (0x500)

7	6	5	4	3	2	1	0
RSVD						FORCE_EC	PDI
N/A						RO	RO
x	x	x	x	x	x	0	0

EEPROM_CFG [7:0]

Bits	Name	Description
1	FORCE_ECAT	Force ECAT access: 0:Do not change Bit 0x0501[0] 1:Reset Bit 0x0501[0] to 0
0	PDI	EEPROM control is offered to PDI: 0:no 1:yes (PDI has EEPROM control)

EEPROM_CFG Bits

57.4.47 EEPROM_PDI_ACC_STAT (0x501)

7	6	5	4	3	2	1	0
RSVD							ACCESS
N/A							RW
x	x	x	x	x	x	x	0

EEPROM_PDI_ACC_STAT [7:0]

Bits	Name	Description
0	ACCESS	Access to EEPROM: 0:PDI releases EEPROM access 1:PDI takes EEPROM access (PDI has EEPROM control)

EEPROM_PDI_ACC_STAT Bits

57.4.48 EEPROM_CTRL_STAT (0x502)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BUSY	ERR_WEN	ERR_ACK_CMD	EE_LDS	CKSM_ERR	CMD			EE_ALGM	NUM_RD_BYTE	EE_EMU	RSVD				ECAT_WEN
RO	RO	RW	RO	RW	RW			RO	RO	RO	N/A				RO
0	0	0	0	0	0	0	0	0	0	0	x	x	x	x	0

EEPROM_CTRL_STAT [15:0]

Bits	Name	Description
15	BUSY	Busy: 0:EEPROM Interface is idle 1:EEPROM Interface is busy

Bits	Name	Description
14	ERR_WEN	Error Write Enable*3 : 0:No error 1:Write Command without Write enable
13	ERR_ACK_CMD	Error Acknowledge/Command*3 : 0:No error 1:Missing EEPROM acknowledge or invalid command EEPROM emulation only:PDI writes 1 if a temporary failure has occurred.
12	EE_LDS	EEPROM loading status: 0:EEPROM loaded, device information ok 1:EEPROM not loaded, device information not available (EEPROM loading in progress or finished with a failure)
11	CKSM_ERR	Checksum Error in ESC Configuration Area: 0:Checksum ok 1:Checksum error EEPROM emulation for IP Core only:PDI writes 1 if a CRC failure has occurred for a reload command.
10-8	CMD	Command register*2: Write:Initiate command. Read:Currently executed command Commands: 000:No command/EEPROM idle (clear error bits) 001:Read 010:Write 100:Reload Others:Reserved/invalid commands (do not issue) EEPROM emulation only:after execution, PDI writes command value to indicate operation is ready.
7	EE_ALGM	Selected EEPROM Algorithm: 0:1 address byte (1Kbit -16Kbit EEPROMs) 1:2 address bytes (32Kbit -4 Mbit EEPROMs)
6	NUM_RD_BYTE	Supported number of EEPROM read bytes: 0:4 Bytes 1:8 Bytes
5	EE_EMU	EPROM emulation: 0:Normal operation (I ² C interface used) 1:PDI emulates EEPROM (I ² C not used)

Bits	Name	Description
0	ECAT_WEN	ECAT write enable*2 : 0:Write requests are disabled 1:Write requests are enabled This bit is always 1 if PDI has EEPROM control.

EEPROM_CTRL_STAT Bits

57.4.49 EEPROM_ADDR (0x504)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EEPROM_ADDR [31:0]

Bits	Name	Description
31-0	ADDR	EEPROM Address 0:First word (= 16 bit) 1:Second word ... Actually used EEPROM Address bits: 9-0: EEPROM size up to 16 Kbit 17-0: EEPROM size 32 Kbit –4 Mbit 31-0: EEPROM Emulation

EEPROM_ADDR Bits

57.4.50 EEPROM_DATA (0x508)

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
H																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EEPROM_DATA [63:32]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
H																L															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RW																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EEPROM_DATA [31:0]

Bits	Name	Description
63-16	HI	EEPROM Read data (data read from EEPROM, higher bytes)
15-0	LO	EEPROM Write data (data to be written to EEPROM) or EEPROM Read data (data read from EEPROM, lower bytes)

EEPROM_DATA Bits

57.4.51 MII_MNG_CS (0x510)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BUSY	CMD_ERR	RD_ERR	RSVD			CMD		PHY_ADDR					LINK_DC	PDI	WEN
RO	RO	RO	N/A			RW		RO					RO	RO	RO
0	0	0	x	x	x	0	0	0	0	0	0	0	0	0	0

MIIMNGCS [15:0]

Bits	Name	Description
15	BUSY	Busy: 0:MIIM Management Interface is idle 1:MIIM Management Interface is busy
14	CMD_ERR	Command error: 0:Last Command was successful 1:Invalid command or write command without Write Enable Cleared by executing a valid command or by writing “00” to Command register bits [9:8].
13	RD_ERR	Read error: 0:No read error 1:Read error occurred (PHY or register not available) Cleared by writing to register 0x0511

Bits	Name	Description
9-8	CMD	Command register*: Write:Initiate command. Read:Currently executed command 00:No command/MI idle (clear error bits) 01:Read 10:Write Others:Reserved/invalid command (do not issue)
7-3	PHY_ADDR	PHY address of port 0 (this is equal to the PHY address offset, if the PHY addresses are consecutive) IP Core since V3.0.0/3.00c: Translation 0x0512[7]=0: Register 0x0510[7:3] shows PHY address of port 0 Translation 0x0512[7]=1: Register 0x0510[7:3] shows the PHY address which will be used for port 0-3 as requested by 0x0512[4:0] (valid values 0-3)
2	LINK_DC	MI link detection and configuration: 0:Disabled for all ports 1:Enabled for at least one MII port, refer to PHY Port Status (0x0518 ff.) for details
1	PDI	Management Interface can be controlled by PDI (registers 0x0516-0x0517): 0:Only ECAT control 1:PDI control possible
0	WEN	Write enable*: 0:Write disabled 1:Write enabled This bit is always 1 if PDI has MI control. ET1100-0000/-0001 exception: Bit is not always 1 if PDI has MI control, and bit is writable by PDI.

MII_MNG_CS Bits

57.4.52 PHY_ADDR (0x512)

7	6	5	4	3	2	1	0
SHOW		RSVD		ADDR			
RW		N/A		RW			
0	x	x	0	0	0	0	0

PHY_ADDR [7:0]

Bits	Name	Description
7	SHOW	Target PHY Address translation: 0:Enabled 1:Disabled Refer to 0x0512[4:0] and 0x0510[7:3] for details.
4-0	ADDR	Target PHY Address Translation 0x0512[7]=0: 0-3:Target PHY Addresses 0-3 are used to access the PHYs at port 0-3, when the PHY addresses are properly configured 4-31:The configured PHY address of port 0 (PHY address offset) is added to the Target PHY Address values 4-31 when accessing a PHY Translation 0x0512[7]=1: 0-31:Target PHY Addresses is used when accessing a PHY without translation

PHY_ADDR Bits

57.4.53 PHY_REG_ADDR (0x513)

7	6	5	4	3	2	1	0
RSVD		ADDR					
N/A		RW					
x	x	x	0	0	0	0	0

PHY_REG_ADDR [7:0]

Bits	Name	Description
4-0	ADDR	Address of PHY Register that shall be read/written

Bits	Name	Description

PHY_REG_ADDR Bits

57.4.54 PHY_DATA (0x514)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA															
RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PHY_DATA [15:0]

Bits	Name	Description
15-0	DATA	PHY Read/Write Data

PHY_DATA Bits

57.4.55 MIIM_ECAT_ACC_STAT (0x516)

7	6	5	4	3	2	1	0
RSVD							ACC
N/A							RO
x	x	x	x	x	x	x	0

MIIM_ECAT_ACC_STAT [7:0]

Bits	Name	Description
0	ACC	Access to MII management: 0:ECAT enables PDI takeover of MII management interface 1:ECAT claims exclusive access to MII management interface

MIIM_ECAT_ACC_STAT Bits

57.4.56 MIIM_PDI_ACC_STAT (0x517)

7	6	5	4	3	2	1	0
RSVD						FORCE	ACC
N/A						RO	RW

7	6	5	4	3	2	1	0
x	x	x	x	x	x	0	0

MIIM_PDI_ACC_STAT [7:0]

Bits	Name	Description
1	FORCE	Force PDI Access State: 0:Do not change Bit 0x0517[0] 1:Reset Bit 0x0517[0] to 0
0	ACC	Access to MII management: 0:ECAT has access to MII management 1:PDI has access to MII management

MIIM_PDI_ACC_STAT Bits

57.4.57 PHY_STAT (0x518 + 0x1 * n)

7	6	5	4	3	2	1	0
RSVD	PCU	LPE	RE	LSE	LS	PLS	
N/A	RW	RO	RW	RO	RO	RO	RO
x	x	0	0	0	0	0	0

PHY_STAT [7:0]

Bits	Name	Description
5	PCU	PHY configuration updated: 0:No update 1:PHY configuration was updated Cleared by writing any value to at least one of the PHY Port y Status registers.
4	LPE	Link partner error: 0:No error detected 1:Link partner error
3	RE	Read error: 0:No read error occurred 1:A read error has occurred Cleared by writing any value to at least one of the PHY Port y Status registers.
2	LSE	Link status error: 0:No error 1:Link error, link inhibited

Bits	Name	Description
1	LS	Link status (100 Mbit/s, Full Duplex, Auto negotiation): 0:No link 1:Link detected
0	PLS	Physical link status (PHY status register 1.2): 0:No physical link 1:Physical link detected

PHY_STAT Bits

57.4.58 FMMU[LOGIC_START_ADDR] (0x600 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FMMU[LOGIC_START_ADDR] [31:0]

Bits	Name	Description
31-0	ADDR	Logical start address within the EtherCAT Address Space.

FMMU[LOGIC_START_ADDR] Bits

57.4.59 FMMU[LENGTH] (0x604 + 0x10 * n)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET															
RO															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FMMU[LENGTH] [15:0]

Bits	Name	Description
15-0	OFFSET	Offset from the first logical FMMU byte to the last FMMU byte + 1 (e.g., if two bytes are used, then this parameter shall contain 2)

FMMU[LENGTH] Bits

57.4.60 FMMU[LOGIC_START_BIT] (0x606 + 0x10 * n)

7	6	5	4	3	2	1	0
RSVD					START		
N/A					RO		
x	x	x	x	x	0	0	0

FMMU[LOGIC_START_BIT] [7:0]

Bits	Name	Description
2-0	START	Logical starting bit that shall be mapped (bits are counted from least significant bit 0 to most significant bit 7)

FMMU[LOGIC_START_BIT] Bits

57.4.61 FMMU[LOGIC_STOP_BIT] (0x607 + 0x10 * n)

7	6	5	4	3	2	1	0
RSVD					STOP		
N/A					RO		
x	x	x	x	x	0	0	0

FMMU[LOGIC_STOP_BIT] [7:0]

Bits	Name	Description
2-0	STOP	Last logical bit that shall be mapped (bits are counted from least significant bit 0 to most significant bit 7)

FMMU[LOGIC_STOP_BIT] Bits

57.4.62 FMMU[PHYSICAL_START_ADDR] (0x608 + 0x10 * n)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR															
RO															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FMMU[PHYSICAL_START_ADDR] [15:0]

Bits	Name	Description
15-0	ADDR	Physical Start Address (mapped to logical Start address)

FMMU[PHYSICAL_START_ADDR] Bits

57.4.63 FMMU[PHYSICAL_START_BIT] (0x60A + 0x10 * n)

7	6	5	4	3	2	1	0
RSVD					START		
N/A					RO		
x	x	x	x	x	0	0	0

FMMU[PHYSICAL_START_BIT] [7:0]

Bits	Name	Description
2-0	START	Physical starting bit as target of logical start bit mapping (bits are counted from least significant bit 0 to most significant bit 7)

FMMU[PHYSICAL_START_BIT] Bits

57.4.64 FMMU[TYPE] (0x60B + 0x10 * n)

7	6	5	4	3	2	1	0
RSVD					MAP_WR		MAP_RD
N/A					RO		RO
x	x	x	x	x	x	0	0

FMMU[TYPE] [7:0]

Bits	Name	Description
1	MAP_WR	0:Ignore mapping for write accesses 1:Use mapping for write accesses
0	MAP_RD	0:Ignore mapping for read accesses 1:Use mapping for read accesses

FMMU[TYPE] Bits

57.4.65 FMMU[ACTIVATE] (0x60C + 0x10 * n)

7	6	5	4	3	2	1	0
RSVD							ACT
N/A							RO
x	x	x	x	x	x	x	0

FMMU[ACTIVATE] [7:0]

Bits	Name	Description
0	ACT	0:FMMU deactivated 1:FMMU activated. FMMU checks logically addressed blocks to be mapped according to configured mapping

FMMU[ACTIVATE] Bits

57.4.66 SYNCM[PHYSICAL_START_ADDR] (0x800 + 0x8 * n)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR															
RO															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SYNCM[PHYSICAL_START_ADDR] [15:0]

Bits	Name	Description
15-0	ADDR	First byte that will be handled by SyncManager

SYNCM[PHYSICAL_START_ADDR] Bits

57.4.67 SYNCM[LENGTH] (0x802 + 0x8 * n)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LEN															
RO															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SYNCM[LENGTH] [15:0]

Bits	Name	Description
15-0	LEN	Number of bytes assigned to SyncManager (shall be greater than 1, otherwise SyncManager is not activated. If set to 1, only Watchdog Trigger is generated if configured)

SYNCM[LENGTH] Bits

57.4.68 SYNCM[CONTROL] (0x804 + 0x8 * n)

7	6	5	4	3	2	1	0
RSVD	WDG_TRG_EN	INT_AL	INT_ECAT	DIR		OP_MODE	
N/A	RO	RO	RO	RO		RO	
x	0	0	0	0	0	0	0

SYNCM[CONTROL] [7:0]

Bits	Name	Description
6	WDG_TRG_EN	Watchdog Trigger Enable: 0:Disabled 1:Enabled
5	INT_AL	Interrupt in AL Event Request Register: 0:Disabled 1:Enabled
4	INT_ECAT	Interrupt in ECAT Event Request Register: 0:Disabled 1:Enabled
3-2	DIR	Direction: 00:Read:ECAT read access, PDI write access. 01:Write:ECAT write access, PDI read access. 10:Reserved 11:Reserved
1-0	OP_MODE	Operation Mode: 00:Buffered (3 buffer mode) 01:Reserved 10:Mailbox (Single buffer mode) 11:Reserved

SYNCM[CONTROL] Bits

57.4.69 SYNCM[STATUS] (0x805 + 0x8 * n)

7	6	5	4	3	2	1	0
WB_INUSE	RB_INUSE	BUF_MODE		MBX_MODE	RSVD	INT_RD	INT_WR
RO	RO	RO		RO	N/A	RO	RO
0	0	0	0	0	x	0	0

SYNCM[STATUS] [7:0]

Bits	Name	Description
7	WB_INUSE	Write buffer in use (opened)
6	RB_INUSE	Read buffer in use (opened)
5-4	BUF_MODE	Buffered mode:buffer status (last written buffer): 00:1 st buffer 01:2 nd buffer 10:3 rd buffer 11:(no buffer written) Mailbox mode:reserved
3	MBX_MODE	Mailbox mode:mailbox status: 0:Mailbox empty 1:Mailbox full Buffered mode:reserved
1	INT_RD	Interrupt Read: 1:Interrupt after buffer was completely and successfully read 0:Interrupt cleared after first byte of buffer was written NOTE:This interrupt is signalled to the writing side if enabled in the SM Control register
0	INT_WR	Interrupt Write: 1:Interrupt after buffer was completely and successfully written 0:Interrupt cleared after first byte of buffer was read NOTE:This interrupt is signalled to the reading side if enabled in the SM Control register

SYNCM[STATUS] Bits

57.4.70 SYNCM[ACTIVATE] (0x806 + 0x8 * n)

7	6	5	4	3	2	1	0
LATCH_PDI	LATCH_ECAT	RSVD				REPEAT	EN
RO	RO	N/A				RO	RW
0	0	x	x	x	x	0	0

SYNCM[ACTIVATE] [7:0]

Bits	Name	Description
7	LATCH_PDI	Latch Event PDI: 0:No 1:Generate Latch events when PDI issues a buffer exchange or when PDI accesses buffer start address
6	LATCH_ECAT	Latch Event ECAT: 0:No 1:Generate Latch event when EtherCAT master issues a buffer exchange
1	REPEAT	Repeat Request: A toggle of Repeat Request means that a mailbox retry is needed (primarily used in conjunction with ECAT Read Mailbox)
0	EN	SyncManager Enable/Disable: 0:Disable:Access to Memory without SyncManager control 1:Enable:SyncManager is active and controls Memory area set in configuration

SYNCM[ACTIVATE] Bits

57.4.71 SYNCM[PDI_CTRL] (0x807 + 0x8 * n)

7	6	5	4	3	2	1	0
RSVD						REPEAT_ACK	DEACT
N/A						RW	RW
x	x	x	x	x	x	0	0

7	6	5	4	3	2	1	0

SYNCM[PDI_CTRL] [7:0]

Bits	Name	Description
1	REPEAT_ACK	Repeat Ack: If this is set to the same value as that set by Repeat Request, the PDI acknowledges the execution of a previous set Repeat request.
0	DEACT	Deactivate SyncManager: Read: 0:Normal operation, SyncManager activated. 1:SyncManager deactivated and reset. SyncManager locks access to Memory area. Write: 0:Activate SyncManager 1:Request SyncManager deactivation NOTE:Writing 1 is delayed until the end of the frame, which is currently processed.

SYNCM[PDI_CTRL] Bits

57.4.72 RCV_TIME (0x900 + 0x4 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LT																								REQ							
RO																								RO							
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

RCV_TIME [31:0]

Bits	Name	Description
31-8	LT	Local time at the beginning of the last receive frame containing a write access to register 0x0900.

Bits	Name	Description
7-0	REQ	Write: A write access to register 0x0900 with BWR or FPWR latches the local time at the beginning of the receive frame (start first bit of preamble) at each port. Write (ESC20, ET1200 exception): A write access latches the local time at the beginning of the receive frame at port 0. It enables the time stamping at the other ports. Read: Local time at the beginning of the last receive frame containing a write access to this register. NOTE:FPWR requires an address match for accessing this register like any FPWR command. All write commands with address match will increment the working counter (e.g., APWR), but they will not trigger receive time latching.

RCV_TIME Bits

57.4.73 SYS_TIME (0x910)

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
ST																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SYS_TIME [63:32]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SYS_TIME [31:0]

Bits	Name	Description
63-0	ST	

Bits	Name	Description

SYS_TIME Bits

57.4.74 RCVT_ECAT_PU (0x918)

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
R																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RCVT_ECAT_PU [63:32]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RCVT_ECAT_PU [31:0]

Bits	Name	Description
63-0	LT	Local time at the beginning of a frame (start first bit of preamble) received at the ECAT Processing Unit containing a write access to register 0x0900 NOTE:E.g., if port 0 is open, this register reflects the Receive Time Port 0 as a 64 Bit value. Any valid EtherCAT write access to register 0x0900 triggers latching, not only BWR/FPWR commands as with register 0x0900.

RCVT_ECAT_PU Bits

57.4.75 SYS_TIME_OFFSET (0x920)

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
R																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SYS_TIME_OFFSET [63:32]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SYS_TIME_OFFSET [31:0]

Bits	Name	Description
63-0	OFFSET	Difference between local time and System Time. Offset is added to the local time.

SYS_TIME_OFFSET Bits

57.4.76 SYS_TIME_DELAY (0x928)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DLY																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SYS_TIME_DELAY [31:0]

Bits	Name	Description
31-0	DLY	Delay between Reference Clock and the ESC

SYS_TIME_DELAY Bits

57.4.77 SYS_TIME_DIFF (0x92C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIFF	NUM																														
RO	RO																														
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SYS_TIME_DIFF [31:0]

Bits	Name	Description
31	DIFF	0:Local copy of System Time less than received System Time 1:Local copy of System Time greater than or equal to received System Time
30-0	NUM	Mean difference between local copy of System Time and received System Time values Difference = Received System Time - local copy of System Time

SYS_TIME_DIFF Bits

57.4.78 SPD_CNT_START (0x930)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	BW														
N/A	RW														
x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPD_CNT_START [15:0]

Bits	Name	Description
14-0	BW	Bandwidth for adjustment of local copy of System Time (larger values → smaller bandwidth and smoother adjustment) A write access resets System Time Difference (0x092C:0x092F) and Speed Counter Diff (0x0932:0x0933). Valid values:0x0080 to 0x3FFF

SPD_CNT_START Bits

57.4.79 SPD_CNT_DIFF (0x932)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIFF															
RO															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPD_CNT_DIFF [15:0]

Bits	Name	Description
15-0	DIFF	Representation of the deviation between local clock period and Reference Clock's clock period (representation:two's complement) Range:±(Speed Counter Start -0x7F)

SPD_CNT_DIFF Bits

57.4.80 SYS_TIME_DIFF_FD (0x934)

7	6	5	4	3	2	1	0
RSVD				DEPTH			
N/A				RW			
x	x	x	x	0	0	0	0

SYS_TIME_DIFF_FD [7:0]

Bits	Name	Description
3-0	DEPTH	Filter depth for averaging the received System Time deviation IP Core since V2.2.0/V2.02a: A write access resets System Time Difference (0x092C:0x092F)

SYS_TIME_DIFF_FD Bits

57.4.81 SPD_CNT_FD (0x935)

7	6	5	4	3	2	1	0
RSVD				DEPTH			
N/A				RW			
x	x	x	x	0	0	0	0

SPD_CNT_FD [7:0]

Bits	Name	Description
3-0	DEPTH	Filter depth for averaging the clock period deviation IP Core since V2.2.0/V2.02a: A write access resets the internal speed counter filter

Bits	Name	Description

SPD_CNT_FD Bits

57.4.82 RCV_TIME_LM (0x936)

7	6	5	4	3	2	1	0
RSVD							LATCH_MODE
N/A							RO
x	x	x	x	x	x	x	0

RCV_TIME_LM [7:0]

Bits	Name	Description
0	LATCH_MODE	Receive Time Latch Mode: 0:Forwarding mode (used if frames are entering the ESC at port 0 first): Receive time stamps of ports 1-3 are enabled after the write access to 0x0900, so the following frame at ports 1-3 will be time stamped (this is typically the write frame to 0x0900 coming back from the network behind the ESC). 1:Reverse mode (used if frames are entering ESC at port 1-3 first): Receive time stamps of ports 1-3 are immediately taken over from the internal hidden time stamp registers, so the previous frame entering the ESC at ports 1-3 will be time stamped when the write frame to 0x0900 enters port 0 (the previous frame at ports 1-3 is typically the write frame to 0x0900 coming from the master, which will enable time stamp

RCV_TIME_LM Bits

57.4.83 CYC_UNIT_CTRL (0x980)

7	6	5	4	3	2	1	0
RSVD		LATCHI1	LATCHI0		RSVD		SYNCO
N/A		RO	RO		N/A		RO
x	x	0	0	x	x	x	0

CYC_UNIT_CTRL [7:0]

Bits	Name	Description
5	LATCHI1	Latch In unit 1: 0:ECAT-controlled 1:PDI-controlled NOTE:Latch interrupt is routed to ECAT/PDI depending on this setting
4	LATCHI0	Latch In unit 0: 0:ECAT-controlled 1:PDI-controlled NOTE:Latch interrupt is routed to ECAT/PDI depending on this setting. Always 1 (PDI-controlled) if System Time is PDIcontrolled.
0	SYNCO	Cyclic Unit and SYNCO out unit control: 0:ECAT-controlled 1:PDI-controlled

CYC_UNIT_CTRL Bits

57.4.84 SYNCO_ACT (0x981)

7	6	5	4	3	2	1	0
SSDP	NFC	STPC	EXT	AC	SYNC1_GEN	SYNC0_GEN	SOJA
RW	RW	RW	RW	RW	RW	RW	RW
0	0	0	0	0	0	0	0

SYNCO_ACT [7:0]

Bits	Name	Description
7	SSDP	SyncSignal debug pulse (Vasily bit): 0:Deactivated 1:Immediately generate one ping only on SYNC0-1 according to 0x0981[2:1 for debugging This bit is self-clearing, always read 0. All pulses are generated at the same time, the cycle time is ignored. The configured pulse length is used.
6	NFC	Near future configuration (approx.): 0:½ DC width future (231 ns or 263 ns) 1:~2.1 sec. future (231 ns)
5	STPC	Start Time plausibility check: 0:Disabled. SyncSignal generation if Start Time is reached. 1:Immediate SyncSignal generation if Start Time is outside near future (see 0x0981[6])
4	EXT	Extension of Start Time Cyclic Operation (0x0990:0x0993): 0:No extension 1:Extend 32 bit written Start Time to 64 bit
3	AC	Auto-activation by writing Start Time Cyclic Operation (0x0990:0x0997): 0:Disabled 1:Auto-activation enabled. 0x0981[0] is set automatically after Start Time is written.
2	SYNC1_GEN	SYNC1 generation: 0:Deactivated 1:SYNC1 pulse is generated
1	SYNC0_GEN	SYNC0 generation: 0:Deactivated 1:SYNC0 pulse is generated
0	SOUA	Sync Out Unit activation: 0:Deactivated 1:Activated

SYNCO_ACT Bits

57.4.85 PULSE_LEN (0x982)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LEN															
RO															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PULSE_LEN [15:0]

Bits	Name	Description
15-0	LEN	Pulse length of SyncSignals (in Units of 10ns) 0:Acknowledge mode:SyncSignal will be cleared by reading SYNC[1:0] Status register

PULSE_LEN Bits

57.4.86 ACT_STAT (0x984)

7	6	5	4	3	2	1	0
RSVD					CHK_RSLT	SYNC1	SYNC0
N/A					RO	RO	RO
x	x	x	x	x	0	0	0

ACT_STAT [7:0]

Bits	Name	Description
2	CHK_RSLT	Start Time Cyclic Operation (0x0990:0x0997) plausibility check result when Sync Out Unit was activated: 0:Start Time was within near future 1:Start Time was out of near future (0x0981[6])
1	SYNC1	SYNC1 activation state: 0:First SYNC1 pulse is not pending 1:First SYNC1 pulse is pending
0	SYNC0	SYNC0 activation state: 0:First SYNC0 pulse is not pending 1:First SYNC0 pulse is pending

ACT_STAT Bits

57.4.87 SYNC0_STAT (0x98E)

7	6	5	4	3	2	1	0
RSVD							ACK
N/A							RW
x	x	x	x	x	x	x	0

SYNC0_STAT [7:0]

Bits	Name	Description
0	ACK	SYNC0 state for Acknowledge mode. SYNC0 in Acknowledge mode is cleared by reading this register from PDI, use only in Acknowledge mode

SYNC0_STAT Bits

57.4.88 SYNC1_STAT (0x98F)

7	6	5	4	3	2	1	0
RSVD							ACK
N/A							RW
x	x	x	x	x	x	x	0

SYNC1_STAT [7:0]

Bits	Name	Description
0	ACK	SYNC1 state for Acknowledge mode. SYNC1 in Acknowledge mode is cleared by reading this register from PDI, use only in Acknowledge mode

SYNC1_STAT Bits

57.4.89 START_TIME_CO (0x990)

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

START_TIME_CO [63:32]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

START_TIME_CO [31:0]

Bits	Name	Description
63-0	ST	Write:Start time (System time) of cyclic operation in ns Read:System time of next SYNC0 pulse in ns

START_TIME_CO Bits

57.4.90 NXT_SYNC1_PULSE (0x998)

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
TIME																															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

NXT_SYNC1_PULSE [63:32]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIME																															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

NXT_SYNC1_PULSE [31:0]

Bits	Name	Description
63-0	TIME	System time of next SYNC1 pulse in ns

NXT_SYNC1_PULSE Bits

57.4.91 SYNC0_CYC_TIME (0x9A0)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CYC																															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SYNC0_CYC_TIME [31:0]

Bits	Name	Description
31-0	CYC	Time between two consecutive SYNC0 pulses in ns. 0:Single shot mode, generate only one SYNC0 pulse.

SYNC0_CYC_TIME Bits

57.4.92 SYNC1_CYC_TIME (0x9A4)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CYC																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SYNC1_CYC_TIME [31:0]

Bits	Name	Description
31-0	CYC	Time between SYNC0 pulse and SYNC1 pulse in ns

SYNC1_CYC_TIME Bits

57.4.93 LATCH0_CTRL (0x9A8)

7	6	5	4	3	2	1	0
RSVD						NEG_EDGE	POS_EDGE
N/A						RW	RW
x	x	x	x	x	x	0	0

LATCH0_CTRL [7:0]

Bits	Name	Description
1	NEG_EDGE	Latch0 negative edge: 0:Continuous Latch active 1:Single event (only first event active)
0	POS_EDGE	Latch0 positive edge: 0:Continuous Latch active 1:Single event (only first event active)

LATCH0_CTRL Bits

57.4.94 LATCH1_CTRL (0x9A9)

7	6	5	4	3	2	1	0
RSVD						NEG_EDGE	POS_EDGE
N/A						RW	RW
x	x	x	x	x	x	0	0

LATCH1_CTRL [7:0]

Bits	Name	Description
1	NEG_EDGE	Latch1 negative edge: 0:Continuous Latch active 1:Single event (only first event active)
0	POS_EDGE	Latch1 positive edge: 0:Continuous Latch active 1:Single event (only first event active)

LATCH1_CTRL Bits

57.4.95 LATCH0_STAT (0x9AE)

7	6	5	4	3	2	1	0
RSVD					PIN_STAT	NEG_EDGE	POS_EDGE
N/A					RO	RO	RO
x	x	x	x	x	0	0	0

LATCH0_STAT [7:0]

Bits	Name	Description
2	PIN_STAT	Latch0 pin state

Bits	Name	Description
1	NEG_EDGE	Event Latch0 negative edge. 0:Negative edge not detected or continuous mode 1:Negative edge detected in single event mode only. Flag cleared by reading out Latch0 Time Negative Edge.
0	POS_EDGE	Event Latch0 positive edge. 0:Positive edge not detected or continuous mode 1:Positive edge detected in single event mode only. Flag cleared by reading out Latch0 Time Positive Edge.

LATCH0_STAT Bits

57.4.96 LATCH1_STAT (0x9AF)

7	6	5	4	3	2	1	0
RSVD					PIN_STAT	NEG_EDGE	POS_EDGE
N/A					RO	RO	RO
x	x	x	x	x	0	0	0

LATCH1_STAT [7:0]

Bits	Name	Description
2	PIN_STAT	Latch1 pin state
1	NEG_EDGE	Event Latch1 negative edge. 0:Negative edge not detected or continuous mode 1:Negative edge detected in single event mode only. Flag cleared by reading out Latch1 Time Negative Edge.

Bits	Name	Description
0	POS_EDGE	Event Latch1 positive edge. 0:Positive edge not detected or continuous mode 1:Positive edge detected in single event mode only. Flag cleared by reading out Latch1 Time Positive Edge.

LATCH1_STAT Bits

57.4.97 LATCH0_TIME_PE (0x9B0)

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
TIME																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LATCH0_TIME_PE [63:32]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIME																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LATCH0_TIME_PE [31:0]

Bits	Name	Description
63-0	TIME	System time at the positive edge of the Latch0 signal.

LATCH0_TIME_PE Bits

57.4.98 LATCH0_TIME_NE (0x9B8)

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
TIME																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LATCH0_TIME_NE [63:32]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIME																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LATCH0_TIME_NE [31:0]

Bits	Name	Description
63-0	TIME	System time at the negative edge of the Latch0 signal.

LATCH0_TIME_NE Bits

57.4.99 LATCH1_TIME_PE (0x9C0)

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
TIME																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LATCH1_TIME_PE [63:32]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIME																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LATCH1_TIME_PE [31:0]

Bits	Name	Description
63-0	TIME	System time at the positive edge of the Latch1 signal.

LATCH1_TIME_PE Bits

57.4.100 LATCH1_TIME_NE (0x9C8)

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
TIME																															
RW																															

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LATCH1_TIME_NE [63:32]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIME																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LATCH1_TIME_NE [31:0]

Bits	Name	Description
63-0	TIME	System time at the negative edge of the Latch1 signal.

LATCH1_TIME_NE Bits

57.4.101 ECAT_BUF_CET (0x9F0)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIME																															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ECAT_BUF_CET [31:0]

Bits	Name	Description
31-0	TIME	Local time at the beginning of the frame which causes at least one SyncManager to assert an ECAT event

ECAT_BUF_CET Bits

57.4.102 PDI_BUF_SET (0x9F8)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIME																															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PDI_BUF_SET [31:0]

Bits	Name	Description
31-0	TIME	Local time when at least one SyncManager asserts a PDI buffer start event

PDI_BUF_SET Bits

57.4.103 PDI_BUF_CET (0x9FC)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIME																															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PDI_BUF_CET [31:0]

Bits	Name	Description
31-0	TIME	Local time when at least one SyncManager asserts a PDI buffer change event

PDI_BUF_CET Bits

57.4.104 PID (0xE00)

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
PID																															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PID [63:32]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PID																															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PID [31:0]

Bits	Name	Description
63-0	PID	Product ID

PID Bits

57.4.105 VID (0xE08)

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
VID																															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VID [63:32]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VID																															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VID [31:0]

Bits	Name	Description
63-0	VID	Vendor ID: 23-0: Company 31-24: Department NOTE: Test Vendor IDs have [31:28]=0xE

VID Bits

57.4.106 DIO_OUT_DATA (0xF00)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OD																															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DIO_OUT_DATA [31:0]

Bits	Name	Description
31-0	OD	Output Data

DIO_OUT_DATA Bits

57.4.107 GPO (0xF10)

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
GPOD																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPO [63:32]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPOD																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPO [31:0]

Bits	Name	Description
63-0	GPOD	General Purpose Output Data

GPO Bits

57.4.108 GPI (0xF18)

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
GPID																															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPI [63:32]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPID																															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPI [31:0]

Bits	Name	Description
63-0	GPID	General Purpose Input Data

GPI Bits

57.4.109 USER_RAM_BYTE0 (0xF80)

7	6	5	4	3	2	1	0
EXTF							
RW							
0	0	0	0	0	0	0	0

USER_RAM_BYTE0 [7:0]

Bits	Name	Description
7-0	EXTF	Number of extended feature bits

USER_RAM_BYTE0 Bits

57.4.110 USER_RAM_BYTE1 (0xF81)

7	6	5	4	3	2	1	0
PRWO	AEMW	GPO	GPI	CSA	EIM	ALSCR	EDLCR
RW	RW	RW	RW	RW	RW	RW	RW
0	0	0	0	0	0	0	0

USER_RAM_BYTE1 [7:0]

Bits	Name	Description
7	PRWO	Physical Read/Write Offset (0x0108:0x0109)
6	AEMW	AL Event Mask writable (0x0204:0x0207)
5	GPO	General Purpose Outputs (0x0F10:0x0F17)
4	GPI	General Purpose Inputs (0x0F18:0x0F1F)
3	CSA	Configured Station Alias (0x0012:0x0013)
2	EIM	ECAT Interrupt Mask (0x0200:0x0201)
1	ALSCR	AL Status Code Register (0x0134:0x0135)
0	EDLCR	Extended DL Control Register (0x0102:0x0103)

USER_RAM_BYTE1 Bits

57.4.111 USER_RAM_BYTE2 (0xF82)

7	6	5	4	3	2	1	0
ESCFG	EPUPEC	DCSMET	RSVD	RESET	WP	WDGCNT	WDW
RW	RW	RW	N/A	RW	RW	RW	RW
0	0	0	x	0	0	0	0

USER_RAM_BYTE2 [7:0]

Bits	Name	Description
7	ESCFG	EEPROM Size configurable (0x0502[7]): 0:EEPROM Size fixed to sizes up to 16 Kbit 1:EEPROM Size configurable
6	EPUPEC	ECAT Processing Unit/PDI Error Counter (0x030C:0x030D)
5	DCSMET	DC SyncManager Event Times (0x09F0:0x09FF)
3	RESET	Reset (0x0040:0x0041)
2	WP	Write Protection (0x0020:0x0031)
1	WDGCNT	Watchdog counters (0x0442:0x0443)
0	WDW	Watchdog divider writable (0x0400:0x0401) and Watchdog PDI (0x0410:0x0411)

USER_RAM_BYTE2 Bits

57.4.112 USER_RAM_BYTE3 (0xF83)

7	6	5	4	3	2	1	0
RLED	ELDE	ELDM	MMI	LLC	RSVD		
RW	RW	RW	RW	RW	N/A		
0	0	0	0	0	x	x	x

USER_RAM_BYTE3 [7:0]

Bits	Name	Description
7	RLED	Run LED (DEV_STATE LED)
6	ELDE	Enhanced Link Detection EBUS
5	ELDM	Enhanced Link Detection MII
4	MMI	MII Management Interface (0x0510:0x0515)
3	LLC	Lost Link Counter (0x0310:0x0313)

USER_RAM_BYTE3 Bits

57.4.113 USER_RAM_BYTE4 (0xF84)

7	6	5	4	3	2	1	0
LDCM	DTLC	DSOU	RSVD	DLIU	RSVD		LALED
RW	RW	RW	N/A	RW	N/A		RW
0	0	0	x	0	x	x	0

USER_RAM_BYTE4 [7:0]

Bits	Name	Description
7	LDCM	Link detection and configuration by MI
6	DTLC	DC Time loop control assigned to PDI
5	DSOU	DC Sync Out Unit
3	DLIU	DC Latch In Unit
0	LALED	Link/Activity LED

USER_RAM_BYTE4 Bits

57.4.114 USER_RAM_BYTE5 (0xF85)

7	6	5	4	3	2	1	0
RSVD		DDIOR		RSVD	EEU	ATS	MCP
N/A		RW		N/A	RW	RW	RW
x	x	0	x	x	0	0	0

USER_RAM_BYTE5 [7:0]

Bits	Name	Description
5	DDIOR	Disable Digital I/O register (0x0F00:0x0F03)
2	EEU	EEPROM emulation by μ Controller
1	ATS	Automatic TX shift
0	MCP	MI control by PDI possible

USER_RAM_BYTE5 Bits

57.4.115 USER_RAM_BYTE6 (0xF86)

7	6	5	4	3	2	1	0
RSVD					RELEDOR		RSVD
N/A					RW		N/A
x	x	x	x	x	0	x	x

USER_RAM_BYTE6 [7:0]

Bits	Name	Description
2	RELEDOR	RUN/ERR LED Override (0x0138:0x0139)

USER_RAM_BYTE6 Bits

57.4.116 USER_RAM_BYTE7 (0xF87)

7	6	5	4	3	2	1	0
DCST	DCRT	RSVD		DCS1D	RSVD		
RW	RW	N/A		RW	N/A		
0	0	x	x	0	x	x	x

USER_RAM_BYTE7 [7:0]

Bits	Name	Description
7	DCST	DC System Time (0x0910:0x0936)
6	DCRT	DC Receive Times (0x0900:0x090F)
3	DCS1D	DC Sync1 disable

USER_RAM_BYTE7 Bits

57.4.117 USER_RAM_BYTE8 (0xF88)

7	6	5	4	3	2	1	0
RSVD		PPDI	OPDI	APDI	PDICEC	RSVD	DC64
N/A		RW	RW	RW	RW	N/A	RW
x	x	0	0	0	0	x	0

USER_RAM_BYTE8 [7:0]

Bits	Name	Description
5	PPDI	PLB PDI
4	OPDI	OPB PDI
3	APDI	Avalon PDI
2	PDICEC	PDI clears error counter
0	DC64	DC 64 bit

USER_RAM_BYTE8 Bits

57.4.118 USER_RAM_BYTE9 (0xF89)

7	6	5	4	3	2	1	0
DR	RSVD						
RW	N/A						
0	x	x	x	x	x	x	x

USER_RAM_BYTE9 [7:0]

Bits	Name	Description
7	DR	Direct RESET

USER_RAM_BYTE9 Bits

57.4.119 USER_RAM_BYTE10 (0xF8A)

7	6	5	4	3	2	1	0
PDIIR	PDIFA	RSVD		APDI	DCL1D	RSVD	
RW	RW	N/A		RW	RW	N/A	
0	0	x	x	0	0	x	x

USER_RAM_BYTE10 [7:0]

Bits	Name	Description
7	PDIIR	PDI Information register (0x014E:0x014F)
6	PDIFA	PDI function acknowledge by PDI write
3	APDI	AXI PDI
2	DCL1D	DC Latch1 disable

USER_RAM_BYTE10 Bits

57.4.120 USER_RAM_BYTE11 (0xF8B)

7	6	5	4	3	2	1	0
RSVD				LEDTST	RSVD		
N/A				RW	N/A		
x	x	x	x	0	x	x	x

USER_RAM_BYTE11 [7:0]

Bits	Name	Description
3	LEDTST	LED test

USER_RAM_BYTE11 Bits

57.4.121 USER_RAM_BYTE14 (0xF8E)

7	6	5	4	3	2	1	0
DIOBS	RSVD						
RW	N/A						

7	6	5	4	3	2	1	0
0	0	x	x	x	x	x	x

USER_RAM_BYTE14 [7:0]

Bits	Name	Description
7-6	DIOBS	Digital I/O PDI byte size

USER_RAM_BYTE14 Bits

57.4.122 USER_RAM_BYTE15 (0xF8F)

7	6	5	4	3	2	1	0
RSVD			AUCPDI	SSPDI	DIOPDI	RSVD	
N/A			RW	RW	RW	N/A	
x	x	x	0	0	0	x	x

USER_RAM_BYTE15 [7:0]

Bits	Name	Description
4	AUCPDI	Asynchronous μ C PDI
3	SSPDI	SPI Slave PDI
2	DIOPDI	Digital I/O PDI

USER_RAM_BYTE15 Bits

57.4.123 USER_RAM_BYTE19 (0xF93)

7	6	5	4	3	2	1	0
RSVD	SCP	RMII	URGP	RSVD	CIA	IPARO	RGMI
N/A	RW	RW	RW	N/A	RW	RW	RW
x	0	0	0	x	0	0	0

USER_RAM_BYTE19 [7:0]

Bits	Name	Description
6	SCP	Security CPLD protection
5	RMII	RMII
4	URGP	Use RGMII GTX_CLK phase shifted clock input
2	CIA	CLK_PDI_EXT is asynchronous
1	IPARO	Individual PHY address read out (0x0510[7:3])
0	RGMI	RGMI

USER_RAM_BYTE19 Bits

57.4.124 PDRAM (0x1000)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PDRAM [31:0]

Bits	Name	Description
31-0	DATA	Input Data

PDRAM Bits

57.4.125 PDRAM_ALS (0x10000)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PDRAM_ALS [31:0]

Bits	Name	Description
31-0	DATA	

PDRAM_ALS Bits

57.4.126 GPR_CFG0 (0x1F000)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																							
RSVD													PORT2_DIS	PORT1_DIS	PORT0_DIS	RSVD		CLK100_EN	EEPROM_EMU	RSVD							SYNC1_PDI_IRGEN	SYNC0_PDI_IRGEN	I2C_SCLK_EN	RSVD		PROM_SIZE																						
													N/A														RW	RW	RW			N/A	RW	RW	N/A							RW	RW	RW	N/A	RW								
																											x	x	x			x	x	x								x	x	x	x	x	x	0	0	0	x	x	0	1

GPR_CFG0 [31:0]

Bits	Name	Description
18	PORT2_DIS	set to disable related port(read data 00 in core reigster Port Descriptor (0x0007)
17	PORT1_DIS	set to disable related port(read data 00 in core reigster Port Descriptor (0x0007)
16	PORT0_DIS	set to disable related port(read data 00 in core reigster Port Descriptor (0x0007)
13	CLK100_EN	
12	EEPROM_EMU	1 is EEPROM emulation mode (default)
5	SYNC1_PDI_IRQE N	set to enable SYNC1 generate PDI interrupt, also set PDI_SL_CFG.SYNC1_MAER(0x151.bit7)
4	SYNC0_PDI_IRQE N	set to enable SYNC0 generate PDI interrupt, also set PDI_SL_CFG.SYNC1_MAER(0x151.bit3)
3	I2C_SCLK_EN	
0	PROM_SIZE	Sets EEPROM size: 0:up to 16 kbit EEPROM 1:32 kbit-4Mbit EEPROM

GPR_CFG0 Bits

57.4.127 GPR_CFG1 (0x1F004)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
SYNC1_IRQ_EN	SYNC0_IRQ_EN	RSTO_IRQ_EN	RSVD																SYNC1_DMA_EN	SYNC0_DMA_EN	RSVD		LATCH1_FROM_IO	LATCH0_FROM_IO	RSTO_OVRD	RSTO_OVRD_ENJ	RSVD							
RW	RW	RW	N/A																RW	RW	N/A		RW	RW	RW	RW	N/A							
0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	x	x	0	0	0	0	x	x	x	x	x	x			

GPR_CFG1 [31:0]

Bits	Name	Description
31	SYNC1_IRQ_EN	
30	SYNC0_IRQ_EN	
29	RSTO_IRQ_EN	
13	SYNC1_DMA_EN	
12	SYNC0_DMA_EN	
9	LATCH1_FROM_I O	0:from NTM
8	LATCH0_FROM_I O	0:from TRIGGER_MUX

Bits	Name	Description
7	RSTO_OVRD	
6	RSTO_OVRD_EN J	

GPR_CFG1 Bits

57.4.128 GPR_CFG2 (0x1F008)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD		NMII_LINK2_FROM_IO	NMII_LINK2_GPR	RSVD		NMII_LINK1_FROM_IO	NMII_LINK1_GPR	RSVD		NMII_LINK0_FROM_IO	NMII_LINK0_GPR	RSVD																			
		N/A				RW	RW			N/A																		RW	RW		
						N/A																						RW	RW		
																												N/A		RW	RW
N/A		RW	RW	N/A		RW	RW	N/A																							
		N/A				RW	RW																	N/A		RW	RW	N/A			
N/A				RW	RW	N/A		RW	RW	N/A																					
		N/A		RW	RW			N/A																		RW	RW	N/A			
N/A				RW	RW	N/A				RW	RW	N/A																			
		N/A		RW	RW			N/A		RW	RW																	N/A			
N/A				RW	RW	N/A				RW	RW	N/A																			
		N/A		RW	RW			N/A		RW	RW																	N/A			
N/A				RW	RW	N/A				RW	RW	N/A																			
		N/A		RW	RW			N/A		RW	RW																	N/A			
N/A				RW	RW	N/A				RW	RW	N/A																			
		N/A		RW	RW			N/A		RW	RW																	N/A			
N/A				RW	RW	N/A				RW	RW	N/A																			
		N/A		RW	RW			N/A		RW	RW																	N/A			
N/A				RW	RW	N/A				RW	RW	N/A																			
		N/A		RW	RW			N/A		RW	RW																	N/A			
N/A				RW	RW	N/A				RW	RW	N/A																			
		N/A		RW	RW			N/A		RW	RW																	N/A			
N/A				RW	RW	N/A				RW	RW	N/A																			
		N/A		RW	RW			N/A		RW	RW																	N/A			
N/A				RW	RW	N/A				RW	RW	N/A																			
		N/A		RW	RW			N/A		RW	RW																	N/A			
N/A				RW	RW	N/A				RW	RW	N/A																			
		N/A		RW	RW			N/A		RW	RW																	N/A			
N/A				RW	RW	N/A				RW	RW	N/A																			
		N/A		RW	RW			N/A		RW	RW																	N/A			
N/A				RW	RW	N/A				RW	RW	N/A																			
		N/A		RW	RW			N/A		RW	RW																	N/A			
N/A				RW	RW	N/A				RW	RW	N/A																			
		N/A		RW	RW			N/A		RW	RW																	N/A			
N/A				RW	RW	N/A				RW	RW	N/A																			
		N/A		RW	RW			N/A		RW	RW																	N/A			
N/A				RW	RW	N/A				RW	RW	N/A																			
		N/A		RW	RW			N/A		RW	RW																	N/A			
N/A				RW	RW	N/A				RW	RW	N/A																			
		N/A		RW	RW			N/A		RW	RW																	N/A			
N/A				RW	RW	N/A				RW	RW	N/A																			
		N/A		RW	RW			N/A		RW	RW																	N/A			
N/A				RW	RW	N/A				RW	RW	N/A																			
		N/A		RW	RW			N/A		RW	RW																	N/A			
N/A				RW	RW	N/A				RW	RW	N/A																			
		N/A		RW	RW			N/A		RW	RW																	N/A			
N/A				RW	RW	N/A				RW	RW	N/A																			
		N/A		RW	RW			N/A		RW	RW																	N/A			
N/A				RW	RW	N/A				RW	RW	N/A																			
		N/A		RW	RW			N/A		RW	RW																	N/A			
N/A				RW	RW	N/A				RW	RW	N/A																			
		N/A		RW	RW			N/A		RW	RW																	N/A			
N/A				RW	RW	N/A				RW	RW	N/A																			
		N/A		RW	RW			N/A		RW	RW																	N/A			
N/A				RW	RW	N/A				RW	RW	N/A																			
		N/A		RW	RW			N/A		RW	RW																	N/A			
N/A				RW	RW	N/A				RW	RW	N/A																			
		N/A		RW	RW			N/A		RW	RW																	N/A			
N/A				RW	RW	N/A				RW	RW	N/A																			
		N/A		RW	RW			N/A		RW	RW																	N/A			
N/A				RW	RW	N/A				RW	RW	N/A																			
		N/A		RW	RW			N/A		RW	RW																	N/A			
N/A				RW	RW	N/A				RW	RW	N/A																			
		N/A		RW	RW			N/A		RW	RW																	N/A			
N/A				RW	RW	N/A				RW	RW	N/A																			
		N/A		RW	RW			N/A		RW	RW																	N/A			
N/A				RW	RW	N/A				RW	RW	N/A																			
		N/A		RW	RW			N/A		RW	RW																	N/A			
N/A				RW	RW	N/A				RW	RW	N/A																			
		N/A		RW	RW			N/A		RW	RW																	N/A			
N/A				RW	RW	N/A				RW	RW	N/A																			
		N/A		RW	RW			N/A		RW	RW																	N/A			
N/A				RW	RW	N/A				RW	RW	N/A																			
		N/A		RW	RW			N/A		RW	RW																	N/A			
N/A				RW	RW	N/A				RW	RW	N/A																			
		N/A																													

GPR_CFG2 [31:0]

Bits	Name	Description
29	NMII_LINK2_FRO M_IO	
28	NMII_LINK2_GPR	
25	NMII_LINK1_FRO M_IO	
24	NMII_LINK1_GPR	
21	NMII_LINK0_FRO M_IO	
20	NMII_LINK0_GPR	

GPR_CFG2 Bits

57.4.129 PHY_CFG0 (0x1F010)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	MAC_SPEED	RSVD						PHY_OFFSET_VAL								PORT2_RMII_EN															
N/A	RW	N/A						RW								N/A															
x	0	x	0	0	0	0	0	0	x	x	x	x	x	x	x	0	x	x	x	x	x	x	x	0	x	x	x	x	x	x	x

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

PHY_CFG0 [31:0]

Bits	Name	Description
30	MAC_SPEED	1:100M
28-24	PHY_OFFSET_VA L	
23	PORT2_RMII_EN	
15	PORT1_RMII_EN	
7	PORT0_RMII_EN	

PHY_CFG0 Bits

57.4.130 PHY_CFG1 (0x1F014)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					RMII_REFCLK_SEL		REFCK_25M_INV		RMII_P2_TXCK_REFCLK_OE	RMII_P1_TXCK_REFCLK_OE	RMII_P0_TXCK_REFCLK_OE	REFCK_25M_OE	RMII_P2_RXCK_REFCLK_OE	RMII_P1_RXCK_REFCLK_OE	RMII_P0_RXCK_REFCLK_OE
N/A																					RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0

PHY_CFG1 [31:0]

Bits	Name	Description
10-8	RMII_REFCLK_SE L	0:use RXCK as 50M refclk. 1:use TXCK as 50M refclk
7	REFCK_25M_INV	
6	RMII_P2_TXCK_R EFCLK_OE	
5	RMII_P1_TXCK_R EFCLK_OE	
4	RMII_P0_TXCK_R EFCLK_OE	
3	REFCK_25M_OE	
2	RMII_P2_RXCK_R EFCLK_OE	
1	RMII_P1_RXCK_R EFCLK_OE	

Bits	Name	Description
0	RMII_P0_RXCK_R EFCLK_OE	

PHY_CFG1 Bits

57.4.131 GPIO_CTRL (0x1F020)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SW_LATCH_GPI	SW_LATCH_GPO	RSVD																GPI_OVERRIDE_EN	GPI_TRIG_EN	GPI_TRIG_SEL			RSVD			GPO_TRIG_EN	GPO_TRIG_SEL				
WO	WO	N/A																RW	RW	RW			N/A			RW	RW				
0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	x	x	x	0	0	0	0	0

GPIO_CTRL [31:0]

Bits	Name	Description
31	SW_LATCH_GPI	if gpi_trig_sel is set to 4'b1001, setting this bit will latch GPI to gpi_reg0/1
30	SW_LATCH_GPO	if gpo_trig_sel is set to 4'b1001, setting this bit will latch GPO to gpo_reg0/1
13	GPI_OVERRIDE_EN	set this bit will use GPI from the software register gpi_override0/1 clr to use GPI from pad directly
12	GPI_TRIG_EN	use gpi_trig_sel can select the trigger event to latch GPI signal(from reg or pad) set to use triggered signal; clr to use signals directly(from reg or pad) assign pdi_gpi = gpi_trig_en ? gpi_reg : (gpi_override_en ? gpi_override : pad_di_ecat_gpi);
11-8	GPI_TRIG_SEL	select the trigger signal to latch GPI. 0000: SOF; 0001: EOF; 0010: pos of SYNC0; 0011: pos of SYNC1; 0100: pos of LATCH0; 0101: pos of LATCH1; 0110: neg of LATCH0; 0111: neg of LATCH1 1000: wdog trigger; 1001: sw set gpio_ctrl[31]; 1110: use gpi_trg_cfg for separate byte control others no trigger

Bits	Name	Description
4	GPO_TRIG_EN	use gpo_trig_sel can select the trigger event to latch GPO signal(from core) set to use triggered signal; clr to use GPO signals directly(from reg or pad)
3-0	GPO_TRIG_SEL	select the trigger signal to latch GPO. 0000: SOF; 0001: EOF; 0010: pos of SYNC0; 0011: pos of SYNC1; 0100: pos of LATCH0; 0101: pos of LATCH1; 0110: neg of LATCH0; 0111: neg of LATCH1 1000: wdog trigger; 1001: sw set gpio_ctrl[30]; 1110: use gpo_trg_cfg for separate byte control others no trigger

GPIO_CTRL Bits

57.4.132 GPI_TRIG_CFG (0x1F024)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPI_TRIG_CFG																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPI_TRIG_CFG [31:0]

Bits	Name	Description
31-0	GPI_TRIG_CFG	separage GPI trigger control for latch GPI, 0000~1001 are same as gpi_trig_sel, just 4bit for one byte. Bit3:0 to control GPI[7:0], bit7:4 to control GPI[15:8]...bit31:28 to control GPI[63:56]

GPI_TRIG_CFG Bits

57.4.133 GPO_TRIG_CFG (0x1F028)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPO_TRIG_CFG																															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPO_TRIG_CFG [31:0]

Bits	Name	Description
31-0	GPO_TRIG_CFG	separage GPO trigger control for latch GPO, 0000~1001 are same as gpo_trig_sel, just 4bit for one byte. Bit3:0 to control GPO[7:0], bit7:4 to control GPO[15:8]...bit31:28 to control GPO[63:56]

GPO_TRIG_CFG Bits

57.4.134 GPI_OVERRIDE0 (0x1F030)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPR_OVERRIDE_LOW																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPI_OVERRIDE0 [31:0]

Bits	Name	Description
31-0	GPR_OVERRIDE_LOW	

GPI_OVERRIDE0 Bits

57.4.135 GPI_OVERRIDE1 (0x1F034)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPR_OVERRIDE_HIGH																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

GPI_OVERRIDE1 [31:0]

Bits	Name	Description
31-0	GPR_OVERRIDE_HIGH	

GPI_OVERRIDE1 Bits

57.4.136 GPO_REG0 (0x1F038)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPO_REG0 [31:0]

Bits	Name	Description
31-0	VALUE	

GPO_REG0 Bits

57.4.137 GPO_REG1 (0x1F03C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPO_REG1 [31:0]

Bits	Name	Description
31-0	VALUE	

GPO_REG1 Bits

57.4.138 GPI_REG0 (0x1F040)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPI_REG0 [31:0]

Bits	Name	Description
31-0	VALUE	

GPI_REG0 Bits

57.4.139 GPI_REG1 (0x1F044)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPI_REG1 [31:0]

Bits	Name	Description
31-0	VALUE	

GPI_REG1 Bits

57.4.140 GPR_STATUS (0x1F060)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NLINK2_PADSEL				NLINK1_PADSEL				NLINK0_PADSEL				PDI_SOF	PDI_EOF	PDI_WD_TRIGGER	PDI_WD_STATE	RSVD						SYNC_OUT1	SYNC_OUT0	RSVD	LED_STATE_RUN	LED_ERR	LED_RUN	DEV_STATE	LINK_ACT		
RO				RO				RO				RO	RO	RO	RO	N/A						RO	RO	N/A	RO	RO	RO	RO	RO		
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x	x	x	x	x	0	0	x	0	0	0	0	0	0	0

GPR_STATUS [31:0]

Bits	Name	Description
31-28	NLINK2_PADSEL	
27-24	NLINK1_PADSEL	
23-20	NLINK0_PADSEL	
19	PDI_SOF	
18	PDI_EOF	

Bits	Name	Description
17	PDI_WD_TRIGGE R	
16	PDI_WD_STATE	
9	SYNC_OUT1	
8	SYNC_OUT0	
6	LED_STATE_RUN	
5	LED_ERR	
4	LED_RUN	
3	DEV_STATE	
2-0	LINK_ACT	

GPR_STATUS Bits

57.4.141 IO_CFG (0x1F080 + 0x4 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																									INVERT	FUNC_ALT					
N/A																									RW	RW					
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	

IO_CFG [31:0]

Bits	Name	Description
4	INVERT	1:invert the IO
3-0	FUNC_ALT	IO usage: 0:NMII_LINK0 1:NMII_LINK1 2:NMII_LINK2 3:LINK_ACT0 4:LINK_ACT1 5:LINK_ACT2 6:LED_RUN 7:LED_ERR 8:RESET_OUT

IO_CFG Bits

58 Ethernet Physical Layer ENET_PHY

58.1 Overview

Some models of this product have built-in Ethernet PHY, and its features include:

- has 2 built-in Ethernet PHYs: PHY0 and PHY1
- 100BASE-TX full duplex
- supports connection between MII interface and main chip
- supports Auto-Negotiation
- supports Auto-MDIX

58.2 Pinout

58.2.1 External Pin

has a built-in Ethernet PHY that multiplexes the VIO_B01 power pin. Users need to connect VIO_B01 to the 3.3V power supply. The 1.2V power supply to the digital part of the built-in Ethernet PHY of

is realized by the PHY's internal LDO, which requires external connection of decoupling capacitors of 10uF and 0.1uF through two pins: VDD_PHY0CAP and VDD_PHY1CAP. In addition to the above power pins, other functional pins of

will occupy the GPIO of the chip, as shown in [Table 228](#).

pin	Built-in PHY name	Built-in PHY chip IO function
PA27	ENET PHY0	bias resistor RBIAS, 2.49K ohm resistor should be connected
PA28	ENET PHY0	100M Network Connection Indicator (LED1)/PHY Address Bit 1
PA29	ENET PHY0	10M Network Connection Indicator (LED1)/PHY Address Bit 0
PA30	ENET PHY0	MDIO
PA31	ENET PHY0	MDC
PA20	ENET PHY0	Data Receiving Negative RD_N
PA21	ENET PHY0	Data Receive Positive RD_P
PA22	ENET PHY0	Data Send Negative TD_N
PA23	ENET PHY0	Data Receive Positive TD_P
PA24	ENET PHY1	Bias resistor RBIAS, 2.49K ohm resistor should be connected
PA25	ENET PHY1	100M Network Connection Indicator (LED1)/PHY Address Bit 1
PA26	ENET PHY1	10M Network Connection Indicator (LED0)/PHY Address Bit 0

pin	Built-in PHY name	Built-in PHY chip IO function
PA30	ENET PHY1	MDIO
PA31	ENET PHY1	MDC
PA16	ENET PHY1	Data Receiving Negative RD_N
PA17	ENET PHY1	Data Receive Positive RD_P
PA18	ENET PHY1	Data Send Negative TD_N
PA19	ENET PHY1	Data Receive Positive TD_P

Table 228: Built-in Ethernet PHY pin

The bias resistor pin needs to be grounded through a 2.49K ohm resistor. The

LED pin can be used to set the PHY address, and its input value is read during power-up and reset. It is recommended to use a 4.7K ohm resistor for pull-up or pull-down. The

differential data pin is connected to a transformer.

Note that the above pins other than MDIO need to occupy the GPIO pins of the main chip, which requires that the functional selection of these pins be configured as analog functions in the main chip to avoid signal interference.

MDIO bus is multiplexed by 2 Ethernet PHYs and is led out through pins. Users can connect MIDOs of other external PHYs to it.

58.2.2 internal pins

Ethernet PHY's MII interface and clock reset signals are directly connected to the IO of the main chip inside the package, mainly using two groups of unextracted internal IO, PV and PW. See [Table 229](#) for relevant pin functions of Ethernet PHY. The IO connected to the

main chip supports mapping to ESC Port0 and Port1, as well as ENET0 on PINMUX. Specific information needs to be combined with [Table 229](#) and [Chapter 19](#).

Main chip IO	Built-in PHY chip name	Built-in PHY chip IO function
PW21	ENET PHY0	Clock Input XI
PW17	ENET PHY0	LDO Enable EN_DLDO
PV15	ENET PHY0	Receive Error RXER
PV14	ENET PHY0	Collision Detection COL
PV13	ENET PHY0	Carrier Sense CRS
PV12	ENET PHY0	Reset (active low) RSTn
PV11	ENET PHY0	Send Data Enable TXEN
PV10	ENET PHY0	Send Data Bits 3 TXD3
PV09	ENET PHY0	Send Data Bits 2 TXD2
PV08	ENET PHY0	Send Data Bit 1 TXD1
PV07	ENET PHY0	Send Data Bit 0 TXD0
PV06	ENET PHY0	Transmit Clock TXCLK

Main chip IO	Built-in PHY chip name	Built-in PHY chip IO function
PV05	ENET PHY0	Receive Clock RXCLK
PV04	ENET PHY0	Receive Data Bits 3 RXD3
PV03	ENET PHY0	Receive Data Bits 2 RXD2
PV02	ENET PHY0	Receive Data Bit 1 RXD1
PV01	ENET PHY0	Receive Data Bit 0 RXD0
PV00	ENET PHY0	Receive Data Valid RXDV
PW20	ENET PHY1	Clock Input XI
PW16	ENET PHY1	LDO Enable EN_DLDO
PW15	ENET PHY1	Receive Error RXER
PW14	ENET PHY1	Conflict Detection COL
PW13	ENET PHY1	Carrier Sense CRS
PW12	ENET PHY1	Reset (active low) RSTn
PW11	ENET PHY1	Send Data Enable TXEN
PW10	ENET PHY1	Send Data Bits 3 TXD3
PW09	ENET PHY1	Send Data Bits 2 TXD2
PW08	ENET PHY1	Send Data Bit 1 TXD1
PW07	ENET PHY1	Send Data Bit 0 TXD0
PW06	ENET PHY1	Transmit Clock TXCLK
PW05	ENET PHY1	Receive Clock RXCLK
PW04	ENET PHY1	Receive Data Bits 3 RXD3
PW03	ENET PHY1	Receive Data Bits 2 RXD2
PW02	ENET PHY1	Receive Data Bit 1 RXD1
PW01	ENET PHY1	Receive Data Bit 0 RXD0
PW00	ENET PHY1	Received Data Valid RXDV

Table 229: IO connection between main chip and built-in Ethernet PHY

PHY has an internal pull-up. If the main chip does not drive it, it will be the default to use the PHY's internal LDO to supply 1.2V power. The RSTn port of the

PHY requires the GPIO driver of the main chip. The clock input XI of

PHY requires the main chip to provide a clock source for it.

58.3 register description

Address Offset	Reg Name	Description	Reset Value
0x0000	BMCR	Mode Control Registers	0x3100
0x0001	BMSR	Mode Status Registers	0x4000
0x0028	FMLR	loopback Registers	0x0006

Table 230: ENET_PHY register list

58.3.1 BMCR (0x0)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESET	LOOPBACK	SPEED_SEL	AUTO_NEGOTIATION	POWER_DOWN	ISOLATE	RESTART_AUTONEG	DUPLEX_MODE	COLLISION_TEST	RSVD						
RW	RW	RW	RW	RW	RW	RW	RW	RW	N/A						
0	0	1	1	0	0	0	1	0	x	x	x	x	x	x	x

BMCR [15:0]

Bit Field	Name	Description
15	RESET	1: PHY reset 0 0: reset cancellation After the software reset is completed, it needs to be held for 10ms to wait for the chip to stabilize
14	LOOPBACK	Loopback Enable
13	SPEED_SEL	Speed selection 1: 100Mb/s 0: 10Mb/s
12	AUTO_NEGOTIATION	1: Enable self-negotiation, the speed_sel bit and duplex_mode bit will be ignored 0: Turn off self-negotiation
11	POWER_DOWN	1: Power off the PHY, but MDC and MDIO can access 0: Normal operation
10	ISOLATE	1: Isolate the PHY from MII/RMII, but MDC and MDIO can access 0: Works normally
9	RESTART_AUTO_NEG	1: Restart the self-negotiation operation 0: Working normally
8	DUPLEX_MODE	1: Full duplex 0: Half duplex
7	COLLISION_TEST	1: Conflict test enables 0: Normal operation When this bit is set, the COL signal will be pulled high within 512 bits after TXEN is set in response. The COL signal is reset within the 4 bit time of TXEN reset.

BMCR bit field

58.3.2 BMSR (0x1)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	BASE100_TX_FD	RSVD								AUTONEG_COMPLETE	REMOTE_FAULT	RSVD	LINK_STATUS	JABER_DETECT	RSVD
N/A	RO	N/A								RO	RC	N/A	RO	RO	N/A
x	1	x	x	x	x	x	x	x	x	0	0	x	0	0	x

BMSR [15:0]

Bit Field	Name	Description
14	BASE100_TX_FD	1: 100Base-TX full duplex enabled
5	AUTONEG_COMPLETE	1: Auto-negotiation operation completed
4	REMOTE_FAULT	1: Remote fault detected (read clear)
2	LINK_STATUS	1: Link established
1	JABER_DETECT	1: Jabber detected

BMSR bit field

58.3.3 FMLR (0x28)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD												AUTO_MDIX_EN	MDI_MODE	RSVD	
N/A												RW	RW	N/A	
x	x	x	x	x	x	x	x	x	x	x	x	x	1	1	x

FMLR [15:0]

Bit Fields	Name	Description
2	AUTO_MDIX_EN	1: Enable Auto MDI-X function
1	MDI_MODE	Modify MDI mode when auto_mdix_en is 0: 1: MDI mode 0: MDI-X mode

FMLR bit field

59 Analog Peripherals Overview and Product-Specific Information

59.1 16-Bit Analog-to-Digital Converter ADC16

This product supports 2 channels of 16-bit Analog-to-Digital Converters (ADC16). The ADC16 is an analog-to-digital converter using the successive approximation method, converting voltage analog signals on the chip pins into digital quantities.

When the ADC16 conversion precision is set to 16 bits, the maximum sampling rate is 2 MSPS.

On this product, the ADC's high-level reference voltage pin is VREFH, and the low-level reference voltage pin is VREFL.

Note: When using the internal DCDC, the 16-bit accuracy of the ADC will degrade to some extent; for precautions regarding using the on-chip ADC, please refer to the ADC-related application notes at: <https://www.hpmicro.com/design-resources/chip-information>. For power supply structures not using the internal DCDC, please see [Figure 5](#).

59.1.1 ADC Clock

On this product, the ADC's bus clock comes from CLK_TOP_AHB0.

The ADC clock for ADC0 comes from CLK_TOP_ADC0, which by default comes from CLK_TOP_AHB0.

The ADC clock for ADC1 comes from CLK_TOP_ADC1, which by default comes from CLK_TOP_AHB0.

For more information on functional clocks, please see [Section 10.3](#).

59.1.2 ADC Input Channel Assignment

Both ADC0 and ADC1 support input channels ina0~15, all from IOs.

For the specific connection between ADC input channels and IOs, please refer to [Chapter 19](#).

59.1.3 ADC Conversion Trigger Signal Connections

On this product, the trigger inputs and outputs of the 2 ADCs are connected to the Trigger Manager TRIGMUX. For specific connections, see [Subsection 36.7.1](#).

All ADC modules on this product share the same connections for their preemptive trigger input signals. Therefore, the same trigger signal can simultaneously trigger the preemptive conversion sequences of the same index on all on-chip ADCs.

59.1.4 ADC Data Forwarding Connections

Both ADCs provide ADC data output ports, which are connected to the ADC data forwarder ADC_MATRIX in the Interconnect Manager. For specific connections, see [Subsection 36.7.5](#).

Note: When using the ADC direct output function, the ADC clock must be sourced from the same clock source as the motor control system. On this product, CLK_TOP_ADCn must be selected to come from CLK_TOP_AHB0.

59.1.5 ADC Differential Mode

Adjacent ADCs can be configured into differential mode, with ADC0 and ADC1 forming a pair.

59.2 Analog Comparator Controller ACMP

This product supports 1 ACMP, comprising a total of 2 channels of analog comparators.

On this product, the reference voltage for the ACMP's internal DAC comes from the pin VREFH.

The OUT comparison result output signal of the ACMP is connected to the input of the Trigger Manager TRIGMUX.

ACMP supports window comparison, with the window control signal WIN coming from the output of the Trigger Manager TRIGMUX.

For specific connections to the Trigger Manager, please see [Subsection 36.7.1](#).

On this product, the ACMP's operating clock uses the clock source CLK_TOP_AHB0.

ACMP supports external input of DAC values, with the input coming from the DAC forwarder DAC_MATRIX. For specific connections, please see [Subsection 36.7.6](#).

Both analog comparators have their INN1~7 and INP1~7 connected to IOs. For specific IO connections, please refer to [Chapter 19](#).

59.3 Temperature Sensor

This product integrates a temperature sensor, which can be used to measure the internal temperature of the chip.

60 16-bit analog-to-digital converter ADC16

This chapter describes the main functions and characteristics of the 16-bit analog-to-digital converter ADC16:

60.1 Feature Summary

This chapter describes the main features of the 16-bit analog-to-digital converter ADC16:

- 16-bit successive approximation ADC
- Configure sampling time and conversion time to adjust accuracy and sampling rate
- Maximum 2MHz sampling rate at 16-bit accuracy
- supports single-ended and differential inputs
- Built-in DMA can directly write ADC conversion results to memory
- supports direct output of AD conversion results to the on-chip ADC repeater
- supports generation of various interrupts
- supports read conversion mode
 - Read result register directly triggers conversion
- supports periodic conversion mode
 - Built-in timer converts on a periodic basis
 - supports hardware threshold comparison and alarms for out-of-range conversion results
- supports sequence conversion mode
 - can be triggered by software or hardware
 - sequences can be up to 16
 - supports single conversion or continuous conversion
 - supports sequence loops
- supports preemption conversion mode
 - can be triggered by software or hardware
 - supports up to 12 independent preemption queues
 - Continuous conversion queues can be up to 4

60.2 Function Description

This section describes the functions of the 16-bit analog-to-digital converter ADC16. The structural block diagram of

ADC16 is shown in [Figure 96](#) :

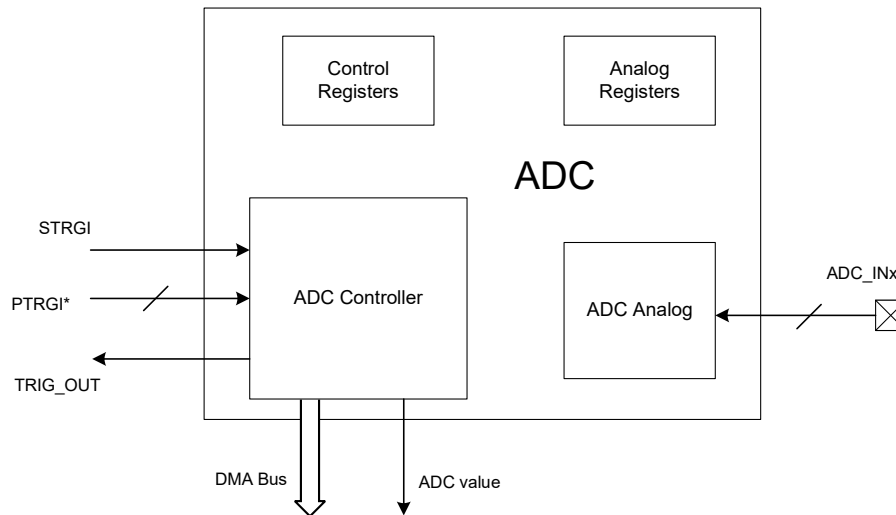


Figure 96: ADC16 Block Diagram

60.2.1 ADC basic configuration

60.2.1.1 ADC clock

ADC clock structure is shown in [Figure 97](#) :

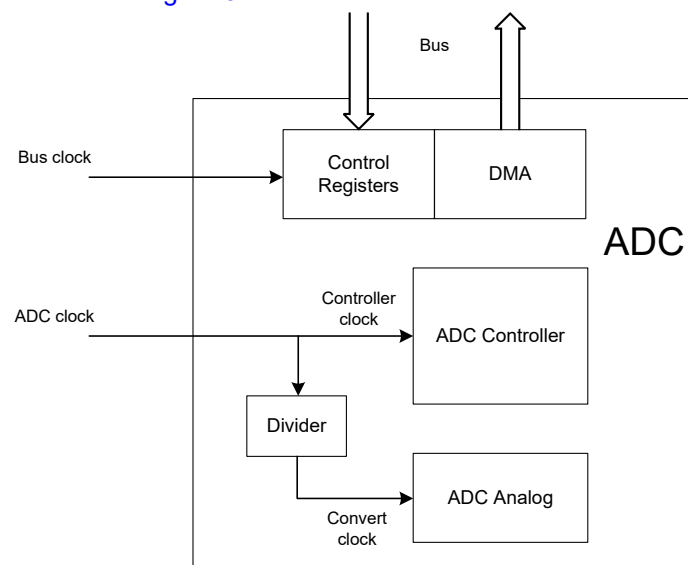


Figure 97: ADC16 clock structure diagram

ADC's clock inputs include bus clock and ADC clock. Please refer to [Section 59.1](#) for the connection of these two clocks in the SoC. The

ADC clock serves as the ADC controller clock. The

ADC conversion clock is a clock controlled by the ADC's successive approximation of the analog conversion clock. It is obtained by prescaling the ADC controller clock. The user can configure the prescaling value of the ADC conversion clock relative to the ADC controller clock by setting the CONV_CFG1 [CLOCK_DIVIDER] register bit. When the value of CONV_CFG1 [CONVCLKDIV] is 0, it means that the prescale value is/1; when the value is 1, the prescale value is/2, with a maximum of/16.

With 16-bit accuracy, in order to ensure conversion quality, the frequency of the ADC conversion clock should not be higher than 50 MHz.

60.2.1.2 ADC initialization

Calibration operations should be performed before ADC runs. The specific operation method can be performed according to relevant functions of the SDK. Registers with

offset addresses of 0x1400 and later are analog registers. When writing to these analog registers, you must first turn on the relevant clock by setting the ANA_CTRL0 [ADC_CLK_ON] register bit 1.

ADC need to be enabled, and the corresponding registers are the [REG_EN] and [BANDGAP_EN] bits of ADC16_CONFIG0.

60.2.1.3 ADC sampling and conversion configuration

Each ADC can only have 1 channel to perform sampling and conversion at the same time, and multiple channels must run in time. Each analog-to-digital conversion process of the

ADC consists of two parts: sampling and conversion. The total time of each conversion is the sum of the sampling time and conversion time. This total time determines the maximum sampling frequency of the ADC. Each channel of

can set the sampling time independently: it is configured through the [SAMPLE_CLOCK_NUMBER] bit field of the SAMPLE_CFGx register and the

[SAMPLE_CLOCK_NUMBER_SHIFT] bit field. The sampling time of

is $SAMPLE\_CLOCK\_NUMBER \times 2^{SAMPLE\_CLOCK\_NUMBER\_SHIFT}$ clock cycles. The minimum sampling time of is 1 clock cycle. This clock is the ADC conversion clock.

conversion time configuration is shared by all channels and is set in the CONV_CFG1[CONVERT_CLOCK_NUMBER] register. The conversion accuracy of

is limited by the conversion time. The following lists the required conversion time under various accuracies, that is, the number of ADC conversion clock cycles:

- 16-bit accuracy: minimum 21 clock cycles
- 12-bit accuracy: minimum 14 clock cycles
- 10-bit accuracy: minimum of 11 clock cycles
- 8-bit accuracy: minimum 9 clock cycles

It can be calculated from this that when the conversion clock frequency of the ADC16 is 50MHz and the sampling accuracy is required to be 16 bits, the total AD conversion time is:

1 clock cycle sampling time + 21 clock cycles conversion time = 440ns

Therefore, the maximum sampling rate of 16-bit precision AD conversion is about 2MHz.

can shorten the conversion time to achieve higher sampling rates when users do not need higher accuracy.

needs to note that the accuracy requirement for conversion time is the number of times the ADC conversion clock flips rather than the absolute time. That is, if the ADC conversion clock frequency decreases, it still takes 21 clock cycles to achieve 16-bit conversion accuracy. For

, note that the CONV_CFG1[CONVERT_CLOCK_NUMBER] register needs to be used in conjunction with the ADC16_CONFIG1[COV_END_CNT] register to execute correctly. The requirements must be configured as:

$$\text{COV\_END\_CNT} = 21 - \text{CONVERT\_CLOCK\_NUMBER} + 1$$

60.2.1.4 AD conversion result calculation

The AD conversion result can be read from the register or DMA buffer in different working modes, and the conversion result is 16-bit data.

When a shorter ADC conversion time is configured to reduce AD conversion accuracy, the validity of the low-order data (LSB) of the 16-bit conversion result is reduced. If the ADC conversion time is shortened to the 12-bit accuracy requirement, the upper 12 bits of the conversion result are valid values, and the lower 4 bits can be ignored.

The conversion result to the actual voltage needs to be based on the voltage value of the ADC high-level reference voltage VREFH pin. The calculation formula is:

$$V_{OUT} = AD\_RESULT \div 65536 \times V_{REFH}$$

60.2.2 ADC working mode

ADC has 4 working modes, namely:

- Read conversion mode: Software reads the register to trigger AD conversion and obtain the conversion result
- Periodic conversion mode: AD conversion is automatically executed at regular intervals
- Sequence conversion mode: Pre-defined conversion channel sequences, triggered by software or hardware
- Preemption Conversion Mode: Supports multiple hardware trigger source trigger execution, suitable for working in conjunction with motor systems

60.2.2.1 Read conversion mode

Users can intuitively read the conversion result of an ADC input channel through the read conversion mode.

_EN=1 Set the register BUF_CFG0[BUS_MODE_EN] bit to 1 to enable read conversion mode.

When the user reads the BUS_RESURTx register, an AD conversion to input channel x is triggered.

By default, that is, when the BUF_CFG0[WAITDIS] bit is 0, reading the BUS_RESURTx register blocks the CPU process. The phenomenon is that the CPU suspends responding until the conversion is completed and the register value is obtained, during which time the CPU cannot respond to any interrupt, so it is only suitable for scenarios where the CPU does not currently have any other tasks. The blocking time depends on the total AD conversion time configured for the channel.

The user can turn off the read conversion blocking by turning the BUF_CFG0[WAITDIS] bit 1. At this time, reading the BUS_RESULT register will also immediately trigger the AD conversion on input channel x, but it will not block the CPU process. The software can immediately obtain the value of the BUS_RESULT register, but the value of BUS_RESULT [CHAN_RESULT] only reflects the most recent conversion result. The software should wait for the BUS_RESULT [VALID] bit 1, indicating that the current conversion has been completed.

BUS_RESULT [CHAN_RESULT] is the new AD conversion result. The BUS_RESURTx [VALID] bit is automatically cleared after being read.

60.2.2.2 periodic conversion mode

Users can require one or more input channels of the ADC to perform periodic conversions through periodic conversion mode.

is suitable for situations where there is a need for timed transitions, or monitoring whether a channel exceeds a threshold.

Each channel can independently enable the periodic conversion mode. The user can set the length of the cycle through the PRD_CFGx[PRD] and [PRESCALE] register bits, and a cycle conversion for channel x is initiated when PRD_CFGx[PRD] is set to a non-zero value. The length of the

cycle is specifically $PRD \times 2^{PRESCALE}$ ADC controller clock cycles.

To stop periodic conversion, clear the PRD_CFGx[PRD] register. The result of the

cycle conversion is stored in the PRD_RESURTx register, and the software can read the most recent conversion result of input channel x from PRD_RESURTx.

Note that PRD_RESURTx always saves the most recent conversion result of ADC input channel x. PRD_RESURTx not only saves the results of periodic conversions, but also saves the results of other conversion modes.

Users can monitor the conversion result of input channel x by setting PRD_THSHD_CFGx[THSHDH] and PRD_THSHD_CFGx[THSHDL]. Once the conversion result is out of range (Result is greater than THSHDH or Result is less than THSHDL), the corresponding flag bit INT_STS[WDOG] bit will be set to 1, and if the corresponding interrupt is enabled, an interrupt alarm will be generated.

Note that ADC conversion result detection is not limited to periodic conversion modes, but is valid for all conversion modes. If the threshold is set incorrectly, for example, THSHDHx is set to 0, an alarm will be triggered every time the channel transition ends.

If multiple channels expire at the same time and need to start conversion, the input channel with the smaller sequence number will be converted first.

60.2.2.3 Sequence Conversion Mode

ADC supports a sequential conversion mode, in which the ADC can convert designated input channels one by one in a pre-programmed order.

is suitable for continuous conversion of one or several channels and cases where the data amount is relatively large, such as testing signal-to-noise ratio or ENOB.

The user can specify which input channel the sequence will be converted to by configuring the SEQ_QUEx registers in turn. The SEQ_QUE0 [CHAN_NUM] bit field can configure the channel sequence number for the first AD conversion after the sequence conversion is triggered; the SEQ_QUE1 [CHAN_NUM] bit field configures the channel sequence number for the second AD conversion, and so on, the SEQ_QUE15[CHNUM] bit configures the channel sequence number for the 16th AD conversion.

In addition, the user needs to configure the SEQ_CFG0 [SEQ_LEN] bit to specify the length of the conversion sequence. The conversion sequence of the ADC can be as long as 16.

The sequence start transition can be triggered by software or hardware:

- Software trigger: When the SEQ_CFG0 [SW_TRIG_EN] bit is 1, writing 1 to the SEQ_CFG0 [SW_TRIG] bit will trigger sequence conversion.
- Hardware trigger: When the SEQ_CFG0 [HW_TRIG_EN] bit is 1, a rising edge is captured on the STRGI input port, triggering a sequence transition

is triggered, the ADC will start the first AD conversion in sequence from the configuration of SEQ_QUE0. The behavior after the conversion is completed will be configured by the SEQ_CFG0 [CONT_EN] register:

- If SEQ_CFG0 [CONT_EN] is 1, the ADC will automatically start the next AD conversion according to the configuration of SEQ_QUE1 until the specified sequence length SEQ_QUE_n is reached.
- SEQ_CFG0 [CONT_EN] is 0, and the ADC will wait until the next software or hardware trigger before starting the next channel conversion.

Single conversion completion flag: If you are concerned about the completion status of one or more conversions in the sequence, you can set the corresponding SEQ_QUE_x [SEQ_INT_EN] bit to 1, then the conversion completion will set the INT_STS [SEQ_CVC] flag to 1. If the interrupt enable of this status bit is configured, an interrupt will also be generated.

Sequence conversion complete flag: When the ADC completes all conversions of the sequence (the length is set by the SEQ_CFG0 [SEQ_LEN] bit), the flag bit INT_STS [SEQ_CMPT] bit will be set to 1. If interrupt enable of this status bit is configured, an interrupt will also be generated.

Sequence cycling: When SEQ_CFG0 [CONT_EN] is 1, when the ADC completes all conversions of the sequence, its behavior is configured by the SEQ_CFG0 [RESTART_EN] register:

- SEQ_CFG0 [RESTART_EN] is 1, and the ADC will automatically restart the sequence conversion based on the configuration of SEQ_QUE0.
- SEQ_CFG0 [RESTART_EN] is 0, and the ADC needs to wait for the next software or hardware trigger before resuming sequence conversion from SEQ_QUE0.

Trigger conflict: If a new sequence conversion trigger signal is received during ADC sequence conversion, the ADC will ignore this trigger. If a software-triggered sequence conversion, the INT_STS [SEQ_SW_CFLCT] bit is set to 1. If a hardware-triggered sequence transition is triggered by STRGI, the INT_STS [SEQ_HW_CFLCT] bit is set to 1.

DMA in sequence conversion mode

ADC sequence conversion mode supports built-in DMA, which can directly write the conversion result to a user-specified buffer in memory. The user can configure the buffer start address through the register SEQ_DMA_ADDR and set the buffer size through SEQ_DMA_CFG [BUF_LEN]. The unit of BUF_LEN is 4 bytes.

The maximum value of 0xFFF in the SEQ_DMA_CFG [BUF_LEN] register corresponds to a buffer size of 16KB. If a larger buffer is needed, the SEQ_HIGH_CFG [BUF_LEN_HIGH] bit can be configured to continue expanding the buffer. BUF_LEN_HIGH is the high bit configured for the buffer size, that is, the final buffer size is:

$$Buffer\_size = BUF\_LEN\_HIGH \times 16KB + BUF\_LEN \times 4B$$

When the DMA enable bit ADC_CFG0 [ADC_AHB_EN] is 1, the results of the sequential conversion are

written to the buffer in order, and each conversion result occupies 4 Bytes of space, so the DMA write pointer continues to accumulate in the form of address +4.

If the write pointer reaches the buffer boundary, it will jump to the buffer start address and continue writing. The data structure of the AD conversion result of the

buffer is as shown in [Subsection 60.2.2.3](#) :

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CYCLE		RES						ADC_CH				SEQ_NUM				RESULT															

DMA data structure in sequence mode

Sequence mode DMA data structure description:

- CYCLE: Buffer circulation flag
- ADC_CH: ADC channel number
- SEQ_NUM: The sequence number in the sequence for this conversion
- RESULT: AD conversion result

Users can use the Cycle (CYCLE) bit in DMA data to help determine the real-time nature of AD conversion results. When the DMA writes the conversion result in the buffer in memory for the first time, it writes 1 to this field. When the entire buffer is filled, the DMA pointer cycles to point to the starting position, and this round of DMA will write 0 to the CYCLE field. When the DMA fills the buffer again, the pointer returns to the starting position again, and the CYCLE field will be written to 1 again this round.

The user can specify the DMA stop position by setting the SEQ_DMA_CFG[STOP_POS] bit and the SEQ_HIGH_CFG[STOP_POS] bit.

$$Stop\_position = STOP\_POS\_HIGH \times 16KB + STOP\_POS \times 4B$$

When the DMA pointer reaches the stop position, the INT_STS[STOP_POS] bit in the register will be set to 1. If the corresponding interrupt is enabled, an interrupt alarm will be generated, regardless of whether the SEQ_DMA_CFG[STOP_EN] bit is 1.

If the register SEQ_DMA_CFG[STOP_EN] bit is 1, the DMA will no longer write data to the buffer after the pointer reaches the stop bit. The results of this and subsequent AD conversions will be lost, and the flag bit INT_STS[SEQ_DMAABT] will be set to 1. If the corresponding interrupt is turned on, the ADC will generate an interrupt alarm.

The user can reset the DMA for sequence conversion by clearing the SEQ_DMA_CFG[DMA_RST] bit to 1 and then 0. This bit resets the DMA's pointer to the buffer start address. The CYCLE bit of DMA data is also reset to 1.

It is recommended that after the user sets the SEQ_DMA_CFG[DMA_RST] bit to 1, the software initializes and clears the buffer in memory, and then clears the SEQ_DMA_CFG [DMA_RST] bit to 0 to complete the DMA reset.

60.2.2.4 Preemption conversion mode



supports up to 12 preemption conversion queues per ADC, and each preemption conversion queue can perform up to 4 consecutive conversions after triggering.

preemption conversion is suitable for applications with high real-time requirements such as motor control systems that require conversion of specified channels at a specified time.

Preemption conversion can be triggered by software or hardware:

- Software trigger: After writing 1 to the TRG_SW_STA[TRG_SW_STA] bit, the preemption queue specified by the [TRG_SW_INDEX] bit will be started.
- Hardware trigger: Capturing a rising edge on the preemption trigger input port PTRGI* will trigger the corresponding preemption queue to start.

ADC module provides 12 hardware preemption trigger input ports, which correspond one-to-one to 12 preemption queues, such as [Table 231](#) :

Queue Sequence Number	Queue Name	Hardware Trigger Port
0	0A	PTRGI0A
1	0B	PTRGI0B
2	0C	PTRGI0C
3	1A	PTRGI1A
4	1B	PTRGI1B
5	1C	PTRGI1C
6	2A	PTRGI2A
7	2B	PTRGI2B
8	2C	PTRGI2C
9	3A	PTRGI3A
10	3B	PTRGI3B
11	3C	PTRGI3C

Table 231: ADC Preemption Queue and Hardware Trigger Port

preemption the trigger port to connect to the trigger manager of the on-chip interconnection manager. For specific information, please refer to the relevant chapters.

The 12 preemption queues correspond to the 12 CONFIG configuration registers.

users first need to configure the queue length for preemption conversions through the [TRIG_LEN] bit. The queue length is at most 4, which means that each queue can trigger up to 4 AD conversions. The [CHAN0],[CHAN1],[CHAN2], and [CHAN3] bit fields of the

configuration register can sequentially configure the AD input channel numbers of the 1st, 2nd, 3rd, and 4th conversions after triggering. After the

configuration is completed, when a software or hardware trigger event occurs, the AD conversion of the corresponding queue is triggered. Once the preemption transition starts, the entire preemption queue will be continuously converted according to the configuration of [TRIG_LEN]. Each conversion in the

preemption queue has a [INTEN] bit field. If it is set to 1, the register INT_STS [TRIG_CMPT] flag will be set to 1 after the conversion is completed. If the corresponding interrupt control bit is enabled, the ADC will generate an interrupt.

If a new preemption conversion trigger signal is received during the ADC preemption conversion process, the ADC will not respond to the new trigger. However, depending on whether the trigger source is software or hardware, the INT_STS [TRIG_SW_CFLICT] flag bit or the INT_STS[TRIG_HW_CFLICT] flag bit will be set to 1, indicating that a preemption conversion trigger conflict has occurred. If the corresponding interrupt is enabled, the ADC will generate an interrupt alarm.

When preemption transitions are triggered at the same time as other types of transitions, preemption transitions have higher priority.

If several different preemption transitions are triggered at the same time, the ADC will arbitrate in the order of PTRGI0A having the highest priority, PTRGI0B having the second priority, and PTRGI3C having the lowest priority. At the same time, the INT_STS[TRIG_HW_CFLICT] flag position is 1.

Preemption of DMA in conversion mode

ADC preemption conversion mode supports built-in DMA, which can directly write the conversion result to a user-specified buffer in memory.

When the DMA enable bit ADC_CFG0[ADC_AHB_EN] is 1, the results of preemption conversions are written to a fixed location in the buffer, and each conversion result occupies 4 Bytes of space. The data structure of the AD conversion result of the

buffer is as shown in [Table 60.2.2.4](#) :

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CYCLE		SEQ_NUM		TRIG_CH				ADC_CH				RES				RESULT															

DMA data structure in preemption mode

Sequence mode DMA data structure description:

- CYCLE: DMA always writes 1 to this bit
- TRIG_CH: Preemption queue number (0 ~ 11)
- SEQ_NUM: The sequence number in the preemption queue for this conversion (0 ~ 3)
- ADC_CH: ADC channel number
- RESULT: AD conversion result

CYCLE bit can be used to interact with software, which can clear this bit when initializing the buffer, check that this bit is high when reading the conversion result, make sure that the ADC has written a new result to the buffer, and then clear this bit in the buffer.

The user configures the target address through the register TRG_DMA_ADDR. Starting with this address, the ADC occupies a 192-byte area as a DMA buffer for preemption of the conversion results. Allocation of content in buffer is as follows: [Table 232](#) :

Serial Num-ber	Address	Description
0	TRG_DMA_ADDR + 0x0	Preempted Queue 0A The first AD conversion result and additional information
1	TRG_DMA_ADDR + 0x4	The second AD conversion result and additional information of preemption queue 0A
2	TRG_DMA_ADDR + 0x8	The third AD conversion result and additional information of preemption queue 0A
3	TRG_DMA_ADDR + 0xC	The fourth AD conversion result and additional information of preemption queue 0A
4	TRG_DMA_ADDR + 0x10	Preemption Queue 0B First AD Conversion Result and Additional Information
5	TRG_DMA_ADDR + 0x14	The second AD conversion result and additional information of preemption queue 0B
6	TRG_DMA_ADDR + 0x18	The third AD conversion result and additional information of preemption queue 0B
7	TRG_DMA_ADDR + 0x1C	The fourth AD conversion result of preemption queue 0B and additional information
8	TRG_DMA_ADDR + 0x20	Preemption Queue 0C's First AD Conversion Result and Additional Information
9	TRG_DMA_ADDR + 0x24	Preempted Queue 0C Second AD conversion result and additional information
10	TRG_DMA_ADDR + 0x28	3rd AD conversion result and additional information of preemption queue 0C
11	TRG_DMA_ADDR + 0x2C	The fourth AD conversion result of preemption queue 0C and additional information
12	TRG_DMA_ADDR + 0x30	The first AD conversion result and additional information of preemption queue 1A
13	TRG_DMA_ADDR + 0x34	Preemption Queue 1A Second AD conversion result and additional information
14	TRG_DMA_ADDR + 0x38	Preemption Queue 1A Third AD Conversion Result and Additional Information
15	TRG_DMA_ADDR + 0x3C	The fourth AD conversion result and additional information of preemption queue 1A
16	TRG_DMA_ADDR + 0x40	Preemption Queue 1B First AD conversion result and additional information
17	TRG_DMA_ADDR + 0x44	Preemption Queue 1B Second AD conversion result and additional information
18	TRG_DMA_ADDR + 0x48	3rd AD conversion result and additional information of preemption queue 1B

Serial Number	Address	Description
19	TRG_DMA_ADDR +0x4C	The fourth AD conversion result and additional information of preemption queue 1B
20	TRG_DMA_ADDR +0x50	The first AD conversion result and additional information of preemption queue 1C
21	TRG_DMA_ADDR +0x54	Preemption Queue 1C Second AD conversion result and additional information
22	TRG_DMA_ADDR +0x58	3rd AD conversion result and additional information of preemption queue 1C
23	TRG_DMA_ADDR +0x5C	The fourth AD conversion result and additional information of Preemption Queue
24	TRG_DMA_ADDR +0x60	Preemption Queue 2A's First AD Conversion Result and Additional Information
25	TRG_DMA_ADDR +0x64	Preemption Queue 2A Second AD Conversion Result and Additional Information
26	TRG_DMA_ADDR +0x68	Preemption Queue 2A Third AD Conversion Result and Additional Information
27	TRG_DMA_ADDR +0x6C	The fourth AD conversion result and additional information of preemption queue 2A
28	TRG_DMA_ADDR +0x70	Preemption Queue 2B's First AD Conversion Result and Additional Information
29	TRG_DMA_ADDR +0x74	Preemption Queue 2B Second AD conversion result and additional information
30	TRG_DMA_ADDR +0x78	3rd AD conversion result and additional information of preemption queue 2B
31	TRG_DMA_ADDR +0x7C	The fourth AD conversion result and additional information of preemption queue 2B
32	TRG_DMA_ADDR +0x80	Preemption Queue 2C First AD Conversion Result and Additional Information
33	TRG_DMA_ADDR +0x84	Preemption Queue 2C Second AD conversion result and additional information
34	TRG_DMA_ADDR +0x88	Preemption Queue 2C Third AD conversion result and additional information
35	TRG_DMA_ADDR +0x8C	The fourth AD conversion result and additional information of preemption queue 2C
36	TRG_DMA_ADDR +0x90	Preemption Queue 3A First AD Conversion Result and Additional Information
37	TRG_DMA_ADDR +0x94	Preemption Queue 3A Second AD Conversion Result and Additional Information

Serial Number	Address	Description
38	TRG_DMA_ADDR +0x98	Preemption Queue 3A Third AD Conversion Result and Additional Information
39	TRG_DMA_ADDR +0x9C	Preemption Queue 3A Fourth AD conversion result and additional information
40	TRG_DMA_ADDR +0xA0	The first AD conversion result and additional information of preemption queue 3B
41	TRG_DMA_ADDR +0xA4	Preemption Queue 3B Second AD conversion result and additional information
42	TRG_DMA_ADDR +0xA8	Preemption Queue 3B Third AD conversion result and additional information
43	TRG_DMA_ADDR + 0xAC	Preemption Queue 3B Fourth AD conversion result and additional information
44	TRG_DMA_ADDR +0xB0	Preemption Queue 3C's First AD Conversion Result and Additional Information
45	TRG_DMA_ADDR + 0xB4	Preemption Queue 3C Second AD conversion result and additional information
46	TRG_DMA_ADDR +0xB8	3C's third AD conversion result and additional information
47	TRG_DMA_ADDR + 0xBC	Preemption Queue 3C Fourth AD conversion result and additional information

Table 232: ADC preemption conversion DMA data buffer allocation

60.2.3 ADC differential mode

ADC supports differential mode, in which two ADC modules (such as ADC0 and ADC1, or ADC2 and ADC3) need to be used at the same time.

The two ADCs need to be configured in the same working mode and at the same sampling rate, ensuring that they perform exactly the same AD conversion operation after being sampled by hardware at the same time. There are no requirements for the conversion channels used by each of the two ADCs of

One of the ADCs needs to be configured in Master mode, that is, set CONV_CFG0[DIFF_MASTER] and [ADC_DIFF_MODE] to 1. When the sample conversion is completed, the Master ADC will subtract the result from the other ADC and use it as the final conversion result (stored in an appropriate register or DMA buffer)

There are three modes for differential data results, which are configured through the CONV_CFG0 register:

- Full amplitude mode: When [FULL_RESOLUTION] is set to 1. The output is a 16-bit unsigned number. If the subtraction of two adc results is less than 0, the output is 0. This mode can be used to ground one end to eliminate jitter on the ground signal.
- Positive and negative mode: When [FULL_RESOLUTION] is set to 0 and [POS_MODE] is set to 0. The output is a 16-bit signed number and the ADC precision is 15-bit. If the two ADC results are the same,

the output is 0, 0x8000 represents the smallest negative value (-32768), and 0x7FFF is the largest positive value (32767).

- Positive difference mode: When [FULL_RESOLUTION] is set to 0 and [POS_MODE] is set to 1. The output is a 16-bit unsigned number. If the two ADC results are the same, the output is 0x8000, which is equivalent to the result of +32768 in positive and negative mode.

60.2.4 Direct output of AD conversion results

When the register ANA_CTRL0[MOTO_EN] bit is set to 1, the conversion result will be output to the ADC data forwarder of the interconnection manager every time the AD conversion is completed. Please refer to the interconnection manager chapter for specific connections.

60.2.5 ADC interrupt

ADC supports the following status flag bits in the INT_STS register:

- Completion flag:
 - TRIG_CMPT -Preemption Conversion Completed: In preemption conversion mode, a conversion completion bit in the preemption queue is post-set.
 - SEQ_CMPT-Sequence conversion complete: In sequence conversion mode, the bit is post-set after the entire sequence conversion is completed.
 - SEQ_CVC -Single conversion completion in sequence mode: Bit after the completion of a certain AD conversion in the sequence in sequence conversion mode.
- Conflict flag:
 - TRIG_SW_CFLCT -Preemption conversion software trigger conflict: Set when the ADC preemption conversion queue is in progress and the software triggers a preemption conversion again.
 - TRIG_HW_CFLCT -Preemption conversion hardware trigger conflict: Set when the ADC preemption conversion queue is in progress, the hardware triggers a preemption conversion again, or when multiple hardware triggers occur at the same time.
 - TREAD_CFLCT -Read conversion conflict: If the WAIT_DIS bit is 1, this bit will be 1 after the software reads the read result register of other channels during AD conversion.
 - SEQ_SW_CFLCT -Sequence conversion software trigger conflict: Set when the ADC sequence conversion is in progress and the software triggers a sequence conversion again.
 - SEQ_HW_CFLCT -Sequence conversion hardware trigger conflict: Set when the ADC sequence conversion is in progress and the hardware triggers a sequence conversion again.
- DMA flag:
 - SEQ_DMAABT -Sequence conversion DMA stop: When SEQ_DMA_CFG[STOP_EN], the DMA write pointer in sequence conversion mode reaches the stop bit.
 - STOP_POS -Sequential conversion DMA reaches stop bit: Set when the DMA write pointer reaches stop bit in sequential conversion mode
 - DMA_FIFO_FULL - DMA FIFO overflow: Set when the DMA writes data to the buffer at a speed lower than the AD conversion data generation speed, causing the DMA FIFO to overflow.
 - AHB_ERR - DMA bus error: Set when an error response occurs when a DMA writes a buffer.
- Excess flag:

- WDOG-AD conversion result exceeds limit: Set when the AD conversion result exceeds the threshold range specified in the THSHD register, one bit corresponding to each AD channel.

The user can clear the flag bit by writing a 1 to each flag bit in the INT_STS register. The

ADC can manage the interrupt enablement of these status flag bits through the INT_EN register. If the corresponding interrupt enable bit is 1, the ADC will issue an interrupt signal when the flag bit is 1.

_HAS_TRIG_OUT=1 ADC supports trigger output. The following status flags in the INT_STS register can set the trigger output, requiring the corresponding trigger output bit to be enabled in the TRIGMUX_EN register:

- SEQ_CMPT
- SEQ_CVC
- STOP_POS
- WDOG

The trigger output pulls the TRIG_OUT signal high, and TRIG_OUT is connected to the trigger manager of the interconnection manager.

60.3 ADC16 register list

The register list of ADC16 is as follows:

ADC0 base address: 0xF0100000

ADC1 base address: 0xF0104000

Offset	Register Name	Description	Reset Value
0x0000	CONFIG[TRG0A]		0x00000000
0x0004	CONFIG[TRG0B]		0x00000000
0x0008	CONFIG[TRG0C]		0x00000000
0x000C	CONFIG[TRG1A]		0x00000000
0x0010	CONFIG[TRG1B]		0x00000000
0x0014	CONFIG[TRG1C]		0x00000000
0x0018	CONFIG[TRG2A]		0x00000000
0x001C	CONFIG[TRG2B]		0x00000000
0x0020	CONFIG[TRG2C]		0x00000000
0x0024	CONFIG[TRG3A]		0x00000000
0x0028	CONFIG[TRG3B]		0x00000000
0x002C	CONFIG[TRG3C]		0x00000000
0x0030	TRG_DMA_ADDR		0x00000000
0x0034	TRG_SW_STA		0x00000000
0x0400	BUS_RESULT[CHN0]		0x00000000
0x0404	BUS_RESULT[CHN1]		0x00000000
0x0408	BUS_RESULT[CHN2]		0x00000000
0x040C	BUS_RESULT[CHN3]		0x00000000
0x0410	BUS_RESULT[CHN4]		0x00000000
0x0414	BUS_RESULT[CHN5]		0x00000000

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

60 16-BIT ANALOG-TO-DIGITAL CONVERTER

ADC16

Offset	Register Name	Description	Reset Value
0x0418	BUS_RESULT[CHN6]		0x00000000
0x041C	BUS_RESULT[CHN7]		0x00000000
0x0420	BUS_RESULT[CHN8]		0x00000000
0x0424	BUS_RESULT[CHN9]		0x00000000
0x0428	BUS_RESULT[CHN10]		0x00000000
0x042C	BUS_RESULT[CHN11]		0x00000000
0x0430	BUS_RESULT[CHN12]		0x00000000
0x0434	BUS_RESULT[CHN13]		0x00000000
0x0438	BUS_RESULT[CHN14]		0x00000000
0x043C	BUS_RESULT[CHN15]		0x00000000
0x0500	BUF_CFG0		0x00000000
0x0800	SEQ_CFG0		0x00000000
0x0804	SEQ_DMA_ADDR		0x00000000
0x0808	SEQ_WR_ADDR		0x00000000
0x080C	SEQ_DMA_CFG		0x00000000
0x0810	SEQ_QUEUE[CFG0]		0x00000000
0x0814	SEQ_QUEUE[CFG1]		0x00000000
0x0818	SEQ_QUEUE[CFG2]		0x00000000
0x081C	SEQ_QUEUE[CFG3]		0x00000000
0x0820	SEQ_QUEUE[CFG4]		0x00000000
0x0824	SEQ_QUEUE[CFG5]		0x00000000
0x0828	SEQ_QUEUE[CFG6]		0x00000000
0x082C	SEQ_QUEUE[CFG7]		0x00000000
0x0830	SEQ_QUEUE[CFG8]		0x00000000
0x0834	SEQ_QUEUE[CFG9]		0x00000000
0x0838	SEQ_QUEUE[CFG10]		0x00000000
0x083C	SEQ_QUEUE[CFG11]		0x00000000
0x0840	SEQ_QUEUE[CFG12]		0x00000000
0x0844	SEQ_QUEUE[CFG13]		0x00000000
0x0848	SEQ_QUEUE[CFG14]		0x00000000
0x084C	SEQ_QUEUE[CFG15]		0x00000000
0x0850	SEQ_HIGH_CFG		0x00000000
0x0C00	PRD_CFG[CHN0][PRD_CFG]		0x00000000
0x0C04	PRD_CFG[CHN0][PRD_THSHD_CFG]		0x00000000
0x0C08	PRD_CFG[CHN0][PRD_RESULT]		0x00000000
0x0C10	PRD_CFG[CHN1][PRD_CFG]		0x00000000
0x0C14	PRD_CFG[CHN1][PRD_THSHD_CFG]		0x00000000
0x0C18	PRD_CFG[CHN1][PRD_RESULT]		0x00000000
0x0C20	PRD_CFG[CHN2][PRD_CFG]		0x00000000
0x0C24	PRD_CFG[CHN2][PRD_THSHD_CFG]		0x00000000

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

60 16-BIT ANALOG-TO-DIGITAL CONVERTER

ADC16

Offset	Register Name	Description	Reset Value
0x0C28	PRD_CFG[CHN2][PRD_RESULT]		0x00000000
0x0C30	PRD_CFG[CHN3][PRD_CFG]		0x00000000
0x0C34	PRD_CFG[CHN3][PRD_THSHD_CFG]		0x00000000
0x0C38	PRD_CFG[CHN3][PRD_RESULT]		0x00000000
0x0C40	PRD_CFG[CHN4][PRD_CFG]		0x00000000
0x0C44	PRD_CFG[CHN4][PRD_THSHD_CFG]		0x00000000
0x0C48	PRD_CFG[CHN4][PRD_RESULT]		0x00000000
0x0C50	PRD_CFG[CHN5][PRD_CFG]		0x00000000
0x0C54	PRD_CFG[CHN5][PRD_THSHD_CFG]		0x00000000
0x0C58	PRD_CFG[CHN5][PRD_RESULT]		0x00000000
0x0C60	PRD_CFG[CHN6][PRD_CFG]		0x00000000
0x0C64	PRD_CFG[CHN6][PRD_THSHD_CFG]		0x00000000
0x0C68	PRD_CFG[CHN6][PRD_RESULT]		0x00000000
0x0C70	PRD_CFG[CHN7][PRD_CFG]		0x00000000
0x0C74	PRD_CFG[CHN7][PRD_THSHD_CFG]		0x00000000
0x0C78	PRD_CFG[CHN7][PRD_RESULT]		0x00000000
0x0C80	PRD_CFG[CHN8][PRD_CFG]		0x00000000
0x0C84	PRD_CFG[CHN8][PRD_THSHD_CFG]		0x00000000
0x0C88	PRD_CFG[CHN8][PRD_RESULT]		0x00000000
0x0C90	PRD_CFG[CHN9][PRD_CFG]		0x00000000
0x0C94	PRD_CFG[CHN9][PRD_THSHD_CFG]		0x00000000
0x0C98	PRD_CFG[CHN9][PRD_RESULT]		0x00000000
0x0CA0	PRD_CFG[CHN10][PRD_CFG]		0x00000000
0x0CA4	PRD_CFG[CHN10][PRD_THSHD_CFG]		0x00000000
0x0CA8	PRD_CFG[CHN10][PRD_RESULT]		0x00000000
0x0CB0	PRD_CFG[CHN11][PRD_CFG]		0x00000000
0x0CB4	PRD_CFG[CHN11][PRD_THSHD_CFG]		0x00000000
0x0CB8	PRD_CFG[CHN11][PRD_RESULT]		0x00000000
0x0CC0	PRD_CFG[CHN12][PRD_CFG]		0x00000000
0x0CC4	PRD_CFG[CHN12][PRD_THSHD_CFG]		0x00000000
0x0CC8	PRD_CFG[CHN12][PRD_RESULT]		0x00000000
0x0CD0	PRD_CFG[CHN13][PRD_CFG]		0x00000000
0x0CD4	PRD_CFG[CHN13][PRD_THSHD_CFG]		0x00000000
0x0CD8	PRD_CFG[CHN13][PRD_RESULT]		0x00000000
0x0CE0	PRD_CFG[CHN14][PRD_CFG]		0x00000000
0x0CE4	PRD_CFG[CHN14][PRD_THSHD_CFG]		0x00000000
0x0CE8	PRD_CFG[CHN14][PRD_RESULT]		0x00000000
0x0CF0	PRD_CFG[CHN15][PRD_CFG]		0x00000000
0x0CF4	PRD_CFG[CHN15][PRD_THSHD_CFG]		0x00000000
0x0CF8	PRD_CFG[CHN15][PRD_RESULT]		0x00000000

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

60 16-BIT ANALOG-TO-DIGITAL CONVERTER

ADC16

Offset	Register Name	Description	Reset Value
0x1000	SAMPLE_CFG[CHN0]		0x00000001
0x1004	SAMPLE_CFG[CHN1]		0x00000001
0x1008	SAMPLE_CFG[CHN2]		0x00000001
0x100C	SAMPLE_CFG[CHN3]		0x00000001
0x1010	SAMPLE_CFG[CHN4]		0x00000001
0x1014	SAMPLE_CFG[CHN5]		0x00000001
0x1018	SAMPLE_CFG[CHN6]		0x00000001
0x101C	SAMPLE_CFG[CHN7]		0x00000001
0x1020	SAMPLE_CFG[CHN8]		0x00000001
0x1024	SAMPLE_CFG[CHN9]		0x00000001
0x1028	SAMPLE_CFG[CHN10]		0x00000001
0x102C	SAMPLE_CFG[CHN11]		0x00000001
0x1030	SAMPLE_CFG[CHN12]		0x00000001
0x1034	SAMPLE_CFG[CHN13]		0x00000001
0x1038	SAMPLE_CFG[CHN14]		0x00000001
0x103C	SAMPLE_CFG[CHN15]		0x00000001
0x1100	CONV_CFG0		0x00000000
0x1104	CONV_CFG1		0x00000000
0x1108	ADC_CFG0		0x00000000
0x1110	INT_STS		0x00000000
0x1114	INT_EN		0x00000000
0x1118	TRIGMUX_EN		0x00000000
0x111C	TRG_CMPT_FLAG		0x00000000
0x1200	ANA_CTRL0		0x00000000
0x1210	ANA_STATUS		0x00000000
0x1400	ADC16_PARAMS[ADC16_PARA00]		0x0000
0x1402	ADC16_PARAMS[ADC16_PARA01]		0x0000
0x1404	ADC16_PARAMS[ADC16_PARA02]		0x0000
0x1406	ADC16_PARAMS[ADC16_PARA03]		0x0000
0x1408	ADC16_PARAMS[ADC16_PARA04]		0x0000
0x140A	ADC16_PARAMS[ADC16_PARA05]		0x0000
0x140C	ADC16_PARAMS[ADC16_PARA06]		0x0000
0x140E	ADC16_PARAMS[ADC16_PARA07]		0x0000
0x1410	ADC16_PARAMS[ADC16_PARA08]		0x0000
0x1412	ADC16_PARAMS[ADC16_PARA09]		0x0000
0x1414	ADC16_PARAMS[ADC16_PARA10]		0x0000
0x1416	ADC16_PARAMS[ADC16_PARA11]		0x0000
0x1418	ADC16_PARAMS[ADC16_PARA12]		0x0000
0x141A	ADC16_PARAMS[ADC16_PARA13]		0x0000
0x141C	ADC16_PARAMS[ADC16_PARA14]		0x0000

ADC16

Offset	Register Name	Description	Reset Value
0x141E	ADC16_PARAMS[ADC16_PARA15]		0x0000
0x1420	ADC16_PARAMS[ADC16_PARA16]		0x0000
0x1422	ADC16_PARAMS[ADC16_PARA17]		0x0000
0x1424	ADC16_PARAMS[ADC16_PARA18]		0x0000
0x1426	ADC16_PARAMS[ADC16_PARA19]		0x0000
0x1428	ADC16_PARAMS[ADC16_PARA20]		0x0000
0x142A	ADC16_PARAMS[ADC16_PARA21]		0x0000
0x142C	ADC16_PARAMS[ADC16_PARA22]		0x0000
0x142E	ADC16_PARAMS[ADC16_PARA23]		0x0000
0x1430	ADC16_PARAMS[ADC16_PARA24]		0x0000
0x1432	ADC16_PARAMS[ADC16_PARA25]		0x0000
0x1434	ADC16_PARAMS[ADC16_PARA26]		0x0000
0x1436	ADC16_PARAMS[ADC16_PARA27]		0x0000
0x1438	ADC16_PARAMS[ADC16_PARA28]		0x0000
0x143A	ADC16_PARAMS[ADC16_PARA29]		0x0000
0x143C	ADC16_PARAMS[ADC16_PARA30]		0x0000
0x143E	ADC16_PARAMS[ADC16_PARA31]		0x0000
0x1440	ADC16_PARAMS[ADC16_PARA32]		0x0000
0x1442	ADC16_PARAMS[ADC16_PARA33]		0x0000
0x1444	ADC16_CONFIG0		0x00000000
0x1460	ADC16_CONFIG1		0x00000100

Table 233: ADC16 Register List

60.4 ADC16 register description

The register of ADC16 is described in detail as follows:

60.4.1 CONFIG (0x0 + 0x4 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRIG_LEN	INTEN3				CHAN3			RSVD	INTEN2				CHAN2			RSVD	INTEN1				CHAN1			RSVD	QUEUE_LEN	INTEN0				CHAN0	
WO	RW				RW			N/A	RW				RW			N/A	RW				RW			N/A	RW	RW				RW	
0	0	0	0	0	0	0	0	x	x	0	0	0	0	0	0	x	x	0	0	0	0	0	0	x	0	0	0	0	0	0	0

CONFIG [31:0]

Bits	Name	Description
31-30	TRIG_LEN	length for current trigger, can up to 4 conversions for one trigger, from 0 to 3
29	INTEN3	interrupt enable for 4th conversion

Bits	Name	Description
28-24	CHAN3	channel number for 4th conversion
21	INTEN2	interrupt enable for 3rd conversion
20-16	CHAN2	channel number for 3rd conversion
13	INTEN1	interrupt enable for 2nd conversion
12-8	CHAN1	channel number for 2nd conversion
6	QUEUE_EN	preemption queue enable control
5	INTEN0	interrupt enable for 1st conversion
4-0	CHAN0	channel number for 1st conversion

CONFIG Bits

60.4.2 TRG_DMA_ADDR (0x30)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRG_DMA_ADDR																RW														RSVD	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x

TRG_DMA_ADDR [31:0]

Bits	Name	Description
31-2	TRG_DMA_ADDR	buffer start address for trigger queue, 192byte total, 16 bytes for each trigger (4 bytes for each conversion)

TRG_DMA_ADDR Bits

60.4.3 TRG_SW_STA (0x34)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
RSVD																												TRG_SW_STA	TRIG_SW_INDEX				
N/A																												RW	RW				
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0			

TRG_SW_STA [31:0]

Bits	Name	Description
4	TRG_SW_STA	SW trigger start bit, HW will clear it after all conversions(up to 4) finished. SW should make sure it's 0 before set it.
3-0	TRIG_SW_INDEX	which trigger for the SW trigger 0 for trig0a, 1 for trig0b... 3 for trig1a, ...11 for trig3c

TRG_SW_STA Bits

60.4.4 BUS_RESULT (0x400 + 0x4 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSD															VALID	CHAN_RESULT															
N/A															RO	RO															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BUS_RESULT [31:0]

Bits	Name	Description
16	VALID	set after conversion finished if wait_dis is set, cleared after software read. The first time read with 0 will trigger one new conversion. If SW read other channel when one channel conversion is in progress, it will not trigger new conversion at other channel, and will get old result with valid 0, also with read_cflct interrupt status bit set. the result may not realtime if software read once and wait long time to read again
15-0	CHAN_RESULT	read this register will trigger one adc conversion. If wait_dis bit is set, SW will get the latest conversion result(not current one) with valid bit is 0, SW need polling valid bit till it's set to get current result If wait_dis bit is 0, SW can get the current conversion result with holding the bus, valid bit is always set at this mode. this is not recommended if channel sample time is too long

BUS_RESULT Bits

60.4.5 BUF_CFG0 (0x500)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RSVD																															BUS_MODE_EN	WAIT_DIS
N/A																															RW	RW
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0		

BUF_CFG0 [31:0]

Bits	Name	Description
1	BUS_MODE_EN	bus mode enable
0	WAIT_DIS	set to disable read waiting, get result immediately but maybe not current conversion result.

BUF_CFG0 Bits

60.4.6 SEQ_CFG0 (0x800)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
CYCLE	RSVD																				SEQ_LEN				RSVD			RESTART_EN	CONT_EN	SW_TRIG	SW_TRIG_EN	HW_TRIG_EN
RO	N/A																				RW				N/A			RW	RW	WO	RW	RW
0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	x	x	x	0	0	0	0	0	

SEQ_CFG0 [31:0]

Bits	Name	Description
31	CYCLE	current dma write cycle bit
11-8	SEQ_LEN	sequence queue length, 0 for one, 0xF for 16
4	RESTART_EN	if set together with cont_en, HW will continue process the whole queue after trigger once. If cont_en is 0, this bit is not used
3	CONT_EN	if set, HW will continue process the queue till end(seq_len) after trigger once
2	SW_TRIG	SW trigger, pulse signal, cleared by HW one cycle later
1	SW_TRIG_EN	set to enable SW trigger
0	HW_TRIG_EN	set to enable external HW trigger, only trigger on posedge

SEQ_CFG0 Bits

60.4.7 SEQ_DMA_ADDR (0x804)

ADC16

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAR_ADDR																RSVD															
RW																N/A															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x

SEQ_DMA_ADDR [31:0]

Bits	Name	Description
31-2	TAR_ADDR	dma target address, should be 4-byte aligned

SEQ_DMA_ADDR Bits

60.4.8 SEQ_WR_ADDR (0x808)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								SEQ_WR_POINTER																							
N/A								RO																							
x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SEQ_WR_ADDR [31:0]

Bits	Name	Description
23-0	SEQ_WR_POINTER	HW update this field after each dma write, it indicate the next dma write pointer. dma write address is (tar_addr+seq_wr_pointer)*4

SEQ_WR_ADDR Bits

60.4.9 SEQ_DMA_CFG (0x80C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD				STOP_POS												RSVD		DMA_RST	STOP_EN	BUF_LEN											
N/A				RW												N/A		RW	RW	RW											
x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SEQ_DMA_CFG [31:0]

Bits	Name	Description
27-16	STOP_POS	if stop_en is set, SW is responsible to update this field to the next read point, HW should not write data to this point since it's not read out by SW yet
13	DMA_RST	set this bit will reset HW dma write pointer to seq_dma_addr, and set HW cycle bit to 1. dma is halted if this bit is set. SW should clear all cycle bit in buffer to 0 before clear dma_rst
12	STOP_EN	set to stop dma if reach the stop_pos
11-0	BUF_LEN	dma buffer length, after write to (tar_addr[31:2]+buf_len)*4, the next dma address will be tar_addr[31:2]*4 0 for 4byte; 0xFFF for 16kbyte.

SEQ_DMA_CFG Bits

60.4.10 SEQ_QUE (0x810 + 0x4 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								SEQ_INT_EN	CHAN_NUM_4_0						
N/A																													RW	RW	
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	

SEQ_QUE [31:0]

Bits	Name	Description
5	SEQ_INT_EN	interrupt enable for current conversion
4-0	CHAN_NUM_4_0	channel number for current conversion

SEQ_QUE Bits

60.4.11 SEQ_HIGH_CFG (0x850)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								STOP_POS_HIGH												BUF_LEN_HIGH											
N/A								RW												RW											
x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

SEQ_HIGH_CFG [31:0]

Bits	Name	Description
23-12	STOP_POS_HIGH	
11-0	BUF_LEN_HIGH	

SEQ_HIGH_CFG Bits

60.4.12 PRD_CFG[PRD_CFG] (0xC00 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																			PRESCALE				PRD								
N/A																			RW				RW								
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	

PRD_CFG[PRD_CFG] [31:0]

Bits	Name	Description
12-8	PRESCALE	0: 1xclock, 1: 2x, 2: 4x, 3: 8x,...,15: 32768x,...,31: 2Gx
7-0	PRD	conver period, with prescale. Set to 0 means disable current channel

PRD_CFG[PRD_CFG] Bits

60.4.13 PRD_CFG[PRD_THSHD_CFG] (0xC04 + 0x10 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
THSHDH																THSHDL															
RW																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

PRD_CFG[PRD_THSHD_CFG] [31:0]

Bits	Name	Description
31-16	THSHDH	threshold high, assert interrupt(if enabled) if result exceed high or low.
15-0	THSHDL	threshold low

PRD_CFG[PRD_THSHD_CFG] Bits

60.4.14 PRD_CFG[PRD_RESULT] (0xC08 + 0x10 * n)

ADC16

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																CHAN_RESULT															
N/A																RO															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PRD_CFG[PRD_RESULT] [31:0]

Bits	Name	Description
15-0	CHAN_RESULT	adc convert result, update after each valid conversion. it may be updated period according to config, also may be updated due to other queue convert the same channel

PRD_CFG[PRD_RESULT] Bits

60.4.15 SAMPLE_CFG (0x1000 + 0x4 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																SAMPLE_CLOCK_NUMBER_SHIFT				SAMPLE_CLOCK_NUMBER											
N/A																RW				RW											
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	1

SAMPLE_CFG [31:0]

Bits	Name	Description
11-9	SAMPLE_CLOCK_NUMBER_SHIFT	shift for sample clock number
8-0	SAMPLE_CLOCK_NUMBER	sample clock number, base on clock_period, default one period

SAMPLE_CFG Bits

60.4.16 CONV_CFG0 (0x1100)

ADC16

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																												FULL_RESOLUTION	POS_MODE	DIFF_MASTER	ADC_DIFF_MODE
N/A																												RW	RW	RW	RW
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

CONV_CFG0 [31:0]

Bits	Name	Description
3	FULL_RESOLUTION	in differential mode, if result is less than 0, then output 0;
2	POS_MODE	0: result is from -32768 to +32767(result 0 if two adc results are same); 1: result is from 0 to 65535 (result 32768 if two adc results are same);
1	DIFF_MASTER	set to act as master, will use the result of current adc minus another adc result as final result. Refer to soc UM for which is the another adc.
0	ADC_DIFF_MODE	set to enable differential mode

CONV_CFG0 Bits

60.4.17 CONV_CFG1 (0x1104)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								CONVERT_CLOCK_NUMBER				CLOCK_DIVIDER			
N/A																								RW				RW			
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	

CONV_CFG1 [31:0]

Bits	Name	Description
8-4	CONVERT_CLOCK_NUMBER	convert clock numbers, set to 21 (0x15) for 16bit mode, which means convert need 21 adc clock cycles(based on clock after divider); user can use small value to get faster conversion, but less accuracy, need to config cov_end_cnt at adc16_config1 also. Ex: use 200MHz bus clock for adc, set sample_clock_number to 4, sample_clock_number_shift to 0, convert_clk_number to 21 for 16bit mode, clock_divder to 3, then each ADC conversion(plus sample) need 25 cycles(50MHz).
3-0	CLOCK_DIVIDER	clock_period, N half clock cycle per half adc cycle 0 for same adc_clk and bus_clk, 1 for 1:2, 2 for 1:3, ... 15 for 1:16 Note: set to 2 can generate 66.7MHz adc_clk at 200MHz bus_clk

CONV_CFG1 Bits

60.4.18 ADC_CFG0 (0x1108)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEL_SYNC_AHB	RSVD	ADC_AHB_EN	RSVD																												PORT3_REALTIME
RW	N/A	RW	N/A																												RW
0	x	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0

ADC_CFG0 [31:0]

Bits	Name	Description
31	SEL_SYNC_AHB	set to 1 will enable sync AHB bus, to get better bus performance. Adc_clk must to be set to same as bus clock at this mode
29	ADC_AHB_EN	set to 1 to enable ADC DMA to write data to soc memory bus, for trig queue and seq queue;
0	PORT3_REALTIME	set to enable trg queue stop other queues

ADC_CFG0 Bits

60.4.19 INT_STS (0x1110)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																						
TRIG_CMPT		TRIG_SW_CFLCT		TRIG_HW_CFLCT		READ_CFLCT		SEQ_SW_CFLCT		SEQ_HW_CFLCT		SEQ_DMAABT		SEQ_CMPT		SEQ_OVC		DMA_FIFO_FULL		AHB_ERR		STOP_POS		RSVD															WDOG														
RW1C		RW1C		RW1C		RW1C		RW1C		RW1C		RW1C		RW1C		RW1C		RW1C		RW1C		RW1C																															
0	0	0	0	0	0	0	0	0	0	0	0	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																					

INT_STS [31:0]

Bits	Name	Description
31	TRIG_CMPT	interrupt for one trigger conversion complete if enabled
30	TRIG_SW_CFLCT	
29	TRIG_HW_CFLCT	
28	READ_CFLCT	read conflict interrupt, set if wait_dis is set, one conversion is in progress, SW read another channel
27	SEQ_SW_CFLCT	sequence queue conflict interrupt, set if HW or SW trigger received during conversion
26	SEQ_HW_CFLCT	
25	SEQ_DMAABT	dma abort interrupt, set if sequence dma write pointer reaches sw read pointer if stop_en is set
24	SEQ_CMPT	the whole sequence complete interrupt
23	SEQ_CVC	one conversion complete in seq_queue if related seq_int_en is set
22	DMA_FIFO_FULL	DMA fifo full interrupt, user need to check clock frequency if it's set.
21	AHB_ERR	set if got hresp=1, generally caused by wrong trg_dma_addr or seq_dma_addr
20	STOP_POS	
15-0	WDOG	set if one chanel watch dog event triggered

INT_STS Bits

60.4.20 INT_EN (0x1114)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRIG_CMPT	TRIG_SW_CFLCT	TRIG_HW_CFLCT	READ_CFLCT	SEQ_SW_CFLCT	SEQ_HW_CFLCT	SEQ_DMAABT	SEQ_CMPT	SEQ_CVC	DMA_FIFO_FULL	AHB_ERR	STOP_POS	RSVD				WDOG															

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

60 16-BIT ANALOG-TO-DIGITAL CONVERTER

ADC16

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	N/A				RW															
0	0	0	0	0	0	0	0	0	0	0	0	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

INT_EN [31:0]

Bits	Name	Description
31	TRIG_CMPT	interrupt for one trigger conversion complete if enabled
30	TRIG_SW_CFLCT	
29	TRIG_HW_CFLCT	
28	READ_CFLCT	read conflict interrupt, set if wait_dis is set, one conversion is in progress, SW read another channel
27	SEQ_SW_CFLCT	sequence queue conflict interrupt, set if HW or SW trigger received during conversion
26	SEQ_HW_CFLCT	
25	SEQ_DMAABT	dma abort interrupt, set if sequence dma write pointer reaches sw read pointer if stop_en is set
24	SEQ_CMPT	the whole sequence complete interrupt
23	SEQ_CVC	one conversion complete in seq_queue if related seq_int_en is set
22	DMA_FIFO_FULL	DMA fifo full interrupt, user need to check clock frequency if it's set.
21	AHB_ERR	set if got hresp=1, generally caused by wrong trg_dma_addr or seq_dma_addr
20	STOP_POS	
15-0	WDOG	set if one chanel watch dog event triggered

INT_EN Bits

60.4.21 TRIGMUX_EN (0x1118)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD							SEQ_CMPT	SEQ_CVC	RSVD		STOP_POS	RSVD				WDOG															
N/A							RW	RW	N/A		RW	N/A				RW															
x	x	x	x	x	x	x	0	0	x	x	0	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TRIGMUX_EN [31:0]

Bits	Name	Description
24	SEQ_CMPT	
23	SEQ_CVC	

Bits	Name	Description
20	STOP_POS	
15-0	WDOG	

TRIGMUX_EN Bits

60.4.22 TRG_CMPT_FLAG (0x111C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD												TRG_CMPT_FLAG																			
N/A												RW1C																			
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0

TRG_CMPT_FLAG [31:0]

Bits	Name	Description
11-0	TRG_CMPT_FLAG	

TRG_CMPT_FLAG Bits

60.4.23 ANA_CTRL0 (0x1200)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
MOTO_EN	RSVD																		ADC_CLK_ON	RSVD										STARTCAL	RSVD	
RW	N/A																		RW	N/A										RW	N/A	
0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	x	x	x	x	x	x	x	x	0	x	x		

ANA_CTRL0 [31:0]

Bits	Name	Description
31	MOTO_EN	"set to enable moto_soc and moto_valid. Should use AHB clock for adc, this bit can be used avoid async output"
12	ADC_CLK_ON	set to enable adc clock to analog, Software should set this bit before access to any adc16_* register. MUST set clock_period to 0 or 1 for adc16 reg access

HPM5E00 Series

32-bit High-Performance RISC-V MCU User Manual Rev0.3

60 16-BIT ANALOG-TO-DIGITAL CONVERTER

ADC16

Bits	Name	Description
2	STARTCAL	set to start the offset calibration cycle (Active H). user need to clear it after setting it.

ANA_CTRL0 Bits

60.4.24 ANA_STATUS (0x1210)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																							CALON	RSVD							
N/A																							RW	N/A							
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	x	x	x	x	x	x	x	

ANA_STATUS [31:0]

Bits	Name	Description
7	CALON	Indicates if the ADC is in calibration mode (Active H).

ANA_STATUS Bits

60.4.25 ADC16_PARAMS (0x1400 + 0x2 * n)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PARAM_VAL															
RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ADC16_PARAMS [15:0]

Bits	Name	Description
15-0	PARAM_VAL	

ADC16_PARAMS Bits

60.4.26 ADC16_CONFIG0 (0x1444)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD							REG_EN	BANDGAP_EN	CAL_AVG_CFG			RSVD					CONV_PARAM														

ADC16

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
N/A								RW	RW	RW			N/A					RW													
x	x	x	x	x	x	x	x	0	0	0	0	0	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0

ADC16_CONFIG0 [31:0]

Bits	Name	Description
24	REG_EN	set to enable regulator
23	BANDGAP_EN	set to enable bandgap. user should set reg_en and bandgap_en before use adc16.
22-20	CAL_AVG_CFG	for average the calibration result. 0- 1 loop; 1- 2 loops; 2- 4 loops; 3- 8 loops; 4- 16 loops; 5-32 loops; others reserved
13-0	CONV_PARAM	conversion parameter

ADC16_CONFIG0 Bits

60.4.27 ADC16_CONFIG1 (0x1460)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																			COV_END_CNT				RSVD								
N/A																			RW				N/A								
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	1	x	x	x	x	x	x	x	

ADC16_CONFIG1 [31:0]

Bits	Name	Description
12-8	COV_END_CNT	used for faster conversion, user can change it to get higher convert speed(but less accuracy). should set to (21-convert_clock_number+1).

ADC16_CONFIG1 Bits

61 Analog Comparison Controller ACMP

This chapter introduces the functions and features of the analog comparison controller ACMP.

61.1 Feature Summary

The main features of the analog comparator ACMP are as follows:

- 3.0 to 3.6V operation
- supports rail-to-rail input
- Programmable hysteresis
- supports window mode output
- supports digital filtering of output
- supports generating interrupts
- supports generating DMA requests
- built-in 12bit DAC

61.2 Function Description

This section describes the functionality of the analog comparator ACMP.

ACMP module block diagram is shown in [Figure 98](#) :

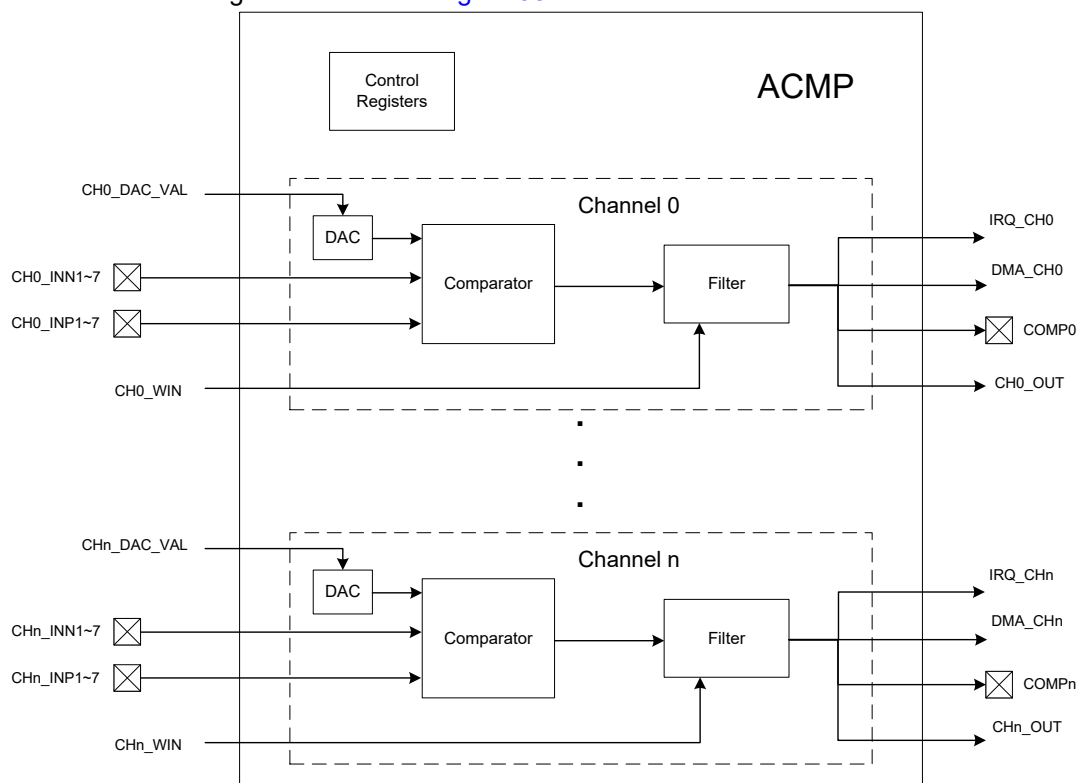


Figure 98: ACMP block diagram

In this product, each ACMP has 2 Channels and can manage 2 analog comparators.

Each Channel has 7 INP and 7 INN analog input signals from IO. You can select any INP and any INN signal for comparison and output the result.

Each Channel has a built-in DAC to provide a reference voltage, which can be compared with any signal of INP or INN and output the result.

comparison results can be filtered and support multiple filtering methods.

The comparison result can trigger an interrupt or DMA request, and there are also independent signal outputs to the IO or interconnect manager (see [Section 59.2](#) for specific connections).

61.2.1 Analog comparator basic configuration

Select the input source of the INP signal by configuring the CFG[PINSEL] bit of the register:

When PINSEL is 0, INP selects the DAC output. When PINSEL is 1,

Select the input source of the INN signal by configuring the CFG[MINSEL] bit of the register:

When MINSEL is 0, INN selects the DAC output. When MINSEL is 1,

By default, when the INP voltage is higher than the INN voltage, the comparison result is 1, otherwise when the INP voltage is lower than the INN voltage, the comparison result is 0. The analog comparator is enabled when the CFG[CMPEEN] bit of the

register is set to 1.

61.2.2 Processing of comparison results

The post-processing process of analog comparator comparison results is as shown in [Figure 99](#) :

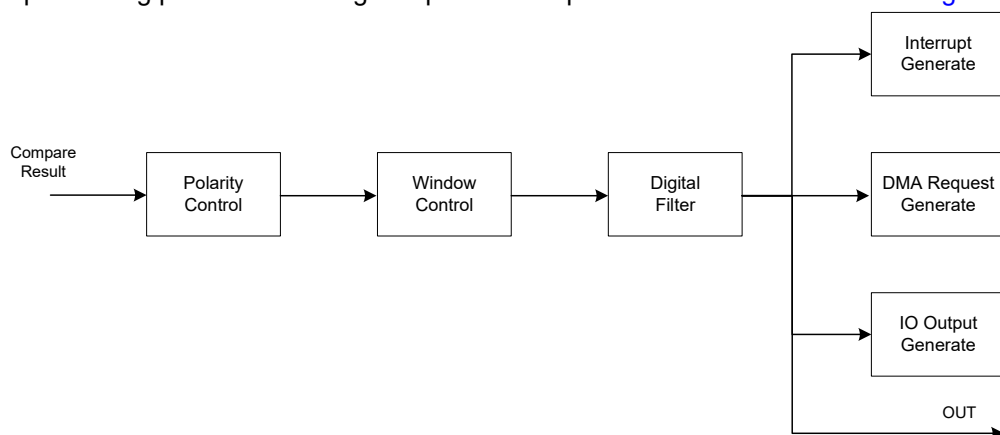


Figure 99: ACMP comparison result processing

61.2.2.1 Comparison result polarity control

If the CFG[OPOL] bit of the register is set to 1, the comparison result will be inverted.

61.2.2.2 window comparison mode

When the register CFG[WINEN] bit is set to 1, the window comparison mode is entered. In window mode, the external signal WIN is used to control the comparison result. When WIN is high level, the comparison result is valid. When WIN is low level, the comparison result is 0. Please see [Section 59.2](#) for the specific source of the WIN signal, which usually comes from the motor system to achieve synchronous linkage with other modules.

61.2.2.3 Digital filtering

The comparison result can be processed by digital filtering. When the register CFG[FLTBYP] is set to 0, the digital filter is valid. When it is set to 1, the digital filter is invalid.

When the digital filter is valid, the configuration register CFG[FLTMODE] can specify the working mode of the filter:

- b000, bypass mode, digital filter off
- b100, filtering mode, after the filter input is flipped, the output will also flip immediately, and then the filter input will be ignored for a certain period of time. In this mode, the filter output will follow the input while avoiding glitches in the output signal.
- b101, delay filter, after the filter input flips, it needs to be maintained for a certain period of time before the filter output flips
- b110, after the filter input is set to logic 0, it needs to be maintained for a certain period of time before the filter output will be set to logic 0. If the filter input is set to 1, the filter output will be set to 1 immediately.
- b111, after the filter input is set to logic 1, it needs to be maintained for a certain period of time before the filter output will be set to logic 1. If the filter input is set to 0, the filter output will be set to 0 immediately.

ACMP operating clock frequency see [Section 59.2](#).

61.2.2.4 Interrupt, DMA and comparison result output

The comparison result can trigger an interrupt. The rising edge and falling edge of the comparison result can be enabled to generate interrupts by configuring the IRQEN[REDGEN] and IRQEN[FEDGEN] bits respectively. The

comparison result can trigger a DMA request. The rising edge and falling edge of the comparison result can be enabled to generate DMA requests through the configuration register DMAEN[REDGEN] and DMAEN[FEDGEN] bits respectively. The

comparison result signal can be output directly from the module. For specific uses, see [Section 59.2](#).

Some products support outputting comparison results from IO. Please check whether [Chapter 19](#) has an ACMP IO output signal. The output enable is controlled by the register CFG[CMPOEN], and the IO output signal can bypass the digital filter by configuring the register CFG[FLTBYP]. When the bypass configuration is enabled, it does not affect the digital filter control of interrupts, DMA, and direct output of comparison results.

61.2.3 DAC configuration

The built-in DAC of the comparator provides a comparison reference voltage, and the register CFG[DACEN] bit enables the DAC.

DAC accuracy is 12 bits,

Configure the DAC value through the register DACCFG[DACCFG] bit field.

Some products support DAC voltage value setting from other module inputs instead of from the DACCFG register, such as the DAC transponder from the interconnection manager, to achieve linkage of the ACMP powered motor system. Please check from [Section 59.2](#) whether this product supports this attribute.

When the register CFG[DAC_TRIG_EN] is set to 1, the DAC voltage value setting selects the external

module input.

61.2.4 Comparator operating mode

Analog Comparator supports two operating modes: high performance mode and normal mode. The user sets the CFG[HPMODE] bit to 1 to set the comparator to high performance mode. In high-performance mode, the output delay of the comparator is smaller than in normal mode. For detailed data, please refer to the electrical characteristics chapter of the analog comparator in the data sheet document. High performance mode increases power consumption.

The analog comparator supports 4 different levels of hysteresis value configuration. The comparison result is changed only when the difference between the two compared voltage values exceeds the hysteresis value. The default value of register CFG[HYST] is 0, which represents a hysteresis value configuration of about 30mV. A value of 1 corresponds to about 20mV, and a value of 2 corresponds to about 10mV. Setting it to 3 means disabling hysteresis. The hysteresis value is slightly different in normal mode and high performance mode. For detailed data, please refer to the Analog Comparator Electrical Characteristics chapter of the data sheet document.

61.3 ACMP register list

ACMP register list is as follows:

ACMP0 base address: 0xF0130000

Offset	Register Name	Description	Reset Value
0x0000	CHANNEL[CHN0][CFG]	Configure Register	0x00000020
0x0004	CHANNEL[CHN0][DACCFG]	DAC configure register	0x00008080
0x0010	CHANNEL[CHN0][SR]	Status register	0x00000000
0x0014	CHANNEL[CHN0][IRQEN]	Interrupt request enable register	0x00000000
0x0018	CHANNEL[CHN0][DMAEN]	DMA request enable register	0x00000000
0x0020	CHANNEL[CHN1][CFG]	Configure Register	0x00000020
0x0024	CHANNEL[CHN1][DACCFG]	DAC configure register	0x00008080
0x0030	CHANNEL[CHN1][SR]	Status register	0x00000000
0x0034	CHANNEL[CHN1][IRQEN]	Interrupt request enable register	0x00000000
0x0038	CHANNEL[CHN1][DMAEN]	DMA request enable register	0x00000000

Table 234: ACMP Register List

61.4 ACMP register description

ACMP register details are as follows:

61.4.1 CHANNEL[CFG] (0x0 + 0x20 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
HYST		DACEN	HPMODE	CMPEN	MINSEL			DAC_TRIG_EN	PINSEL			CMPOEN	FLTBYPS	WINEN	OPOL	FLTMODE			RSVD	FLTLEN												
RW		RW	RW	RW	RW			RW	RW			RW	RW	RW	RW	RW			N/A	RW												
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	0	0	0	0	0	0	0	1	0	0	0	0	0

CHANNEL[CFG] [31:0]

Bits	Name	Description
31-30	HYST	This bitfield configure the comparator hysteresis. 0: Hysteresis about 30mV 1: Hysteresis about 20mV 2: Hysteresis about 10mV 3: Disable hysteresis
29	DACEN	This bit enable the comparator internal DAC 0: DAC disabled 1: DAC enabled
28	HPMODE	This bit enable the comparator high performance mode. 0: HP mode disabled 1: HP mode enabled
27	CMPEN	This bit enable the comparator. 0: ACMP disabled 1: ACMP enabled
26-24	MINSEL	PIN select, from pad_ai_acmp[7:1] and dac_out
23	DAC_TRIG_EN	if set, the dac value is from moto system when valid if clr, use dac_cfg value
22-20	PINSEL	MIN select, from pad_ai_acmp[7:1] and dac_out
19	CMPOEN	This bit enable the comparator output on pad. 0: ACMP output disabled 1: ACMP output enabled
18	FLTBYPS	This bit bypass the comparator output digital filter. 0: The ACMP output need pass digital filter 1: The ACMP output digital filter is bypassed.
17	WINEN	This bit enable the comparator window mode. 0: Window mode is disabled 1: Window mode is enabled
16	OPOL	The output polarity control bit. 0: The ACMP output remain un-changed. 1: The ACMP output is inverted.

Bits	Name	Description
15-13	FLTMODE	This bitfield define the ACMP output digital filter mode: 000-bypass 100-change immediately; 101-change after filter; 110-stalbe low; 111-stable high
11-0	FLTLEN	This bitfield define the ACMP output digital filter length. The unit is ACMP clock cycle.

CHANNEL[CFG] Bits

61.4.2 CHANNEL[DACCFG] (0x4 + 0x20 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																DACCFG												RSVD			
N/A																RW												N/A			
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	0	0	0	0	0	0	0	1	0	0	0	x	x	x	x

CHANNEL[DACCFG] [31:0]

Bits	Name	Description
15-4	DACCFG	12bit DAC digital value.

CHANNEL[DACCFG] Bits

61.4.3 CHANNEL[SR] (0x10 + 0x20 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																														FEDGF	REDGF
N/A																														RW	RW
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0

CHANNEL[SR] [31:0]

Bits	Name	Description
1	FEDGF	Output falling edge flag. Write 1 to clear this flag.
0	REDGF	Output rising edge flag. Write 1 to clear this flag.

CHANNEL[SR] Bits

61.4.4 CHANNEL[IRQEN] (0x14 + 0x20 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																												FEDGEN	REDGEN		
N/A																												RW	RW		
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	

CHANNEL[IRQEN] [31:0]

Bits	Name	Description
1	FEDGEN	Output falling edge flag interrupt enable bit.
0	REDGEN	Output rising edge flag interrupt enable bit.

CHANNEL[IRQEN] Bits

61.4.5 CHANNEL[DMAEN] (0x18 + 0x20 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																											FEDGEN	REDGEN			
N/A																											RW	RW			
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	

CHANNEL[DMAEN] [31:0]

Bits	Name	Description
1	FEDGEN	Output falling edge flag DMA request enable bit.
0	REDGEN	Output rising edge flag DMA request enable bit.

CHANNEL[DMAEN] Bits

62 Temperature Sensor TSNS

This chapter describes the functions and characteristics of the temperature sensor TSNS.

62.1 Feature Summary

This product includes a temperature sensor that measures the temperature of the die. The main characteristics are as follows:

- Single measurement
- Continuous measurement
- Average multiple measurements
- Record the upper and lower limits of temperature
- Temperature Range Comparison
- An interrupt occurs when the temperature is out of range
- Reset chip when temperature exceeds range

62.2 Function Description

TSNS converts the built-in temperature-sensitive diode through the built-in ADC to get the die temperature. The software needs to enable its bias circuit to perform temperature collection.

During the temperature acquisition process, ADC performs multiple measurements and samples on the internal temperature sensor and averages it to obtain higher accuracy. The average number of times is set by the AVERAGE field, and a maximum of 128 times is supported.

AGE register, the temperature sensor module provides the time since the last temperature collection, and the time provided in this register is counted in 24M clock cycles.

The temperature sensor internally records the highest and lowest temperatures that have occurred. It can be read through TMAX and TMIN, cleared through RECORD_MAX_CLR and RECORD_MIN_CLR, and re-recorded.

62.2.1 Temperature acquisition mode

conversion can be triggered by software or can be collected continuously at regular intervals.

In single acquisition mode, the software measures the temperature by writing TRIGGER. After the measurement is completed, VALID is set to 1, indicating that the temperature value is valid.

continuous mode is set through the CONTINUOUS field. In continuous mode, the user sets the time interval for temperature sampling by setting the VALIDITY field, and the time interval is counted at a 24M clock.

62.2.2 Temperature comparison function

temperature sensor has an interrupt and reset function that allows high and low temperatures to be compared by setting COMPARE_MIN_EN and COMPARE_MAX_EN. The temperature range for

interrupts and resets is set independently. The temperature for the

interrupt comparison is set in the UPPER_LIM_IRQ and LOWER_LIM_IRQ registers. The temperature for the

reset comparison is set in the UPPER_LIM_RST and LOWER_LIM_RST registers.

When the measured die temperature exceeds the set range, the temperature sensor module generates a flag bit. An interrupt is generated if the interrupt enable bit IRQ_EN is set to 1. If the reset enable bit RST_EN is set to 1, a reset is generated.

62.3 TSNS register

TSNS is as follows:

TSNS base address: 0xF0090000

Offset	Register Name	Description	Reset Value
0x0000	T	Temperature	0x00000000
0x0004	TMAX	Maximum Temperature	0xFF800000
0x0008	TMIN	Minimum Temperature	0x007FFFFFFF
0x000C	AGE	Sample age	0x00000000
0x0010	STATUS	Status	0x00000000
0x0014	CONFIG	Configuration	0x00600300
0x0018	VALIDITY	Sample validity	0x016E3600
0x001C	FLAG	Temperature flag	0x00000000
0x0020	UPPER_LIM_IRQ	Maximum temperature to interrupt	0x00000000
0x0024	LOWER_LIM_IRQ	Minimum temperature to interrupt	0x00000000
0x0028	UPPER_LIM_RST	Maximum temperature to reset	0x00000000
0x002C	LOWER_LIM_RST	Minimum temperature to reset	0x00000000
0x0030	ASYNC	Configuration in asynchronous mode	0x00000000
0x0038	ADVAN	Advance configuration	0x00000000

Table 235: TSNS Register List

62.4 TSNS register details

TSNS is as follows:

62.4.1 T (0x0)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T																															
R																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

T [31:0]

Bits	Name	Description
31-0	T	Signed number of temperature in 256 x celsius degree

T Bits

62.4.2 TMAX (0x4)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T																															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TMAX [31:0]

Bits	Name	Description
31-0	T	maximum temperature ever found

TMAX Bits

62.4.3 TMIN (0x8)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T																															
RO																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TMIN [31:0]

Bits	Name	Description
31-0	T	minimum temperature ever found

TMIN Bits

62.4.4 AGE (0xC)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AGE																															
RO																															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

AGE [31:0]

Bits	Name	Description
31-0	AGE	age of T register in 24MHz clock cycles

AGE Bits

62.4.5 STATUS (0x10)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALID	RSVD																													TRIGGER	
RO	N/A																													W1C	
0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	

STATUS [31:0]

Bits	Name	Description
31	VALID	indicate value in T is valid or not 0: not valid 1:valid
0	TRIGGER	Software trigger for sensing in trigger mode, trigger will be ignored if in sensing or other mode

STATUS Bits

62.4.6 CONFIG (0x14)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRQ_EN	RST_EN	RSVD				COMPARE_MIN_EN	COMPARE_MAX_EN	SPEED								RSVD					AVERAGE			RSVD		CONTINUOUS	RSVD		ASYNC	ENABLE	
RW	RW	N/A				RW	RW	RW								N/A					RW			N/A		RW	N/A		RW	RW	
0	0	x	x	x	x	0	0	0	0	0	0	0	0	0	0	x	x	x	x	x	0	0	0	x	x	x	0	x	x	0	0

CONFIG [31:0]

Bits	Name	Description
31	IRQ_EN	Enable interrupt
30	RST_EN	Enable reset

Bits	Name	Description
25	COMPARE_MIN_EN	Enable compare for minimum temperature
24	COMPARE_MAX_EN	Enable compare for maximum temperature
23-16	SPEED	cycles of a progressive step in 24M clock, valid from 24-255, default 96 24: 24 cycle for a step 25: 25 cycle for a step 26: 26 cycle for a step ... 255: 255 cycle for a step
10-8	AVERAGE	Average time, default in 3 0: measure and return 1: twice and average 2: 4 times and average ... 7: 128 times and average
4	CONTINUOUS	continuous mode that keep sampling temperature peridically 0: trigger mode 1: continuous mode
1	ASYNC	Acynchronous mode, this mode can work without clock, only available function ios compare to certain ADC value 0: active mode 1: Async mode
0	ENABLE	Enable temperature 0: disable, temperature sensor is shut down 1: enable. Temperature sensor enabled

CONFIG Bits

62.4.7 VALIDITY (0x18)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALIDITY																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VALIDITY [31:0]

Bits	Name	Description
31-0	VALIDITY	time for temperature values to expire in 24M clock cycles

VALIDITY Bits

62.4.8 FLAG (0x1C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD										RECORD_MIN_CLR	RECORD_MAX_CLR	RSVD		UNDER_TEMP	OVER_TEMP	RSVD														IRQ	
N/A										RW	RW	N/A		RW	RW	N/A														RW	
x	x	x	x	x	x	x	x	x	x	0	0	x	x	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0

FLAG [31:0]

Bits	Name	Description
21	RECORD_MIN_CLR	Clear minimum recorder of temerature, write 1 to clear
20	RECORD_MAX_CLR	Clear maximum recorder of temerature, write 1 to clear
17	UNDER_TEMP	Clear under temperature status, write 1 to clear
16	OVER_TEMP	Clear over temperature status, write 1 to clear
0	IRQ	IRQ flag, write 1 to clear

FLAG Bits

62.4.9 UPPER_LIM_IRQ (0x20)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																T															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

UPPER_LIM_IRQ [31:0]

Bits	Name	Description
31-0	T	Maximum temperature for compare

UPPER_LIM_IRQ Bits

62.4.10 LOWER_LIM_IRQ (0x24)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LOWER_LIM_IRQ [31:0]

Bits	Name	Description
31-0	T	Minimum temperature for compare

LOWER_LIM_IRQ Bits

62.4.11 UPPER_LIM_RST (0x28)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

UPPER_LIM_RST [31:0]

Bits	Name	Description
31-0	T	Maximum temperature for compare

UPPER_LIM_RST Bits

62.4.12 LOWER_LIM_RST (0x2C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LOWER_LIM_RST [31:0]

Bits	Name	Description
31-0	T	Minimum temperature for compare

LOWER_LIM_RST Bits

62.4.13 ASYNC (0x30)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD							ASYNC_TYPE	RSVD							POLARITY	RSVD					VALUE										
N/A							RW	N/A							RW	N/A					RW										
x	x	x	x	x	x	x	0	x	x	x	x	x	x	x	0	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0

ASYNC [31:0]

Bits	Name	Description
24	ASYNC_TYPE	Compare hotter than or colder than in asynchrononous mode 0: hotter than 1: colder than
16	POLARITY	Polarity of internal comparator
10-0	VALUE	Value of async mode to compare

ASYNC Bits

62.4.14 ADVAN (0x38)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD						ASYNC_IRQ	ACTIVE_IRQ	RSVD							SAMPLING	RSVD										NEG_ONLY	POS_ONLY				
N/A						RO	RO	N/A							RO	N/A										RW	RW				
x	x	x	x	x	x	0	0	x	x	x	x	x	x	x	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0

ADVAN [31:0]

Bits	Name	Description
25	ASYNC_IRQ	interrupt status of asynchronous mode
24	ACTIVE_IRQ	interrupt status of active mode
16	SAMPLING	temperature sampling is working
1	NEG_ONLY	use negative compare polarity only
0	POS_ONLY	use positive compare polarity only

ADVAN Bits

63 Information Security Module Overview

This chapter describes the information security module of this product.

The information security modules of this product are mainly divided into the following categories:

- Encryption and decryption engine
 - Online decryption module EXIP
- encryption and decryption engine
 - Keyzone in OTP
 - Based on product life cycle management
 - Monitoring JTAG debugging interface
 - Monitor that the processor enters Debug mode
- BOOT ROM safe boot mechanism ensures safe and trustworthy loading of user program images

63.1 Online decryption engine EXIP

This product includes an online decryption engine EXIP0, which is closely coupled with XPI0. Users can use AES-128 CTR mode to encrypt external NOR Flash stored data according to specifications. When this product is connected to encrypted external NOR Flash through XPI, EXIP decrypts online in real time with a zero-waiting period, restoring the encrypted code to plaintext for direct online execution by the processor.

EXIP supports restoring DEK(the key used to encrypt data by Data Encryption Key) from the Key Blob using EXIP KEK(the key used to encrypt data) stored in OTP. Protect DEK.

63.2 One-time programmable storage OTP

The OTP of this product supports storing data related to various security functions:

- 128-bit debug key. This product supports JTAG port locking. After locking, the debugger needs to provide a debugging password to unlock the JTAG interface for debugging.
- 128-bit UUID, unique identification of the chip
- The public key HASH for secure boot is the 256-bit SHA-256 hash value of the public key
- Used for EXIP to decrypt the 256-bit KEK of the Key Blob containing DEK, EXIP0 KEK

For specific information about this security-related data, please refer to the relevant chapters of OTP and other security modules.

63.3 BOOT ROM

This product integrates BOOT ROM, and the BOOT ROM code is always executed first after the system is reset. The physical characteristics of read-only memory ensure that BOOT ROM code is almost impossible to tamper with. This resulted in the original trusted secure execution environment.

BOOT ROM supports the following security features:

- Executes code directly from encrypted serial NOR Flash via EXIP and XPI.
- is safely launched. You can verify that the source of the image is legal and trustworthy before jumping to the user's software image. Support signature verification for ECDSA or SM2
- supports loading encrypted software images and AES or SM4 encryption algorithms

OVERVIEW

- Flash loader supports burning encrypted firmware

For details of the security features supported by BOOT ROM, please refer to the relevant chapters of the ROM.

64 Execution In Place Encryption Engine (EXIP)

This chapter describes the main features and functions of the Execution In Place (EXIP) encryption engine.

64.1 Feature Summary

The EXIP engine enables real-time decryption of external encrypted NOR Flash. The main features of EXIP are:

- Supports AES-128 CTR mode decryption.
- Supports 4 segments; each segment can be encrypted with a different key.
- Supports encapsulating sensitive information like Data Encryption Keys (DEK) using Key Blobs (compliant with RFC3394).
- Supports hardware unwrapping of Key Blobs (compliant with RFC3394).
- Supports reading the KEK (Key-Encryption Key) used to unwrap Key Blobs from OTP.

64.2 Functional Description

On this product, users can connect various brands and capacities of serial NOR Flash via the XPI controller. The XPI controller supports mapping external memory directly onto the system's address space. Consequently, master devices like the processor and DMA can access external memory directly, loading data or executing code, just as they would access on-chip memory. This access method, which does not require copying data from external memory to RAM before execution, is called Execution In Place (XIP). The EXIP engine is tightly coupled with the XPI controller, supporting real-time data decryption while performing XIP from external serial NOR Flash. Therefore, users can protect data and code by encrypting the data and code on the external NOR Flash using the AES-128 CTR algorithm.

64.2.1 EXIP Segments, Keys, and Counters

EXIP supports 4 segments, each of which can have an independent key and counter.

Each segment can define a start address and an end address. The area between the start and end addresses is the encrypted region. EXIP is tightly coupled with the XPI controller. When a bus master (e.g., processor or DMA) initiates an access to external memory via XPI, EXIP monitors the system address of the access. If the address falls within an encrypted region, the decryption operation is performed.

RGNx_DSCR_W0[START] holds the start address of segment x, and RGNx_DSCR_W1[END] holds the end address of segment x. EXIP requires the encrypted region length to be in units of 1KB, and the start address must be 1KB-aligned. Assuming the encrypted region length is x KB, the end address should be $START + x \times 1024 - 1$. Therefore, only the upper 22 bits of START are taken, and the lower 10 bits should be 0. Similarly, only the upper 22 bits of END are taken, and the lower 10 bits should be 1.

EXIP supports AES-128 CTR mode decryption with a 128-bit key. The key is stored in registers RGNx_KEY0, RGNx_KEY1, RGNx_KEY2, and RGNx_KEY3.

The counter length is the same as the AES block length, which is 128 bits. AES CTR mode requires that the counter value for each data block encrypted with the same key must not repeat. For each 128-bit (16-byte) data block, its counter is composed of the 32-bit system address of the data block and a 64-bit NONCE. The NONCE is stored in RGNx_CTR0 and RGNx_CTR1. The counter composition is as follows:

- Counter[127:96] = CTR0[31:0]
- Counter[95:64] = CTR1[31:0]
- Counter[63:32] = CTR0[31:0] XOR CTR1[31:0]
- Counter[31:0] = Sysaddr[31:0] AND 0xFFFFFFFF0

Where:

- CTR0 is the lower 32 bits of the 64-bit NONCE.
- CTR1 is the upper 32 bits of the 64-bit NONCE.
- SysAddr is the system-mapped address of the ciphertext block. The ciphertext block length is 128 bits / 16 bytes. One ciphertext block corresponds to one counter value, so only the upper 28 bits of the address are taken for the counter, and the lower 4 address bits are set to 0.

64.2.2 EXIP Key Wrapping and Unwrapping

EXIP uses the key wrapping and unwrapping algorithms defined in RFC3394 to protect the decryption keys and other sensitive data used by EXIP. For clarity, the key used by EXIP for wrapping/unwrapping is called the KEK (Key-Encryption Key). The key used by EXIP to decrypt data/code on external memory is called the DEK (Data Encryption Key).

For each EXIP segment, the DEK and related data to be wrapped consist of 5 data blocks of 64 bits each, totaling 40 bytes. A summary is provided below:

Address Offset	Description
0x00	DEKx[31:0], i.e., the part of the key stored in RGNx_KEY0.
0x04	DEKx[63:32], i.e., the part of the key stored in RGNx_KEY1.
0x08	DEKx[95:64], i.e., the part of the key stored in RGNx_KEY2.
0x0C	DEKx[127:96], i.e., the part of the key stored in RGNx_KEY3.
0x10	NONCE[31:0], i.e., the lower part of the NONCE stored in RGNx_CTR0.
0x14	NONCE[63:32], i.e., the upper part of the NONCE stored in RGNx_CTR1.
0x18	START, i.e., RGNx_RGN_SA[31:10], the start address of the encrypted region; other bits are 0.
0x1C	END, i.e., RGNx_RGN_EA[31:10], the end address of the encrypted region. RO, i.e., RGNx_RGN_EA[2], when set to 1, enables read protection for RGNx configuration registers. DECEN, i.e., RGNx_RGN_EA[1], when set to 1, enables EXIP decryption for this segment. VALID, i.e., RGNx_RGN_EA[0], when set to 1, indicates segment x is valid. Other bits are set to 1.
0x20	Padding data, all set to 0.

Address Offset	Description
0x24	Padding data, all set to 0.

Table 236: EXIP Key Wrap Data Summary

After the key wrapping (Key Wrap) process, this data becomes 6 ciphertext blocks of 64 bits each, totaling 48 bytes.

When the BE and KBPE bits in the CFG register are set to 1, EXIP initiates the hardware key unwrapping process. EXIP fetches the ciphertext blocks directly from the start address of the external memory via the XPI controller. The location of the ciphertext blocks in external memory is as follows:

Address Offset	Description
0x00	Ciphertext package for segment 0 (RGN0), containing DEK, NONCE, region address information.
0x40	Ciphertext package for segment 1 (RGN1), containing DEK, NONCE, region address information.
0x80	Ciphertext package for segment 2 (RGN2), containing DEK, NONCE, region address information.
0xC0	Ciphertext package for segment 3 (RGN3), containing DEK, NONCE, region address information.

Table 237: EXIP Key Wrap Data Location in External Flash

Once EXIP starts unwrapping the ciphertext packages, it automatically performs the unwrapping (Key Unwrap) for the ciphertext packages of all 4 segments. After unwrapping, it loads the recovered DEK, NONCE, START, END, and other information into the corresponding registers of RGNx. If the recovered initial value does not match the defined IV (0xA6A6A6A6A6A6A6A6), an error is indicated. RGNx_DESC_W1[0] VALID bit is set to 1, and the STA[KBERR] flag is also set to 1.

If the user actually uses fewer than 4 segments, they still need to generate ciphertext packages for all 4 segments following the key wrapping procedure, burn them to the corresponding addresses in Flash, and fill the start and end addresses of the unused segments with 0.

The KEK used by EXIP for key unwrapping is loaded directly from the on-chip OTP via a dedicated data path. Once the KEK is burned into the designated location in OTP and configured with appropriate read/write protection, it is no longer visible to software. For information on OTP programming and read/write protection configuration, users can refer to the OTP chapter.

On this product, enabling EXIP and key unwrapping is managed by the BOOT ROM. After the ENCRYPT_XIP field in OTP is set to 1, the ROM forces EXIP to be enabled during XPI NOR Flash boot.

In summary, the steps for a user to utilize EXIP for encrypted execution in place are:

- Generate a KEK and use it to perform key wrapping (KeyWrap) on the DEK, NONCE, START, END, and other data, generating ciphertext packages. Note that ciphertext packages for all 4 EXIP segments must be generated.

- Burn the ciphertext packages to the specified addresses in the external serial NOR Flash.
- Encrypt the image according to the DEK and NONCE, and burn the encrypted image to the external serial NOR Flash.
- Burn the KEK to the designated location in OTP and configure read/write protection.
- Set the ENCRYPT_XIP field in OTP to 1.
- After a reset, the chip will support booting from the encrypted external serial NOR Flash.

64.3 Appendix

64.3.1 RFC3394 Overview

EXIP supports AES-128 CTR mode decryption of encrypted data and code on external NOR Flash. Clearly, the security of this data and code depends on the security of the decryption keys used by EXIP. EXIP supports a key wrapping algorithm called Key Wrap to protect the decryption keys and other sensitive data used by EXIP.

The Key Wrap algorithm used by EXIP complies with RFC3394, "Advanced Encryption Standard (AES) Key Wrap Algorithm". RFC3394 defines an algorithm called Key Wrap, which is based on the AES standard. It can encrypt and protect data of any length and provides data integrity verification. RFC3394 refers to the key used to wrap the protected data as the KEK (Key-Encryption Key). The KEK can be a key supported by the AES standard: AES-128, AES-192, or AES-256. EXIP uses a 128-bit KEK.

RFC3394 defines that the protected data can be divided into several data blocks, each 64 bits in size. The minimum number of blocks is 2, and there is no upper limit. After the key wrapping process, n plaintext data blocks result in $n+1$ ciphertext data blocks. Conversely, $n+1$ ciphertext data blocks, after the key unwrapping process, can be restored to n plaintext data blocks.

Quoting the Key Wrap steps from RFC3394:

Input: Plaintext, n 64-bit data blocks $\{P_1, P_2, \dots, P_n\}$ and a key K (the KEK).

Output: Ciphertext, $(n+1)$ 64-bit ciphertext blocks $\{C_0, C_1, \dots, C_n\}$.

Step 1, Initialize

Set A to the initial value IV .

For $i = 1$ to n

$$R[i] = P[i]$$

Step 2, Calculate intermediate values

For $j = 0$ to 5

For $i = 1$ to n

$$B = \text{AES}(K, A \parallel R[i])$$

$$A = \text{MSB}(64, B) \oplus t \text{ where } t = (n*j)+i$$

$$R[i] = \text{LSB}(64, B)$$

Step 3, Output result

$$C[0] = A$$

For $i = 1$ to n

$$C[i] = R[i]$$



Quoting the Key Unwrap steps from RFC3394:

Input: Ciphertext, $(n+1)$ 64-bit ciphertext blocks $\{C_0, C_1, \dots, C_n\}$, and a key K (the KEK).

Output: Plaintext, n 64-bit data blocks $\{P_1, P_2, \dots, P_n\}$.

Step 1, Initialize

$A = C[0]$

For $i = 1$ to n

$R[i] = C[i]$

Step 2, Calculate intermediate values

For $j = 5$ to 0

For $i = n$ to 1

$B = \text{AES-1}(K, (A \wedge t) \mid R[i])$ where $t = n*j+i$

$A = \text{MSB}(64, B)$

$R[i] = \text{LSB}(64, B)$

Step 3, Output result

If A matches the initial value IV , then

For $i = 1$ to n

$P[i] = R[i]$

Otherwise, an error is indicated.

Explanation of expressions used in the above key wrapping and unwrapping steps:

$\text{AES}(K, W)$ Encrypt data W with key K using AES algorithm.

$\text{AES-1}(K, W)$ Decrypt data W with key K using AES algorithm.

$\text{MSB}(j, W)$ Returns the j Most Significant Bits (MSBs) of data W .

$\text{LSB}(j, W)$ Returns the j Least Significant Bits (LSBs) of data W .

$B_1 \wedge B_2$ Bitwise XOR of data B_1 and B_2 .

$B_1 \mid B_2$ Concatenation of data B_1 and B_2 .

K The KEK, Key-Encryption Key.

n Number of 64-bit data blocks.

s Number of key wrapping steps, $s = 6n$.

$P[i]$ The i -th 64-bit plaintext data block.

$C[i]$ The i -th 64-bit ciphertext data block.

A 64-bit register holding integrity check data.

$R[i]$ An array of 64-bit registers, where $i = 1, \dots, n$.

IV 64-bit initial value for wrapping; used for comparison with A during unwrapping.

EXIP uses the default initial value defined in RFC3394: 0xA6A6A6A6A6A6A6A6.

65 System Debugging Overview

This chapter describes the debugging module of this product.

The Debug debugging system complies with The RISC-V Debug Specification, Version 0.13. The Debug debugging system consists of two parts: JTAG Interface Conversion Module (DTM) and Debugging Module (DM). DTM interfaces with an external debugger through a standard JTAG interface, which can convert debugging instructions received on JTAG into read-write access to the DM module. The debugging module DM integrates debugging functions, which can suspend or resume CPU operation, generate reset, and access on-chip resources.

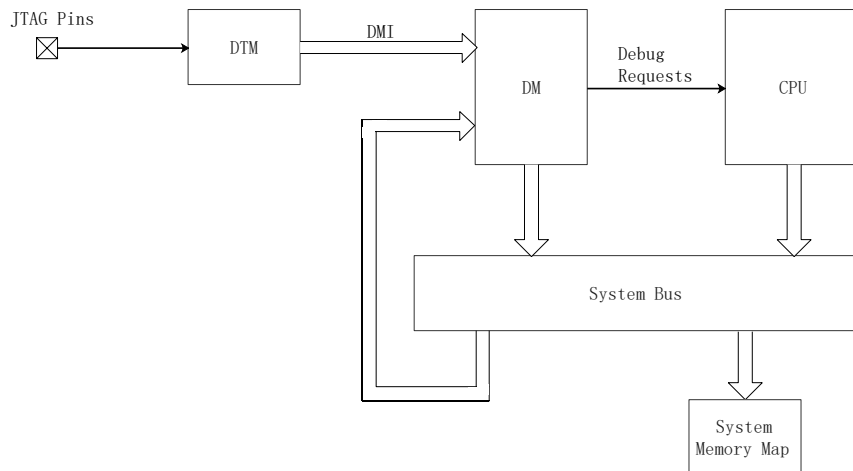


Figure 100: debugging system diagram

Debug system features are as follows:

- supports a 4-wire JTAG interface and complies with IEEE Std 1149.1
- allows off-chip debuggers to connect to the Debug debugging module through the JTAG interface conversion module
- allows the processor to access the Debug debugging module through system address mapping
- supports 8 hardware breakpoints
- supports single-step debugging
- supports System Bus Access. The debugging interface can be used as the bus master device and can directly access on-chip resources without disrupting kernel operation
- supports the Debug security mechanism, which can open, close or lock the Debug debugging interface. When the Debug is locked, the user needs to unlock it through the security code.

The JTAG ID of this product is 0x1000563d.

66 Debugging Transport Module DTM

Debug Transport Module (DTM)

This module conforms to the definition of RISC-V0.13 and supports the JTAG interface defined in IEEE1149.1. The instruction length is 5 bits.

66.1 DTM command

INSTRUMENTS list is as follows:

Value	Name	Description	Data Width
0x00	USERCODE	Device identification code	32
0x01	IDCODE	Device identification code	32
0x08	CLAMP	CLAMP instruction	404
0x09	EXTEST	EXTEST instruction	404
0x0A	SAMPLE/PRELOAD	SAMPLE/PRELOAD instruction	404
0x0B	HIGHZ	HIGHZ instruction	1
0x0F	AUTHEN	Debug authentication instruction	32
0x10	DTMCS	DTM Control and Status	32
0x11	DMI	DMI interface	41
0x1F	BYPASS	BYPASS	1

Table 238: DTM Instruction List

66.2 DTM command description

INSTRUCTION is as follows:

66.2.1 USERCODE (0x0)

Bits	Name	Access	Description
31-0	USERCODE	RO	

USERCODE Bits

66.2.2 IDCODE (0x1)

Bits	Name	Access	Description
31-0	IDCODE	RO	

IDCODE Bits

66.2.3 CLAMP (0x8)

Bits	Name	Access	Description
403-0	BOUNDRY	RO	Refer to IEEE 1149.1 for details.

CLAMP Bits

66.2.4 EXTEST (0x9)

Bits	Name	Access	Description
403-0	BOUNDRY	RO	Refer to IEEE 1149.1 for details.

EXTEST Bits

66.2.5 SAMPLE/PRELOAD (0xA)

Bits	Name	Access	Description
403-0	BOUNDRY	RO	Refer to IEEE 1149.1 for details.

SAMPLE/PRELOAD Bits

66.2.6 HIGHZ (0xB)

Bits	Name	Access	Description
0	BYPASS	RO	Refer to IEEE 1149.1 for details.

HIGHZ Bits

66.2.7 AUTHEN (0xF)

Bits	Name	Access	Description
31-0	AUTH	RW	Debug authentication, perform 8 DR shift to authenticate debugger, first 4 shift shift out 128 bit die trace information. following 4 shift accept 128 bit authentication password in OTP.

AUTHEN Bits

66.2.8 DTMCS (0x10)

Bits	Name	Access	Description
17	DMIHARDRESET	W1	Writing 1 to this bit does a hard reset of the DTM, causing the DTM to forget about any outstanding DMI transactions. In general, this should only be used when the debugger has reasons to expect that the outstanding DMI transaction will never complete (e.g., a reset condition causes an inflight DMI transaction to be canceled).
16	DMIRESET	W1	Writing 1 to this bit clears the sticky error state and allows the DTM to retry or complete the previous transaction.
14-12	IDLE	RO	This is a hint to the debugger of the minimum number of cycles a debugger should spend in RunTest/Idle after every DMI scan to avoid a busy return code (dmistat of 3).
11-10	DMISTAT	RO	State of DMI 0: No error 1: Reserved 2: An operation failed (resulted in op of 2) 3: An operation was attempted while a DMI access was still in progress (resulted in op of 3)
9-4	ABITS	RO	Bit width of DMI address is 7
3-0	VERSION	RO	Version of the implemented DTM. 0x1 indicates that the current implementation conforms to RISC-V External Debug Support (TD003) V0.13.

DTMCS Bits

66.2.9 DMI (0x11)

Bits	Name	Access	Description
40-34	ADDRESS	RW	Address used for DMI access. In Update-DR this value is used to access the DM over the DMI.
33-2	DATA	RW	The data to send to the DM over the DMI during Update-DR, and the data returned from the DM as a result of the previous operation.

Bits	Name	Access	Description
1-0	OP	RW	Write operation: 0: Ignore data and address. (nop) 1: Read from address. (read) 2: Write data to address. (write) 3: Reserved Read operation: 0: The previous operation completed successfully. 1: Reserved 2: A previous operation failed. 3: An operation was attempted while a DMI request is still in progress. The data scanned into dmi in this access will be ignored.

DMI Bits

66.2.10 BYPASS (0x1F)

Bits	Name	Access	Description
0	BYPASS	RO	Refer to IEEE 1149.1 for details.

BYPASS Bits

67 Debugging Module DM

This module conforms to the definition of RISC-V0.13 and has a 16x32-bit program buffer. There are two ways to access the registers in the debugging module. The CPU accesses them through the system bus or the debugger accesses them through DTM. The two access methods have different memory mappings.

67.1 System memory mapping

Offset	Name	Description
0x0000 - 0x007F	DEBUGROM	Debug ROM
0x0080 - 0x00BF	PROGBUF	Program Buffer 0-15
0x00C0 - 0x00CF	DATA	Abstract Data 0-3
0x00D0 - 0x01FF	RESERVED	Reserved for internal use

Table 239: DM system memory mapping

67.2 DMI memory mapping

Offset	Register Name	Description	Reset Value
0x0004	DATA[DATA0]	Abstract Data 0-3	0x00000000
0x0005	DATA[DATA1]	Abstract Data 0-3	0x00000000
0x0006	DATA[DATA2]	Abstract Data 0-3	0x00000000
0x0007	DATA[DATA3]	Abstract Data 0-3	0x00000000
0x0010	DMCONTROL	Debug Module Control	0x00000000
0x0011	DMSTATUS	Debug Module Status	0x004000A2
0x0012	HARTINFO	Hart Info	0x002140C0
0x0013	HALTSUM	Halt Summary	0x00000000
0x0014	HAWINDOWSEL	Hart Array Window Select	0x00000000
0x0015	HAWINDOW	Hart Array Window	0x00000000
0x0016	ABSTRACTCS	Abstract Control and Status	0x08000004
0x0017	COMMAND	Abstract Command	0x00000000
0x0018	ABSTRACTAUTO	Abstract Command Autoexec	0x00000000
0x0020	PROGBUF[PROGBUF0]	Program Buffer 0-15	0x00000000
0x0021	PROGBUF[PROGBUF1]	Program Buffer 0-15	0x00000000
0x0022	PROGBUF[PROGBUF2]	Program Buffer 0-15	0x00000000
0x0023	PROGBUF[PROGBUF3]	Program Buffer 0-15	0x00000000
0x0024	PROGBUF[PROGBUF4]	Program Buffer 0-15	0x00000000
0x0025	PROGBUF[PROGBUF5]	Program Buffer 0-15	0x00000000
0x0026	PROGBUF[PROGBUF6]	Program Buffer 0-15	0x00000000
0x0027	PROGBUF[PROGBUF7]	Program Buffer 0-15	0x00000000
0x0038	SBCS	System Bus Access Control and Status	0x20040407
0x0039	SBADDRESS[SBADDRESS0]	System Bus Address	0x00000000

Offset	Register Name	Description	Reset Value
0x003A	SBADDRESS[SBADDRESS1]	System Bus Address	0x00000000
0x003B	SBADDRESS[SBADDRESS2]	System Bus Address	0x00000000
0x003C	SBDATA[SBDATA0]	System Bus Data	0x00000000
0x003D	SBDATA[SBDATA1]	System Bus Data	0x00000000
0x003E	SBDATA[SBDATA2]	System Bus Data	0x00000000
0x003F	SBDATA[SBDATA3]	System Bus Data	0x00000000

Table 240: DM Register List

67.3 DM register description

67.3.1 DATA (0x4 + 0x1 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

DATA [31:0]

Bits	Name	Description
31-0	DATA	The registers are accessible from both the DMI interface and the system bus interface to support data exchanges between the external debugger and the processor

DATA Bits

67.3.2 DMCONTROL (0x10)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HALTREQ	RESUMEREQ	RSVD	ACKHAVERESET	RSVD	HASEL	HARTSEL										RSVD										SETRESETHALTREQ	CLRRESETHALTREQ	NDMRESET	DMACTIVE		
WO	WO	N/A	RW	N/A	RW	RW										N/A										W1S	W1C	RW	RW		
0	0	x	0	x	0	0	0	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

DMCONTROL [31:0]

Bits	Name	Description
31	HALTREQ	Writes the halt request bit for all currently selected harts. When set to 1, each selected hart will halt if it is not currently halted. Writing 1 or 0 has no effect on a hart which is already halted, but the bit must be cleared to 0 before the hart is resumed. Harts are selected based on the new value of hartsel and hasel being written.
30	RESUMEREQ	Writes the resume request bit for all currently selected harts. When set to 1, each selected hart will resume if it is currently halted. The resume request bit is ignored while the halt request bit is set. Harts are selected based on the new value of hartsel and hasel being written.
28	ACKHAVERESET	Writing 1 to this bit clears the havereset bits for any selected harts. Harts are selected based on the new value of hartsel and hasel being written.
26	HASEL	Selects the definition of currently selected harts.
25-16	HARTSEL	Selecting the target hart to be debugged.
3	SETRESETHALTREQ	This optional field writes the halt-on-reset request bit for all currently selected harts, unless clrrsethaltreq is simultaneously set to 1. When set to 1, each selected hart will halt upon the next deassertion of its reset. The halt-on-reset request bit is not automatically cleared. The debugger must write to clrrsethaltreq to clear it. Writes apply to the new value of hartsel and hasel. If hasresethaltreq is 0, this field is not implemented.
2	CLRRESETHALTREQ	This optional field clears the halt-on-reset request bit for all currently selected harts. Writes apply to the new value of hartsel and hasel.
1	NDMRESET	Controlling reset signal from the Debug Module to the rest of the system.
0	DMACTIVE	Controlling reset signal for Debug Module itself.

DMCONTROL Bits

67.3.3 DMSTATUS (0x11)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RSVD									IMPEBREAK	RSVD		ALLHAVERESET	ANYHAVERESET	ALLRESUMEACK	ANYRESUMEACK	ALLNONEXISTENT	ANYNONEXISTENT	ALLUNAVAIL	ANYUNAVAIL	ALLRUNNING	ANYRUNNING	ALLHALTED	ANYHALTED	AUTHENTICATED	AUTHBUSY	HASRESETHALTREQ	DEVTREEVALID	VERSION				
N/A									RO	N/A		RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO			

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
x	x	x	x	x	x	x	x	x	1	x	x	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0

DMSTATUS [31:0]

Bits	Name	Description
22	IMPEBREAK	Indicates whether there is an implicit EBREAK instruction at the nonexistent word immediately after the program buffer.
19	ALLHAVERESET	Indicates whether all currently selected harts have been reset but the reset has not been acknowledged
18	ANYHAVERESET	Indicates whether any currently selected hart has been reset but the reset has not been acknowledged
17	ALLRESUMEACK	Indicates whether all currently selected harts have acknowledged the previous resume request.
16	ANYRESUMEACK	Indicates whether any currently selected hart has acknowledged the previous resume request.
15	ALLNONEXISTENT	Indicates whether all currently selected harts do not exist in this system.
14	ANYNONEXISTENT	Indicates whether any currently selected hart does not exist in this system.
13	ALLUNAVAIL	Indicates whether all currently selected harts are unavailable.
12	ANYUNAVAIL	Indicates whether any currently selected hart is unavailable.
11	ALLRUNNING	Indicates whether all currently selected harts are running.
10	ANYRUNNING	Indicates whether any currently selected hart is running.
9	ALLHALTED	Indicates whether all currently selected harts are halted.
8	ANYHALTED	Indicates whether any currently selected hart is halted.
7	AUTHENTICATED	Indicates whether authentication check has passed.
6	AUTHBUSY	Indicates whether authentication module is busy.
5	HASRESETHALTREQ	This bit indicates the supported status of resethaltreq.
4	DEVTREEVALID	Whether the information in devtreeaddr0 - devtreeaddr3 registers holds the address of the Device Tree.
3-0	VERSION	Version of the implemented RISC-V External Debug Support. 0x2 indicates that the current implemented version is 0.13.

DMSTATUS Bits

67.3.4 HARTINFO (0x12)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								NSCRATCH				RSVD		DATAACCESS	DATASIZE				DATAADDR												

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
N/A								RO				N/A				RO				RO											
x	x	x	x	x	x	x	x	0	0	1	0	x	x	x	1	0	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0

HARTINFO [31:0]

Bits	Name	Description
23-20	NSCRATCH	Number of dscratch registers available for the debugger to use during program buffer execution, starting from dscratch0.
16	DATAACCESS	The method for accessing the shadowed data registers. The value of this field is 0x1, indicating that the data registers are shadowed in the memory map of the selected hart under Debug Mode. 0: The data registers are shadowed in the hart by CSR registers. 1: The data registers are shadowed in the hart's memory map. Each register takes up 4 bytes in the memory map.
15-12	DATASIZE	Number of 32-bit words in the memory map dedicated to shadowing the data registers.
11-0	DATAADDR	Signed offset for accessing the shadowed data registers by the processor, to be used as offsets for load/store instructions with the zero register as the base register in Debug Mode.

HARTINFO Bits

67.3.5 HALTSUM (0x13)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																															HALT
N/A																															RW
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0

HALTSUM [31:0]

Bits	Name	Description
0	HALT	Bit 0 contains the logical OR of 32 halt bits for hart 0 - hart 31.

HALTSUM Bits

67.3.6 HAWINDOWSEL (0x14)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																HAWINDOWSEL															
N/A																RW															
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

HAWINDOWSEL [31:0]

Bits	Name	Description
14-0	HAWINDOWSEL	This register selects which of the 32-bit portion of the hart array mask register is accessible in hawindow.

HAWINDOWSEL Bits

67.3.7 HAWINDOW (0x15)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																HAWINDOW															
																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

HAWINDOW [31:0]

Bits	Name	Description
31-0	HAWINDOW	This register provides R/W accesses to a 32-bit portion of the hart array mask register. The position of the window is determined by hawindowssel. That is, bit 0 refers to hart (hawindowssel * 32), while bit 31 refers to hart (hawindowssel * 32 + 31).

HAWINDOW Bits

67.3.8 ABSTRACTCS (0x16)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RSVD			PROGBUFSIZE					RSVD												BUSY	RSVD	CMDERR			RSVD			DATACOUNT				
N/A			RO					N/A												RO	N/A	RW1C			N/A			RO				
x	x	x	0	1	0	0	0	x	x	x	x	x	x	x	x	x	x	x	0	x	0	0	0	x	x	x	0	0	1	0	0	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

ABSTRACTCS [31:0]

Bits	Name	Description
28-24	PROGBUFSIZE	Size of the program buffer, in 32-bit words.
12	BUSY	Flag indicating an abstract command is currently being executed.
10-8	CMDERR	Error code indicating that an abstract command fails. The bits in this field remain set until they are cleared by writing 1 to them. No abstract command is started until the value is reset to 0 0: None: no error 1: Busy: an abstract command was executing while command, abstractcs, or abstractauto was written, or when one of the data or progbuf registers was read or written. 2: Not supported: the requested command is not supported. 3: Exception: an exception occurred while executing the command. 4: Halt/resume: an abstract command cannot execute because the hart wasn't in the expected state (running/halted).
4-0	DATACOUNT	Number of data registers that are implemented.

ABSTRACTCS Bits

67.3.9 COMMAND (0x17)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMDTYPE									CONTROL																						
WO									WO																						
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

COMMAND [31:0]

Bits	Name	Description
31-24	CMDTYPE	Controlling the overall functionality of this abstract command.
23-0	CONTROL	The field is interpreted in a command-specific manner.

COMMAND Bits

67.3.10 ABSTRACTAUTO (0x18)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								AUTOEXECPROGBUF								RSVD								AUTOEXECDATA							
N/A								RW								N/A								RW							
x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

ABSTRACTAUTO [31:0]

Bits	Name	Description
23-16	AUTOEXECPROG BUF	When a bit in this field is 1, read or write accesses to the corresponding progbuf word cause the command in command to be executed again.
3-0	AUTOEXECDATA	When a bit in this field is 1, read or write accesses to the corresponding data word cause the command in command to be executed again.

ABSTRACTAUTO Bits

67.3.11 PROGBUF (0x20 + 0x1 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															
RW																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PROGBUF [31:0]

Bits	Name	Description
31-0	DATA	The progbuf registers provide read/write accesses to the program buffer. DM supports program buffer 0-7 only. Program buffer 8-155 are hardwired to 0x00100073 (the EBREAK instruction).

PROGBUF Bits

67.3.12 SBCS (0x38)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SBVERSION			RSVD						SBBUSYERROR	SBBUSY	SBREADONADDR	SBACCESS			SBAUTOINCREMENT	SBREADONDATA	SBERROR			SBASIZE						SBACCESS128	SBACCESS64	SBACCESS32	SBACCESS16	SBACCESS8	
RO			N/A						RW1C	RO	RW	RW			RW	RW	RW1C			RO						RO	RO	RO	RO	RO	
0	0	1	x	x	x	x	x	x	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	1

SBCS [31:0]

Bits	Name	Description
31-29	SBVERSION	The version of the supported System Bus Access version. It is currently 1 for v0.13 of the RISC-V External Debug Support specification.
22	SBBUSYERROR	Indicate the debugger attempts to execute system bus access before sbbusy is cleared.
21	SBBUSY	Indicate the system bus master is busy.
20	SBREADONADDR	Every write to sbaddress0 automatically triggers a system bus read at the new address.
19-17	SBACCESS	Access size. 0: 8-bit 1: 16-bit 2: 32-bit 3: 64-bit 4: 128-bit
16	SBAUTOINCREMENT	sbaddress is incremented by the size specified in sbaccess_ after every system bus access.
15	SBREADONDATA	Every read from sbdata0 automatically triggers a system bus read at the (possibly auto-incremented) address.
14-12	SBERROR	Error code indicating the failure type of the system bus accesses. Write 1 to clear the status. 0: No bus error 1: Timeout error 2: Bad address 3: Alignment error 4: Unsupported size was requested 7: Others
11-5	SBASIZE	Address width of System bus addresses in bits. This field is 0 if there is no bus access support. Otherwise, it is the value of the SYS_ADDR_WIDTH parameter.

Bits	Name	Description
4	SBACCESS128	This bit indicates the supported status of 128-bit system bus data accesses.
3	SBACCESS64	This bit indicates the supported status of 64-bit system bus data accesses. This bit is 1 if SYS_DATA_WIDTH == 64.
2	SBACCESS32	This bit indicates the supported status of 32-bit system bus data accesses.
1	SBACCESS16	This bit indicates the supported status of 16-bit system bus data accesses.
0	SBACCESS8	This bit indicates the supported status of 8-bit system bus data accesses.

SBCS Bits

67.3.13 SBADDRESS (0x39 + 0x1 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																SBADDRESS															
																RO															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SBADDRESS [31:0]

Bits	Name	Description
31-0	SBADDRESS	bit of the address in sbaddress

SBADDRESS Bits

67.3.14 SBDATA (0x3C + 0x1 * n)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																SBDATA															
																RW															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SBDATA [31:0]

Bits	Name	Description
31-0	SBDATA	bit of the address in sbdata

SBDATA Bits

68 Version Information

Date	version	Description
Rev0.1	2025/04/24	internal version Rev 0.1 released.
Rev0.2	2025/09/29	internal version Rev 0.2 released. Added BGA121 Package。 Improved the GPIO chapter。
Rev0.3	2025/11/20	internal version Rev0.3 released. Added a power system block diagram without the built-in DCDC. Added notes in the ADC chapter that the precision is affected by the built-in DCDC.

Table 241: Version Information

69 Disclaimer

HPMicro Semiconductor Co., Ltd. (hereinafter referred to as "HPMicro") reserves the right to revise, amend, enhance and modify its products and/or this document at any time without prior notice. Users may obtain the latest relevant information on HPMicro's official website <https://www.hpmicro.com>

The information contained in this disclaimer supersedes and replaces all information stated in any previous versions.

List of Tables

List of Tables

1	Summary of Peripheral Abbreviations	3
2	register description acronym list	9
3	CSR Register List	20
4	Event Selectors	46
5	PMPCFG [7:0]	53
6	PMPCFG Bit Fields	54
7	CCTL Command Definition	76
8	CCTL Commands Which Access mcctldata	76
9	CCTL Command Definition	89
10	PMACFG [7:0]	102
11	PMACFG Bit Fields	103
12	IRQ List	112
13	MCHTMR Register List	116
14	PLIC Register List	122
15	PLIC_SW Register List	127
16	Power and Ground Pin Descriptions	133
17	Power supply and package pin reuse	133
18	Power Management Related IO	134
19	Clock source selection and default frequency	137
20	Function Clock Summary	139
21	Module Clock List	140
22	PDGO Register List	144
23	DCDC low power consumption state control	169
24	PCFG Register List	172
25	PPOR Register List	184
26	Functional Module Connection Bit Table 0	194
27	Functional Module Connection Bit Table 1	195
28	Functional Module Connection Bit Table 2	195
29	RETENTION Register Bit Table	198
30	Measurement Clock Selection Table	200
31	SYSCTL Register List	210
32	PLLCTLV2 Register List	233
33	Address space allocation	245
34	OTP List	246
35	Command Data Structure	254
36	Query Runtime Environment (query-rte) Command Data Structure	254
37	ROM Parameters Response Packet	256
38	Active Peripheral Information Packet	256

39	Last ROM Boot Status	257
40	Storage Device Properties Response Packet	258
41	Configure Runtime Environment (configure-rte)	258
42	Configure Runtime Environment Response Packet	259
43	Configure Storage (configure-memory)	259
44	configure-memory Response Frame Data Structure	260
45	XPI NOR Configuration Options (XPI NOR Configuration Option)	262
46	Configure Storage (write-memory) - Command + Data	263
47	Configure Storage (write-memory) - Data Only	263
48	write-memory Response Frame Data Structure	264
49	read-memory Command Data Frame	264
50	read-memory Response Frame Data Structure	265
51	read-memory Response Frame Data Structure	265
52	load-image Data Frame Structure	266
53	load-image Response Frame Data Structure	266
54	erase Data Frame Structure	267
55	erase Response Frame Data Structure	267
56	reset Data Frame Structure	267
57	reset Response Frame Data Structure	268
58	Payload Packet Data Structure	268
59	Acknowledgment Packet	269
60	USB Descriptors	270
61	USB-HID Payload Packet Data Structure	270
62	USB-HID Acknowledgment Packet Data Structure	271
63	Firmware Container Header	271
63	Firmware Container Header	272
63	Firmware Container Header	273
63	Firmware Container Header	273
64	Firmware Container Header	274
65	Configuration Info Block Header	275
66	Configuration Info Header	275
67	XPI NOR Configuration Options	277
68	XPI NOR Configuration Block	278
69	XPI NOR Configuration Block	280
70	Wake-up Verification Info	280
71	OTP Unlock Info	280
72	Signature Block Header	282
73	Root Key Table Header	282
74	Root Key Table Item	283
75	Signature Info	283
76	Certificate Header	284
77	Command Container	284
78	Command Container Header	284

79	Configure Storage Device Data Block	285
80	Write to Storage Device Data Block	286
81	Erase Storage Device Data Block	286
82	Reset Data Block	286
83	BootROM Related OTP Mapping Table	304
84	XPI0 Pin Information	305
85	UART0 Pins	306
86	USB0 Pins	306
87	BOOT_MODE Pins	306
88	BootROM Occupied Registers	307
89	BootROM Memory Mapping	307
90	BootROM Lifecycle Codes	308
91	SOC IOMUX	352
92	ACMP0 SIGNAL PIN	353
93	ADC0 SIGNAL PIN	354
94	ADC1 SIGNAL PIN	354
95	ESC0 HIGH SPEED SIGNAL PIN	356
96	ESC0 NON-HIGH SPEED SIGNAL PIN	362
97	ETH0 HIGH SPEED SIGNAL PIN	363
98	ETH0 NON-HIGH SPEED SIGNAL PIN	364
99	EUI0 SIGNAL PIN	364
100	EUI1 SIGNAL PIN	364
101	EWDOG0 SIGNAL PIN	365
102	EWDOG1 SIGNAL PIN	365
103	GPIO SIGNAL PIN	369
104	GPTMR0 SIGNAL PIN	370
105	GPTMR1 SIGNAL PIN	371
106	GPTMR2 SIGNAL PIN	371
107	GPTMR3 SIGNAL PIN	371
108	I2C0 SIGNAL PIN	372
109	I2C1 SIGNAL PIN	372
110	I2C2 SIGNAL PIN	372
111	I2C3 SIGNAL PIN	373
112	JTAG SIGNAL PIN	373
113	MCAN0 SIGNAL PIN	374
114	MCAN1 SIGNAL PIN	374
115	MCAN2 SIGNAL PIN	375
116	MCAN3 SIGNAL PIN	375
117	OWR0 SIGNAL PIN	375
118	OWR1 SIGNAL PIN	376
119	PPI0 HIGH SPEED SIGNAL PIN	378
120	PWM0 HIGH SPEED SIGNAL PIN	379
121	PWM1 HIGH SPEED SIGNAL PIN	380

122	QEIO SIGNAL PIN	380
123	QEI1 SIGNAL PIN	381
124	QEO0 SIGNAL PIN	382
125	QEO1 SIGNAL PIN	382
126	SDM0 HIGH SPEED SIGNAL PIN	383
127	SOC SIGNAL PIN	384
128	SPI0 HIGH SPEED SIGNAL PIN	384
129	SPI1 HIGH SPEED SIGNAL PIN	384
130	SPI2 HIGH SPEED SIGNAL PIN	385
131	SPI3 HIGH SPEED SIGNAL PIN	385
132	SYSCTL SIGNAL PIN	386
133	TRGM SIGNAL PIN	388
134	UART0 SIGNAL PIN	388
135	UART1 SIGNAL PIN	389
136	UART2 SIGNAL PIN	389
137	UART3 SIGNAL PIN	389
138	UART4 SIGNAL PIN	390
139	UART5 SIGNAL PIN	390
140	UART6 SIGNAL PIN	391
141	UART7 SIGNAL PIN	391
142	USB0 SIGNAL PIN	391
143	XPI0 HIGH SPEED SIGNAL PIN	393
144	Boot Configuration Table	394
145	Special Function Pin Configuration	394
146	IO reset status table	394
147	IOC Register List	407
148	GPIO Register List	422
149	GPIOM Register List	444
150	EUI Port Interface Description	448
151	EUI Register List	451
152	LOBS Register List	470
153	PGPR Register List	487
154	Summary of Read and Write Protection for IDENTITY Type OTP Words	500
155	Summary of Read and Write Protection for SECURE Type OTP Words	500
156	Summary of Read and Write Protection for SECRET Type OTP Words	500
157	Summary of Read and Write Protection for GENERAL Type OTP Words	501
158	OTP Register List	508
159	PPI pin description	520
160	PPI Register List	531
161	Operators Supported by TFA	539
162	CRC Register List	542
163	DMA MUX List	549
164	DMA Pin Description	550

165	DMA Chained Task Descriptor	552
166	DMAV2 Register List	562
167	DMAMUX Register List	577
168	MBX Register List	579
169	TRGM TRIGMUX INPUT List	593
170	TRGM TRIGMUX OUTPUT List	597
171	TRGM DMAMUX List	597
172	TRGM_FILTER list	599
173	TRGM_POS_MATRIX list	599
174	TRGM_ADC_MATRIX list	600
175	TRGM_DAC_MATRIX list	600
176	PWMV2 Register List	622
177	TRGM Register List	657
178	SYNT Register List	667
179	QEI Pin Description	672
180	UVW Phase Position Selection 0	676
181	UVW Phase Position Selection 1	677
182	QEIV2 Register List	688
183	QEOV2 Register List	730
184	PLB Register List	756
185	Sinc Filter Order	763
186	Sinc Filter Output Signal Upper and Lower Bounds	764
187	Relationship Between Number of Initially Incorrect Samples and Filter Type	766
188	Amplitude Path Sinc Filter Output Signal Upper and Lower Bounds	768
189	SDM Register List	770
190	GPTMR Register List	787
191	EWDG Register List	799
192	UART Register List	813
193	Command Values Supported in SPI Slave Mode	831
194	SPI Register List	834
195	I2C Register List	853
196	CAN Pin Description	865
197	DLC Meaning	867
198	Frame Transmission Configuration	879
199	MCAN Register List	882
200	PTPC Register List	928
201	OWR Pin Description	936
202	OWR Register List	939
203	USB pin description	945
204	qhead structure	951
205	dqh structure	955
206	USB Register List	957
207	Ethernet ii Frame Format	1006

208	802.3 Basic Frame Format	1007
209	802.3 Frame Format with Tag	1007
210	MII Interface Signal Description	1008
211	RMII Interface Signal Description	1009
212	RGMII Interface Signal Description	1009
213	MDIO Interface Data Packet Field Description	1010
215	TDES0 Detailed Field Description	1025
216	Detailed Field Description for TDES1	1025
217	Detailed Field Description for TDES2	1026
218	Detailed Field Description for TDES3	1026
219	Detailed Field Description for RDES0	1028
220	Detailed Field Description for RDES1	1029
221	Detailed Field Description for RDES2	1029
222	Detailed Field Description for RDES3	1029
223	Detailed Field Description for RDES4	1031
224	Detailed Field Description for RDES6	1031
225	Detailed Field Description for RDES7	1031
226	ENET Register List	1036
227	ESC Register List	1154
228	Built-in Ethernet PHY pin	1236
229	IO connection between main chip and built-in Ethernet PHY	1237
230	ENET_PHY register list	1237
231	ADC Preemption Queue and Hardware Trigger Port	1249
232	ADC preemption conversion DMA data buffer allocation	1253
233	ADC16 Register List	1259
234	ACMP Register List	1277
235	TSNS Register List	1282
236	EXIP Key Wrap Data Summary	1293
237	EXIP Key Wrap Data Location in External Flash	1293
238	DTM Instruction List	1297
239	DM system memory mapping	1301
240	DM Register List	1302
241	Version Information	1311

List of Figures

List of Figures

1	System Architecture Diagram	1
2	MCHTMR block diagram	115
3	power domain diagram	129
4	Block diagram of a power supply system using built-in DCDC	130
5	Block diagram of a power supply system without built-in DCDC	131
6	Clock System Diagram	136
7	ADC functional clock structure	139
8	Chain Relationship of Resource Nodes	191
9	Chain Relationship between Resource Nodes and Group/CPU	196
10	BootROM Boot Flow Diagram	248
11	XPI QSPI NOR FLASH CA Port Connection	249
12	XPI QSPI NOR FLASH CB Port Connection	249
13	XPI Octal NOR FLASH Connection	250
14	XPI NOR Boot Flow	251
15	XPI NOR Boot Image Layout	252
16	Serial Communication Protocol	253
17	Boot Image Layout	271
18	Layout of Parts in Signature Block	281
19	Firmware Header Part to be Signed	287
20	General IO peripheral multiplexing function selection	395
21	General IO GPIO control options	396
22	EUI Block Diagram	445
23	Schematic Diagram of EUI Driving 7-Segment Digital Tubes and Receiving from Shift Register	446
24	Schematic Diagram of EUI Driving 16-Segment Digital Tubes and Receiving from Shift Register	447
25	Schematic Diagram of EUI Digital Tube Drive Timing Parameters	448
26	DMA Structure Block Diagram	550
27	PWM module external connection diagram	585
28	QEI module external connection diagram	586
29	QEO module external connection diagram	587
30	SDM module external connection diagram	588
31	PWMv2 system diagram	602
32	Schematic diagram of PWMv2 counter calculation unit	603
33	PWM output schematic, $b \leq T$	605
34	PWM output schematic, $a \leq T, b > T$	605
35	PWM output schematic, $a > T, b > T$	605
36	PWM Burst output	607
37	PWMv2 General Purpose Computing Unit	609
38	PWMv2 General Purpose Computing Unit Limiting	610

39	PWMv2 forced output schematic diagram	612
40	PWMv2 fail-safe recovery	614
41	QEI Architecture Diagram	672
42	Forward Rotation Phase Counting Diagram	673
43	Reverse Rotation Phase Counting Diagram	673
44	Z Phase Aligned with Phase A Edge Diagram	674
45	PD Mode Diagram	675
46	Up-Down (UD) Decoding Mode Diagram	675
47	UVW Three-Phase Signal Position Encoding Diagram	676
48	ADC Amplitude/Phase Compensation	678
49	Pulse-based Speed Measurement	680
50	Duration-based Speed Measurement	681
51	QEO Block Diagram	723
52	Example of Sine Wave Output with Line Conversion Set to 3 and 120° Phase Shift	724
53	ABZ Output Waveforms (Z_25, Z_75, Z_100 are three Z-phase output modes)	726
54	SDM Block Diagram	760
55	Filter Block Structure Diagram	762
56	Manchester Mode Transmission Waveform	763
57	Timer Block Diagram	781
58	watchdog block diagram	795
59	UART Block Diagram	809
60	SPI Block Diagram	828
61	SPI Data Timing Diagram	829
62	Two SPI Transmission Formats	830
63	DSPI Address Two-Line Transmission Format Diagram	830
64	QSPI Address Four-Line Transmission Format Diagram	831
65	I2C Block Diagram	849
66	CAN Structure Block Diagram	865
67	CAN Bit Time	866
68	CAN Silent Mode	868
69	CAN External Loopback	868
70	CAN Internal Loopback	869
71	CAN Receive Buffer Data Format	870
72	CAN Transmit Buffer Data Format	871
73	CAN Transmit Event Data Format	873
74	CAN Standard Message ID Filter	874
75	CAN Extended Message ID Filter	875
76	Message Memory Schematic	877
77	OWR Initialization Sequence	937
78	Write Slot Timing	937
79	Read Slot Timing	938
80	Asynchronous Queue Diagram	947
81	qhead schematic	948

82	qtd schematic diagram	951
83	dqhlist diagram	953
84	dqh schematic diagram	953
85	dtd schematic diagram	953
86	ENET Module Structure Diagram	1006
87	Network Time Synchronization Diagram	1018
88	Fine Time Correction System Time Schematic	1020
89	Transmit Descriptor	1022
90	Receive Descriptor	1026
91	ESC clock structure diagram	1142
92	ESC reset function diagram	1143
93	ESC Link function diagram	1144
94	Functional Diagram of ESC SYNC and LATCH	1145
95	ESC GPIO function diagram	1146
96	ADC16 Block Diagram	1243
97	ADC16 clock structure diagram	1243
98	ACMP block diagram	1274
99	ACMP comparison result processing	1275
100	debugging system diagram	1296