



SUNWAY 申威

# 申威 411 处理器 I/O 寄存器手册

2017 年 10 月

成都申威科技有限责任公司



## 免责声明

本档仅提供阶段性信息，所含内容可根据产品的实际情况随时更新，恕不另行通知。如因档使用不当造成的直接或间接损失，本公司不承担任何责任。

### 成都申威科技有限责任公司

Chengdu Sunway Technology Corporation Limited

地址：成都市华府大道四段电子科大科技园 D22 栋

Building D22, National University Science and technology park,

Section 4, Huafu Avenue, Chengdu

Mail: sales@swcpu.cn

Tel : 028-68769016

Fax: 028-68769019



## 阅读指南

《申威 411 处理器 I/O 寄存器手册》主要描述了申威 411 处理器各类寄存器等内容。

## 文档修订

文档更新记录	文档名	申威 411 处理器 IO 寄存器手册
	版本号	V1.0
	创建人	研发部
	创建日期	2017-10-8

## 版本更新

版本号	更新内容	更新日期
V1.0	初稿	2017-10-8

## 技术支持

可通过邮箱或问题反馈网站向我司提交产品使用的问题，并获取技术支持。

售后服务邮箱：[sales@swcpu.cn](mailto:sales@swcpu.cn)

问题反馈网址：<http://www.swcpu.cn/>

# 目 录

<b>1</b>	<b>概述</b> .....	<b>1</b>
1.1	术语与约定.....	1
1.2	IO 寄存器的编址.....	2
1.2.1	IO 寄存器使用的 IO 空间.....	2
1.2.2	存储器接口 IO 寄存器编址.....	2
1.2.3	一致性处理部件IO 寄存器编址.....	2
1.2.4	存储控制器的IO 寄存器编址.....	3
1.2.5	存储器地址/命令/数据接口IO 寄存器编址.....	5
1.2.6	DDR3PHY_FPGA 寄存器编址.....	10
1.2.7	1.2.3 系统接口 IO 寄存器编址.....	11
1.2.8	维护接口 IO 寄存器编址.....	14
1.2.9	PCI-E 接口的 IO 空间划分与寄存器编址.....	17
1.3	IO 寄存器的访问机制.....	21
<b>2</b>	<b>存储器接口的 IO 寄存器</b> .....	<b>21</b>
2.1	一致性处理部件的 IO 寄存器.....	21
2.1.1	CPM_MC0Disable / CPM_MC1Disable.....	21
2.1.2	CPM_CrossMode.....	21
2.1.3	CPM_FRQCmprEn.....	22
2.1.4	CPM_PerfCtrl.....	22
2.1.5	CPM_CoreSleep.....	23
2.1.6	CPM_ProbeOK.....	23
2.1.7	CPM_SerrThres.....	24
2.1.8	CPM_SerrCnt.....	24
2.1.9	CPM_Fault.....	24
2.1.10	CPM_FaultEn.....	26
2.1.11	CPM_TCReqCnt.....	28
2.1.12	CPM_TCHitCnt.....	28
2.1.13	CPM_TCMissCnt.....	28
2.1.14	CPM_AddrCfltCnt.....	29
2.1.15	CPM_FRQCmprCnt.....	29
2.1.16	CPM_TMAFMax.....	29
2.2	存储控制器内部的 IO 寄存器.....	30
2.2.1	CfgTR0.....	30
2.2.2	CfgTR1.....	31
2.2.3	CfgTR2.....	31
2.2.4	CfgTR3.....	32
2.2.5	CfgTR4.....	32
2.2.6	CfgTR5.....	33
2.2.7	CfgTR6.....	33
2.2.8	CfgTR7.....	34
2.2.9	CfgDec.....	34
2.2.10	CfgCR.....	36
2.2.11	MC_CTRL.....	38
2.2.12	INIT_CTRL.....	40
2.2.13	ClosePage.....	41
2.2.14	DTUR.....	41
2.2.15	DTUW.....	41
2.2.16	DTUWDO.....	40

2.2.17	DTUWD1 .....	40
2.2.18	DTURD0 .....	40
2.2.19	DTURD1 .....	40
2.2.20	DTUCTL .....	42
2.2.21	DTUPREA .....	42
2.2.22	WL_CTRL .....	42
2.2.23	MEMSERR_P .....	43
2.2.24	CfgErr .....	43
2.2.25	FSMSTAT .....	44
2.2.26	RC .....	45
2.2.27	RCMRD .....	46
2.2.28	6U .....	46
2.2.29	PUB_INTERFACE .....	46
2.2.30	POWERCTRL .....	47
2.2.31	PUB.DTO_CTL .....	47
2.2.32	CfgMR0 .....	48
2.2.33	CfgMR1 .....	49
2.2.34	CfgMR2 .....	50
2.2.35	CfgMR3 .....	51
2.2.36	PERF_CTRL .....	51
2.2.37	MC_PERF0 .....	52
2.2.38	MEMSERR_CNT .....	52
2.2.39	MEMSERR_TH .....	52
2.2.40	WRDATASERR_CNT .....	53
2.2.41	MC_PERF1 .....	54
2.2.42	RWCNTOFREF .....	54
2.3	存储器地址/命令/数据接口相关 IO 寄存器 .....	54
2.3.1	RIDR .....	54
2.3.2	PIR .....	55
2.3.3	PGCR0-1 .....	57
2.3.4	PGSR0-1 .....	61
2.3.5	PLLCR .....	62
2.3.6	PTRO-4 .....	64
2.3.7	ACMDLR .....	66
2.3.8	ACBDLR .....	66
2.3.9	ACIOCR .....	67
2.3.10	DXCCR .....	69
2.3.11	DSGCR .....	70
2.3.12	DCR .....	71
2.3.13	DTPRO-2 .....	73
2.3.14	MRO-3 .....	71
2.3.15	ODTCR .....	77
2.3.16	DTCR .....	77
2.3.17	DTARO-3 .....	79
2.3.18	DTDRO-1 .....	81
2.3.19	DTEDR .....	82
2.3.20	DCUAR .....	82
2.3.21	DCUDR .....	83
2.3.22	DCURR .....	83
2.3.23	DCULR .....	84
2.3.24	DCUGCR .....	84
2.3.25	DCUTPR .....	85
2.3.26	DCUSRO-1 .....	85
2.3.27	BISTR .....	85

2.3.28	BISTWCR .....	88
2.3.29	BISTMSKRO-2 .....	88
2.3.30	BISTLSR .....	90
2.3.31	BISTARO-2.....	90
2.3.32	BISTUDPR.....	91
2.3.33	BISTGSR.....	92
2.3.34	BISTWER .....	92
2.3.35	BISTBERO-3 .....	92
2.3.36	BISTWCSR .....	94
2.3.37	BISTFWRO-2 .....	94
2.3.38	ZQnCR0-1.....	95
2.3.39	ZQnSR0-1 .....	96
2.3.40	DXnGCR.....	97
2.3.41	DXnGSR0-1.....	100
2.3.42	DXnBDLR0-4.....	101
2.3.43	DXnLCDLR0-2.....	102
2.3.44	DXnMDLR.....	104
2.3.45	DXnGTR.....	104
2.4	DDR3PHY_FPGA 的 IO 寄存器 .....	105
2.4.1	写延时参数调节寄存器 .....	105
2.4.2	读延时参数调节寄存器 .....	107
2.4.3	读 Gate 延时参数调节寄存器.....	101
2.4.4	读响应延时调节寄存器 .....	102
2.4.5	卸载点调节寄存器 .....	102
2.4.6	CfgCR 寄存器 .....	102
2.4.7	CORE_ENABLE 寄存器 .....	103
2.4.8	UPDATE 寄存器 .....	103
2.4.9	MR 寄存器.....	104
2.4.10	读 Gate 宽度调整寄存器 .....	105
2.4.11	写 Gate 延时参数调节寄存器 .....	106
2.4.12	写 Gate 宽度调整寄存器 .....	107
2.4.13	Write Leveling 调整寄存器 .....	108
3	系统接口的 IO 寄存器.....	110
3.1	系统接口 IO 寄存器列表 .....	110
3.1.1	CORE_SLEEP_STAT .....	110
3.1.2	DEVINT_MIS .....	110
3.1.3	IPU_CTRL .....	111
3.1.4	SHORT_CLK_START .....	111
3.1.5	SHORT_CLK_MSK.....	112
3.1.6	MT_INDX .....	112
3.1.7	SHORT_INDX.....	112
3.1.8	INTEN.....	112
3.1.9	LOCKEN.....	113
3.1.10	FMT_ERR .....	113
3.1.11	SYSERR_INDX .....	114
3.1.12	NMI_INDX.....	114
3.1.13	IPU_STAT .....	114
3.1.14	SERR_CNTTH.....	115
3.1.15	MCUSERR_CNT .....	115
3.1.16	PIUOSERR_CNT.....	116
3.1.17	PIU1SERR_CNT.....	116
3.1.18	FXSERR_CNT .....	116

3.1.19	ERRRPT_EN .....	116
3.1.20	SYSERR_NMI_INT_DST .....	117
3.1.21	IPU_STAT_ERREN .....	118
3.1.22	PCIE0_INTx_INDX.....	118
3.1.23	PCIE1_INTx_INDX.....	119
3.1.24	IINT_MIS .....	119
3.1.25	IINTMIS_RPTEN .....	120
3.1.26	DEVINT_MIS_RPTEN.....	120
3.1.27	ECCSERR.....	121
3.1.28	ECCSERR_RPTEN.....	121
3.1.29	ECCMERR .....	121
3.1.30	ECCMERR_RPTEN .....	122
3.1.31	DEVINT_WKEN.....	122
3.1.32	MAIL_BOX_i.....	122
3.1.33	MAIL_STAT_i .....	123
3.1.34	PCIE0_MSI[i]_INDX .....	123
3.1.35	PCIE1_MSI[i]_INDX .....	124
<b>4</b>	<b>维护接口的 IO 寄存器.....</b>	<b>125</b>
4.1	维护 IOR 的复位说明.....	125
4.2	维护 IOR 的寄存器说明 .....	125
4.2.1	MC_ONLINE.....	125
4.2.2	CG_FAULT_STAT .....	125
4.2.3	CG_FAULT_EN .....	128
4.2.4	CG_FAULT_INT_EN .....	130
4.2.5	INIT_CTL.....	132
4.2.6	BIST_CTL .....	130
4.2.7	BIST_OVERTIME_TH .....	132
4.2.8	MT_STATE.....	132
4.2.9	CORE_ONLINE .....	135
4.2.10	MT_INT .....	135
4.2.11	MT_INT_END .....	135
4.2.12	CPUID .....	136
4.2.13	SOFT_INFO~15.....	136
4.2.14	LONG_TIME .....	136
4.2.15	LONG_PRESET.....	136
4.2.16	TESTSEL.....	137
4.2.17	IO_START .....	137
4.2.18	PERST_N_PCIE0~1.....	137
4.2.19	BUTTON_RST_N_PCIE0~1 .....	139
4.2.20	MTRSPMISS .....	139
4.2.21	SLEEP_DONE0~3 .....	139
4.2.22	PLL_CHG_CNT.....	140
4.2.23	CG_BIST_STAT .....	140
4.2.24	BIST_STAT.....	140
4.2.25	TAPSEL .....	140
4.2.26	ST_NEXT_FLAG.....	142
4.2.27	SERR_CNTTH.....	142
4.2.28	IPUSERR_CNT.....	143
4.2.29	PIUxSERR_CNT.....	143
4.2.30	ERRRPT_EN .....	143
4.2.31	ERR_INF .....	144
4.2.32	SCAN_WAIT_CFG.....	146
4.2.33	CLU_TEST .....	148



4.2.34	PC_BIST_STAT.....	148
4.2.35	Corex_ERRRST_N.....	149
4.2.36	SI_FAULT_STAT .....	149
4.2.37	SI_FAULT_EN .....	150
4.2.38	SI_FAULT_INT_EN .....	150
4.2.39	CLK_SEL_PCIE0~1 .....	151
4.2.40	FIFO_SYNSEL.....	151
4.2.41	CPU_INFO.....	152
4.2.42	FlagReg .....	152
4.2.43	OP_MASK.....	152
4.2.44	BIST_GOON.....	153
4.2.45	WAKEUP_CTL .....	153
4.2.46	NMI_CTL .....	153
4.2.47	PIUPLL_CNT.....	154
<b>5</b>	<b>PCI-E 接口 IO 寄存器.....</b>	<b>155</b>
5.1	PCI-E 接口控制部件 IO 寄存器.....	155
5.1.1	PIU 内部 IOR0 部分寄存器 .....	155
5.1.2	PIU 内部 IOR1 部分寄存器.....	169
5.2	PCI-E 根部件内部的 IO 寄存器.....	179
5.2.1	RC 寄存器空间布局 .....	179
5.2.2	PCI 兼容 配置头寄存器 .....	185
5.2.3	PCI 标准能力结构寄存器 .....	194
5.2.4	PCI-E 扩展能力寄存器 .....	207
5.2.5	端口逻辑寄存器 .....	212
附录A	PCI-E术语表.....	222
附录B	PCI-E链路状态编码.....	224

# 1 概述

IO 寄存器（IOR）是申威 411 处理器内部存储器接口、系统接口部件、PCI-E 接口和维护接口 用于控制或回收状态的控制与状态寄存器，这些寄存器可以通过核心进行读写访问，也可以通过维护接口的维护命令进行读写访问。

申威 411 处理器的 IO 寄存器按照所在部件的不同，分成四大类，分别是：

- 1) 存储器接口 IO 寄存器：每个存储控制器及其对应的 DDR3 存储器接口，各有一组 IO 寄存器（共两组）；
- 2) 系统接口 IO 寄存器：系统接口控制部件内的一组 IO 寄存器；
- 3) 维护接口 IO 寄存器：维护接口部件内的一组 IO 寄存器；
- 4) PCI-E 接口 IO 寄存器：每个 PCI-E 接口（包括 PCI-E 控制部件和 PCI-E 根部件），各有一组 IO 寄存器（共两组）。

## 1.1 术语与约定

本手册对 IO 寄存器内容的软件访问特性和复位状态进行描述时，遵循如下符号约定：

- 1) “W” 指示可写；
- 2) “WO” 指示只写；
- 3) “R” 指示可读；
- 4) “RO” 指示只读；
- 5) “RC” 指示读清除；
- 6) “WC” 指示写清除；
- 7) “RW” 指示可读写；
- 8) “RWC” 指示可读且写清除；
- 9) “W1C” 指示写“1”清除，写“0”无影响；
- 10) “RO,x” 指示只读且复位时初值为“x”；
- 11) “WO,x” 指示只写且复位时初值为“x”；
- 12) “RW,x” 指示可读写且复位时初值为“x”；
- 13) “SEXT(x)” 指示将“x”扩展指定位数；
- 14) PCI-E 的 RC 配置空间寄存器的属性描述，遵循表 5-52 的约定。以上“x”可以为“0”，也可以为“非 0 的确定值”，没有指定复位设置值的 IO 寄存器在复位后是一个不确定的值，需要在复位后进行初始化设置。读保留位时，返回“0”；写保留位时，写入的值

被忽略。

本手册中数字的表示方法如下：

- 1) 十六进制：数字前缀为“0x”或者“n’h”（其中 n 为 16 进制数的位数）；
- 2) 十进制：没有前缀的数字；
- 3) 二进制：数字前缀为“0b”或者“n’b”（其中 n 为 2 进制数的位数）。

## 1.2 IO 寄存器的编址

### 1.2.1 IO 寄存器使用的 IO 空间

申威 411 处理器内部的 IO 寄存器统一编址在 IO 空间，即 IO 寄存器对应的物理地址 PA[39]固定为“1”，物理地址 PA[38:36]区分不同部件的 IO 寄存器使用的 IO 空间，物理地址 PA[35:7]作为 IO 寄存器的索引地址。在进行 IO 寄存器访问时 PA[6:0]必须为全“0”。不同部件的 IO 寄存器对应的物理地址 PA[38:36]位如下：

- 1) PA[38:36]=“0b000”，对应存储器接口 IO 寄存器；
- 2) PA[38:36]=“0b100”，对应 PCI-E 0 接口 IO 寄存器；
- 3) PA[38:36]=“0b101”，对应维护接口 IO 寄存器；
- 4) PA[38:36]=“0b110”，对应系统接口控制 IO 寄存器；
- 5) PA[38:36]=“0b111”，对应 PCI-E 1 接口 IO 寄存器；
- 6) 其他保留。

### 1.2.2 存储器接口 IO 寄存器编址

PA[35:26]为全零，PA[25:23]表示一致性处理部件或存储器接口的 IOR，具体如下：

PA[25:23]=“0b000”，对应一致性处理部件的 IO 寄存器； PA[25:23]=“0b100”，对应存储控制器 0 的 IO 寄存器； PA[25:23]=“0b101”，对应存储器地址/命令/数据接口 0 的 IO 寄存器； PA[25:23]=“0b110”，对应存储控制器 1 的 IO 寄存器； PA[25:23]=“0b111”，对应存储器地址/命令/数据接口 1 的 IO 寄存器。

### 1.2.3 一致性处理部件 IO 寄存器编址

一致性处理部件（CPM）IO 寄存器编址如表 1-1 所示。

表 1-1：一致性处理部件 IO 寄存器编址

序号	名称	符号	索引地址	存取特性
----	----	----	------	------

			PA[22:7]	
1	MC#0 不使能寄存器	CPM_MC0Disable	0x0000	RW
2	MC#1 不使能寄存器	CPM_MC1Disable	0x0001	RW
3	地址交叉模式寄存器	CPM_CrossMode	0x0002	RW
4	FRQ 压缩使能寄存器	CPM_FRQCmprEn	0x0004	RW
5	性能计数器控制寄存器	CPM_PerfCtrl	0x0005	RW
6	核心睡眠使能寄存器	CPM_CoreSleep	0x0006	RW
7	CPM 外部 Probe 完成标志	CPM_ProbeOK	0x0007	RO
8	单错计数阈值寄存器	CPM_SerrThres	0x0008	RW
9	CPM 单错计数器	CPM_SerrCnt	0x0009	RWC
10	故障状态寄存器	CPM_Fault	0x000A	RWIC
11	故障报错使能寄存器	CPM_FaultEn	0x000B	RW
12	可 Cache 请求计数器	CPM_TCReqCnt	0x0010	RO
13	TCache 直接命中计数器	CPM_TCHitCnt	0x0011	RO
14	TCache 脱靶计数器	CPM_TCMissCnt	0x0012	RO
15	地址冲突计数器	CPM_AddrCfltCnt	0x0013	RO
16	FRQ 压缩计数器	CPM_FRQCmprCnt	0x0014	RO
17	TMAF 最大使用情况寄存器	CPM_TMAFMax	0x0015	RO

## 1.2.4 存储控制器的IO 寄存器编址

存储控制器的 IO 寄存器编址如表 1-2 所示。

表 1-2: 存储控制器 IO 寄存器编址

序号	名称	符号	索引地址 PA[22:7]	存取 特性
1	时序寄存器 0	CfgTR0	0x0000	RW
2	时序寄存器 1	CfgTR1	0x0001	RW
3	时序寄存器 2	CfgTR2	0x0002	RW
4	时序寄存器 3	CfgTR3	0x0003	RW
5	时序寄存器 4	CfgTR4	0x0004	RW
6	时序寄存器 5	CfgTR5	0x0005	RW

7	时序寄存器 6	CfgTR6	0x0006	RW
8	时序寄存器 7	CfgTR7	0x0007	RW
9	配置译码寄存器	CfgDec	0x0008	RO
10	配置寄存器	CfgCR	0x0009	RW
11	存储控制器控制寄存器	MC_CTRL	0x000a	RW
12	存控初始化控制寄存器	INIT_CTRL	0x000b	RW
13	页关闭策略控制寄存器	ClosePage	0x000c	RW
14	数据通路信号调节读测试寄存器	DTUR	0x000d	RW
15	数据通路信号调节写测试寄存器	DTUW	0x000e	RW
16	数据通路信号调节写测试数据寄存器 0	DTUWD0	0x000f	RW
17	数据通路信号调节写测试数据寄存器 1	DTUWD1	0x0010	RW
18	保留	—	0x0011	—
19	保留	—	0x0012	—
20	数据通路信号调节读测试数据寄存器 0	DTURD0	0x0013	RO
21	数据通路信号调节读测试数据寄存器 1	DTURD1	0x0014	RO
22	保留	—	0x0015	—
23	数据通路信号调节测试控制寄存器	DTUCTL	0x0016	RW
24	数据通路信号调节全局预充电寄存器	DTUPREA	0x0017	RW
25	Write Leveling 控制寄存器	WL_CTRL	0x0018	RW
26	主存数据单错验证码寄存器	MEMSERR_P	0x0019	RW1C
27	配置错误寄存器	CfgErr	0x001a	RO
28	保留	—	0x001b	—
29	初始化状态寄存器	FSMSTAT	0x001c	RO
30	RDIMM 控制字配置寄存器	RC	0x001d	RW
31	RDIMM 控制字配置延时控制寄存器 0	RCMRD	0x001e	RW
32	RDIMM 控制字配置延时控制寄存器 1	6U	0x001f	RW
33	存储器地址/命令/数据接口控制寄存器	PUB_INTERFACE	0x0020	RW
34	功耗控制寄存器	POWERCTRL	0x0021	RW
35	保留	—	0x0022	—
36	存储器地址/命令/数据接口 DTO 引脚控制寄存器	PUB.DTO_CTL	0x0023	RW
37	保留	—	0x0024	—
38	SDRA 模式寄存器 0	CfgMR0	0x0025	RW

39	SDRA 模式寄存器 1	CfgMR1	0x0026	RW
40	SDRA 模式寄存器 2	CfgMR2	0x0027	RW
41	SDRA 模式寄存器 3	CfgMR3	0x0028	RW
42	性能检测控制寄存器	PERF_CTRL	0x0029	RW
43	性能检测寄存器 0	MC_PERF0	0x002a	RO
44	读主存数据单错寄存器	MEMSERR_CNT	0x002b	RW1C
45	单错阈值寄存器	MEMSERR_TH	0x002c	RW
46	写主存数据单错寄存器	WRDATASERR_CNT	0x002d	RW1C
47	性能检测寄存器 1	MC_PERF1	0x002e	RO
	刷新周期性能计数寄存器	RWCNTOFREF	0x002f	RO

## 1.2.5 存储器地址/命令/数据接口 IO 寄存器编址

存储器地址/命令/数据接口的 IO 寄存器编址如表 1-3 所示。

表 1-3: 存储器地址/命令/数据接口的 IO 寄存器编址

序号	名称	符号	索引地址 PA[22:7]	存取 特性
1	版本信息寄存器	RIDR	0x0000	RO
2	初始化寄存器	PIR	0x0001	RW
3~4	通用控制寄存器	PGCR0-1	0x0002~0x0003	RW
5~6	通用状态寄存器	PGSR0-1	0x0004~0x0005	RO
7	PLL 控制寄存器	PLLCR	0x0006	RW
8~12	延时参数寄存器	PTR0-4	0x0007~0x000B	RW
13	地址/命令主延时寄存器	ACMDLR	0x000C	RW
14	地址/命令位延时寄存器	ACBDLR	0x000D	RW
15	地址/命令 IO 配置寄存器	ACIOCR	0x000E	RW
16	数据通用配置寄存器	DXCCR	0x000F	RW
17	系统通用配置寄存器	DSGCR	0x0010	RW
18	SDRAM 配置寄存器	DCR	0x0011	RW
19~21	SDRAM 延时参数寄存器	DTPR0-2	0x0012~0x0014	RW
22	SDRAM 模式寄存器 0	MR0	0x0015	RW

23	SDRAM 模式寄存器 1	MR1	0x0016	RW
24	SDRAM 模式寄存器 2	MR2	0x0017	RW
25	SDRAM 模式寄存器 3	MR3	0x0018	RW
26	ODT 配置寄存器	ODTCR	0x,0019	RW
27	Data Training 配置寄存器	DTCR	0x001A	RW
28~31	Data Training 地址寄存器	DTAR0-3	0x001B~0x001E	RW
32~33	Data Training 数据寄存器	DTDR0-1	0x001F~0x0020	RW
34~35	Data Training 眼图数据寄存器	DTEDR0-1	0x0021~0x0022	RW
36~48	保留	—	0x0023~0x002F	—
49	SDRAM 命令单元地址寄存器	DCUAR	0x0030	RW
50	SDRAM 命令单元数据寄存器	DCUDR	0x0031	RW
51	SDRAM 命令单元运行寄存器	DCURR	0x0032	RW
52	SDRAM 命令单元循环寄存器	DCULR	0x0033	RW
53	SDRAM 命令单元通用配置寄存器	DCUGCR	0x0034	RW
54	SDRAM 命令单元时间延时寄存器	DCUTPR	0x0035	RW
55~56	SDRAM 命令单元状态寄存器	DCUSR0-1	0x0036~0x0037	RO
57~64	保留	—	0x0038~0x003F	—
65	BIST 运行寄存器	BISTRR	0x0040	RW
66	BIST 字计数寄存器	BISTWCR	0x0041	RW
67~69	BIST 屏蔽寄存器	BISTMSKR0-2	0x0042~0x0044	RW
70	BIST 随机测试种子寄存器	BISTLSR	0x0045	RW
71~73	BIST 地址寄存器	BISTAR0-2	0x0046~0x0048	RW
74	BIST 用户数据向量寄存器	BISTUDPR	0x0049	RW
75	BIST 通用状态寄存器	BISTGSR	0x004A	RO
76	BIST 字错误寄存器	BISTWER	0x004B	RO
77~80	BIST 位错误寄存器	BISTBER0-3	0x004C~0x004F	RO
81	BIST 字计数状态寄存器	BISTWCSR	0x0050	RO
82~84	BIST 失败错误字寄存器	BISTFWR0-2	0x0051~0x0053	RO
85~94	保留	—	0x0054~0x005D	—
95~96	保留	—	0x005E~0x005F	—
97~98	ZQ0 电阻控制寄存器	ZQ0CR0-1	0x0060~0x0061	RW
99~100	ZQ0 电阻状态寄存器	ZQ0SR0-1	0x0062~0x0063	RW
101~102	ZQ1 电阻控制寄存器	ZQ1CR0-1	0x0064~0x0065	RW

103~104	ZQ1 电阻状态寄存器	ZQ1SR0-1	0x0066~0x0067	RW
105~106	ZQ2 电阻控制寄存器	ZQ2CR0-1	0x0068~0x0069	RW
107~108	ZQ2 电阻状态寄存器	ZQ2SR0-1	0x006A~0x006B	RW
109~110	保留	—	0x006C~0x006D	—

111~112	保留	—	0x006E~0x006F	—
113~128	DATX0 通用配置寄存器	DX0GCR	0x0070	RW
	DATX0 通用状态寄存器	DX0GSR0-1	0x0071~0x0072	RW
	DATX0 位延时寄存器	DX0BDLR0-4	0x0073~0x0077	RW
	DATX0 本地校准延时寄存器	DX0LCDLR0-2	0x0078~0x007A	RW
	DATX0 主延时寄存器	DX0MDLR	0x007B	RW
	DATX0 通用延时寄存器	DX0GTR	0x007C	RW
	保留	—	0x007D~0x007F	—
129~144	DATX1 通用配置寄存器	DX1GCR	0x0080	RW
	DATX1 通用状态寄存器	DX1GSR0-1	0x0081~0x0082	RO
	DATX1 位延时寄存器	DX1BDLR0-4	0x0083~0x0087	RW
	DATX1 本地校准延时寄存器	DX1LCDLR0-2	0x0088~0x008A	RW
	DATX1 主延时寄存器	DX1MDLR	0x008B	RW
	DATX1 通用延时寄存器	DX1GTR	0x008C	RW
	保留	—	0x008D~0x008F	—
145~160	DATX2 通用配置寄存器	DX2GCR	0x0090	RW
	DATX2 通用状态寄存器	DX2GSR0-1	0x0091~0x0092	RO
	DATX2 位延时寄存器	DX2BDLR0-4	0x0093~0x0097	RW
	DATX2 本地校准延时寄存器	DX2LCDLR0-2	0x0098~0x009A	RW
	DATX2 主延时寄存器	DX2MDLR	0x009B	RW
	DATX2 通用延时寄存器	DX2GTR	0x009C	RW
	保留	—	0x009D~0x009F	—
161~176	DATX3 通用配置寄存器	DX3GCR	0x00A0	RW
	DATX3 通用状态寄存器	DX3GSR0-1	0x00A1~0x00A2	RO
	DATX3 位延时寄存器	DX3BDLR0-4	0x00A3~0x00A7	RW
	DATX3 本地校准延时寄存器	DX3LCDLR0-2	0x00A8~0x00AA	RW
	DATX3 主延时寄存器	DX3MDLR	0x00AB	RW
	DATX3 通用延时寄存器	DX3GTR	0x00AC	RW
	保留	—	0x00AD~0x00AF	—



177~192	DATX4 通用配置寄存器	DX4GCR	0x00B0	RW
	DATX4 通用状态寄存器	DX4GSR0-1	0x00B1~0x00B2	RO
	DATX4 位延时寄存器	DX4BDLR0-4	0x00B3~0x00B7	RW
	DATX4 本地校准延时寄存器	DX4LCDLR0-2	0x00B8~0x00BA	RW
	DATX4 主延时寄存器	DX4MDLR	0x00BB	RW
193~208	DATX4 通用延时寄存器	DX4GTR	0x00BC	RW
	保留	—	0x00BD~0x00BF	—
	DATX5 通用配置寄存器	DX5GCR	0x00C0	RW
	DATX5 通用状态寄存器	DX5GSR0-1	0x00C1~0x00C2	RO
	DATX5 位延时寄存器	DX5BDLR0-4	0x00C3~0x00C7	RW
	DATX5 本地校准延时寄存器	DX5LCDLR0-2	0x00C8~0x00CA	RW
	DATX5 主延时寄存器	DX5MDLR	0x00CB	RW
209~224	DATX5 通用延时寄存器	DX5GTR	0x00CC	RW
	保留	—	0x00CD~0x00CF	—
	DATX6 通用配置寄存器	DX6GCR	0x00D0	RW
	DATX6 通用状态寄存器	DX6GSR0-1	0x00D1~0x00D2	RO
	DATX6 位延时寄存器	DX6BDLR0-4	0x00D3~0x00D7	RW
	DATX6 本地校准延时寄存器	DX6LCDLR0-2	0x00D8~0x00DA	RW
	DATX6 主延时寄存器	DX6MDLR	0x00DB	RW
225~240	DATX6 通用延时寄存器	DX6GTR	0x00DC	RW
	保留	—	0x00DD~0x00DF	—
	DATX7 通用配置寄存器	DX7GCR	0x00E0	RW
	DATX7 通用状态寄存器	DX7GSR0-1	0x00E1~0x00E2	RO
	DATX7 位延时寄存器	DX7BDLR0-4	0x00E3~0x00E7	RW
	DATX7 本地校准延时寄存器	DX7LCDLR0-2	0x00E8~0x00EA	RW
	DATX7 主延时寄存器	DX7MDLR	0x00EB	RW
241~256	DATX7 通用延时寄存器	DX7GTR	0x00EC	RW
	保留	—	0x00ED~0x00EF	—
	DATX8 通用配置寄存器	DX8GCR	0x00F0	RW
	DATX8 通用状态寄存器	DX8GSR0-1	0x00F1~0x00F2	RO
	DATX8 位延时寄存器	DX8BDLR0-4	0x00F3~0x00F7	RW
241~256	DATX8 本地校准延时寄存器	DX8LCDLR0-2	0x00F8~0x00FA	RW
	DATX8 主延时寄存器	DX8MDLR	0x00FB	RW

	DATX8 通用延时寄存器	DX8GTR	0x00FC	RW
	保留	—	0x00FD~0x00FF	—

## 1.2.6 DDR3PHY\_FPGA 寄存器编址

表 1-4: DDR3PHY\_FPGA 空间划分

序号	名称	符号	索引地址 【22: 7】	存取特性
1~3	写延时调整寄存器	WDELAY_SEL	0x0023~ 0x0025	RW
4~6	读延时调整寄存器	RDELAY_SEL	0x0026~ 0x0028	RW
7	读响应延时控制寄存器	RESEDELAY_SEL	0x0029	RW
8	卸载控制寄存器	DOWNLOAD_SEL	0x002a	RW
9~11	读 Gate 延时调整寄存器	QSENDELAY_SEL	0x002b~ 0x002d	RW
12	CfgCR	CfgCR	0x002e	RW
	保留	保留	0x002f	保留
13	CORE_ENABLE	CORE_ENABLE	0x0038	RW
14	PUB0 Update 控制寄存器	PUB0_Update	0x0039	RW
15	PUB1 Update 控制寄存器	PUB1_Update	0x003a	RW
16	MR0	MR0	0x0015	RW
17	MR1	MR1	0x0016	RW
18	MR2	MR2	0x0017	RW
19	MR3	MR3	0x0018	RW
20~21	读 Gate 宽度调整寄存器	QSENWIDTH_SEL	0x003b~ 0x003c	RW
24~22	写 Gate 延时调整寄存器	DSENDELAY_SEL	0x003d~ 0x003f	RW
26~25	写 Gate 宽度调整寄存器	DSENWIDTH_SEL	0x0054~ 0x0055	RW
27~30	Write Leveling 调整寄存器	WL_SEL	0x0056~ 0x0059	RW

说明:

- 1) DDR3PHY FPGA 采用 1 个 AC 和 18 个 DATX8 的方式;
- 2) 内部设置的 IOR 采用 PUB0 的 CFG 接口进行访问, MR0~MR3 的编址与 PUB0 相同, 其他

IOR

编址是利用 PUB 中的保留地址；具体见 FPGA IOR 说明；

## 1.2.7 1.2.3 系统接口 IO 寄存器编址

系统接口部件（IPU）内的 IO 寄存器编址如表 1-4 所示。

表 1-4：系统接口部件内的 IO 寄存器编址

序号	名称	符号	索引地址 PA[35:7]	存取 特性
1	保留	—	0x0000,0000	—
2	核心睡眠状态寄存器	CORE_SLEEP_STAT	0x0000,0001	RO
3	PCI-E 和维护中断丢失寄存器	DEVINT_MIS	0x0000,0002	RW
4	IPU 超时控制寄存器	IPU_CTRL	0x0000,0003	RW
5	短时钟中断计数起始值寄存器	SHORT_CLK_START	0x0000,0004	RW
6	短时钟中断屏蔽寄存器	SHORT_CLK_MSK	0x0000,0005	RW
7	维护中断 MAP 表	MT_INDX	0x0000,0006	RW
8	短时钟中断 MAP 表	SHORT_INDX	0x0000,0007	RW
9	核间中断使能	INTEN	0x0000,0008	RW
10	核心 IO 访问保序控制	LOCKEN	0x0000,0009	RW
11~14	保留	—	0x0000,000a~ 0x0000,000d	—
15	包格式错误寄存器	FMT_ERR	0x0000,000e	RW
16	系统故障 MAP 表	SYSERR_INDX	0x0000,000f	RW
17	IPU 状态寄存器	IPU_STAT	0x0000,0010	RW
18	IPU 单错预警阈值寄存器	SERR_CNTTH	0x0000,0011	RW
19	IPU 接收 MCU 方向请求或响应包 包头 ECC 单错次数统计寄存器	MCUSERR_CNT	0x0000,0012	RW
20	IPU 接收 PIU0 方向请求或响应包 包头 ECC 单错次数统计寄存器	PIU0SERR_CNT	0x0000,0013	RW
21	IPU 接收 PIU1 方向请求或响应包 包头 ECC 单错次数统计寄存器	PIU1SERR_CNT	0x0000,0014	RW
22	IPU 接收 CPM 方向请求或响应包 包头 ECC 单错次数统计寄存器	FXSERR_CNT	0x0000,0015	RW

23	IPU 报错使能寄存器	ERRRPT_EN	0x0000,0016	RW
24	故障中断目标配置寄存器	SYSERRINT_DST	0x0000,0017	RW

25	IPU 状态报错使能寄存器	IPU_STAT_ERREN	0x0000,0018	RW
26~29	PCI-E0 INTx 中断 MAP 表	PCIE0_INTx_INDX	0x0000,0019~ 0x0000,001c	RW
30~33	PCI-E1 INTx 中断 MAP 表	PCIE1_INTx_INDX	0x0000,001d~ 0x0000,0020	RW
34~35	保留	—	0x0000,0021~ 0x0000,0022	—
36	核间中断丢失状态寄存器	IINT_MIS	0x0000,0023	RW
37	核间中断丢失报错使能寄存器	IINT_MIS_RPTEN	0x0000,0024	RW
38	PCI-E 和维护中断丢失寄存器	DEVINT_MIS_RPTE	0x0000,0025	RW
39	ECC 单错状态寄存器	ECCSERR	0x0000,0026	RW
40	ECC 单错报错使能寄存器	ECCSERR_RPTEN	0x0000,0027	RW
41	ECC 多错状态寄存器	ECCMERR	0x0000,0028	RW
42	ECC 多错报错使能寄存器	ECCMERR_RPTEN	0x0000,0029	RW
43	设备中断唤醒使能寄存器	DEVINT_WKEN	0x0000,002a	RW
44	不可屏蔽中断 NMI MAP 表	NMI_INDX	0x0000,002b	RW
45 ~ 128	保留	—	0x0000,002c~ 0x0000,007f	—
129 ~ 130	异步消息队列	MAIL_BOX_i	0x0000,0080~ 0x0000,0083 (这 4 个地址由 低到高对应目标 核心 0~3; 当某 个核心断连时, 对应的地址为非 法地址)	RW
131 ~ 144	保留	—	0x0000,0084~ 0x0000,008f	—
145 ~ 146	异步消息队列状态寄存器	MAIL_STAT_i	0x0000,0090~ 0x0000,0093 (这 4 个地址由	RW

			低到高对应目标核心 0~3；当某个核心断连时，对应的地址为非 法地址)	
147~ 256	保留	—	0x0000,0094~ 0x0000,00ff	—
257~ 512	PCI-E0 MSI 中断 MAP 表	PCIE0_MSI[i]_IND X	0x0000,0100~ 0x0000,01ff	RW
513~ 768	PCI-E1 MSI 中断 MAP 表	PCIE1_MSI[i]_IND X	0x0000,0200~ 0x0000,02ff	RW

## 1.2.8 维护接口 IO 寄存器编址

维护接口部件（MCU）内的 IO 寄存器编址如表 1-5 所示。

表 1-5：维护接口内的 IO 寄存器编址

序号	名称	符号	索引地址 PA[35:7]	存取特 性
1	保留	—	0x0000,0000	—
2	核组总错状态寄存器	CG_FAULT_STAT	0x0000,0001	RW
3~5	保留	—	0x0000,0002~ 0x0000,0004	—
6	核组总错状态报错使能寄存器	CG_FAULT_EN	0x0000,0005	RW
7~9	保留	—	0x0000,0006~ 0x0000,0008	—
10~13	核组故障中断报错使能寄存器	CG_FAULT_INT_EN	0x0000,0009	RW
11~13	保留	—	0x0000,000a~ 0x0000,000c	
14	初始化控制寄存器	INIT_CTL	0x0000,000d	RW
15	芯片状态寄存器	MT_STATE	0x0000,000e	RO
16	核心在位使能寄存器	CORE_ONLINE	0x0000,000f	RW
17	维护中断寄存器	MT_INT	0x0000,00010	WO
18	维护中断结束寄存器	MT_INT_END	0x0000,00011	RW

19	CPU 号寄存器	CPUID	0x0000,00012	RW
20~23	核心软件信息寄存器	SOFT_INF0~3	0x0000,0013~ 0x0000,0016	RW
24~35	保留	—	0x0000,0017~ 0x0000,0022	—
36	长时钟计数寄存器	LONG_TIME	0x0000,0023	RW
37	长时钟计数间隔寄存器	LONG_PRESET	0x0000,0024	RW
38	测试选择控制寄存器	TESTSEL	0x0000,0025	RW
39	外围接口使能寄存器	IO_START	0x0000,0026	RW
40	PCI-E0 接口 PERST_N 复位寄存器	PERST_N_PCIE0	0x0000,0027	RW
41	PCI-E0 接口 BUTTON_RST_N 复位寄存器	BUTTON_RST_N_PCIE0	0x0000,0028	RW
42	BIST 测试控制寄存器	BIST_GOON	0x0000,0029	WO
43	维护响应不匹配计数器	MTRSPMISS	0x0000,002a	RO
44~47	中断睡眠完成寄存器	SLEEP_DONE0~3	0x0000,002b~ 0x0000,002e	WO
48~59	保留	—	0x0000,002f~ 0x0000,003a	—
60	时钟切换时间间隔	PLL_CHG_CNT	0x0000,003b	RW
61	核组 0 BIST 结果信息	CG0_BIST_STAT	0x0000,003c	RO
62~64	保留	—	0x0000,003d~ 0x0000,003f	—
65	芯片 BIST 结果信息寄存器	BIST_STAT	0x0000,0040	RO
66	JTAG TAP 控制选择寄存器	TAPSEL	0x0000,0041	RO
67	PCI-E1 接口 PERST_N 复位寄存器	PERST_N_PCIE1	0x0000,0042	RW
68	PCI-E1 接口 BUTTON_RST_N 复位寄存器	BUTTON_RST_N_PCIE1	0x0000,0043	RW
69	存储器自测试控制寄存器	BIST_CTL	0x0000,0044	RW
70	存储器自测试超时阈值寄存器	BIST_OVERTIME_TH	0x0000,0045	RW
71	状态机控制寄存器	ST_NEXT_FLAG	0x0000,0046	WO
72	维护操作屏蔽寄存器	OP_MASK	0x0000,0047	RW



73	MCU 单错预警阈值寄存器	SERR_CNTTH	0x0000,0048	RW
74	MCU 接收系统接口方向请求或响应单错次数统计寄存器	IPUSERR_CNT	0x0000,0049	RW
75~76	MCU 接收 PCI-E 方向响应单错次数统计寄存器	PIU0SERR_CNT ~	0x0000,004a ~ 0x0000,004b	RW
77	MCU 报错使能寄存器	ERRRPT_EN	0x0000,004c	RW
78	MCU 错误信息寄存器	ERR_INF	0x0000,004d	RW
79	保留	—	0x0000,004e	—
80	SCAN 间隔控制寄存器	SCAN_WAIT_CFG	0x0000,004f	RW
81	时钟测试寄存器	CLU_TEST	0x0000,0050	RW
82	PCACHE BIST 结果寄存器	PC_BIST_STAT	0x0000,0051	RO
83~86	核心错误信息清除寄存器	Core0_ERRRST_N~ Core3_ERRRST_N	0x0000,0052~ 0x0000,0055	RW
87~98	保留	—	0x0000,0056~ 0x0000,0061	—
99	系统接口总错状态寄存器	SI_FAULT_STAT	0x0000,0062	RW
100	系统接口总错状态报错使能寄存器	SI_FAULT_EN	0x0000,0063	RW
101	系统接口故障中断报错使能寄存器	SI_FAULT_INT_EN	0x0000,0064	RW
102~103	PCI-E 时钟选择寄存器	CLK_SEL_PCIE0~1	0x0000,0065~ 0x0000,0066	RW
104	保留	—	0x0000,0067	—
105	异步 FIFO 读写指针同步的级数配置寄存器	FIFO_SYNSEL	0x0000,0068	RW
106	CPU 信息寄存器	CPUINFO	0x0000,0069	RW
107	外部唤醒中断控制寄存器	WAKEUP_CTL	0x0000,006a	RW
108	标志寄存器	FLAGREG	0x0000,006b	RW
109	外部故障中断控制寄存器	NMI_CTL	0x0000,006c	RW
110	PCI-E PHY PLL 等待计数器	PIUPLL_CNT	0x0000,006d	RW
111	保留	—	0x0000,006e	—
112	核组在位使能寄存器	MC_ONLINE	0x0000,006f	RW

## 1.2.9 PCI-E 接口的 IO 空间划分与寄存器编址

申威 411 处理器共集成了两个 PCI-E 接口，这两个接口的 IO 地址空间以 PA[39:36]进行区分，PCI-E0 接口的 IO 空间对应 PA[39:36]为 0xC，PCI-E1 接口的 IO 空间对应 PA[39:36]为 0xF。本档 中 PCI-E 接口中寄存器编址以 PCI-E0 接口为例，PCI-E1 接口类似，只需将 PA[39:36]由 0xC 改为 0xF。每个 PCI-E 接口内部，再按照 PA[35:0]可将该空间划分为以下 6 个子空间：

- 1) PCI-E 终端设备 (EP, End Point) 的 32 位存储器空间；
- 2) PCI-E 终端设备的 64 位存储器空间；
- 3) PCI-E 接口控制部件 (PIU, PCI-E Interface Unit) 的 IO 空间；
- 4) PCI-E 终端设备的 IO 空间；
- 5) PCI-E 根部件 (RC, Root Complex) 的配置空间；
- 6) PCI-E 终端设备的配置空间。

其中 PIU 和 RC 集成在申威 411 处理器内部，EP 是通过 PCI-E 链路与 RC 相连的外部设备。具体划分如表 1-6 所示。

表 1-6: PCI-E 接口内部 IO 空间划分

PA[35:34]	其它	
2'b00	PA[33:32]为“0”	EP 的 32 位存储器空间。
	PA[33:32]为“1”、“2”、“3”	EP 的 64 位存储器空间。
2'b01	PA[33]为“0”	PIU 内部 IO 空间。该空间又划分成两个部分——PA [32] 为 0 为 IOR0 地址空间；PA [32]为 1 为 IOR1 地址空间 寄存器索引地址为 PA[13:7]，PA[6:0]为全 0。 PIU 内部 IO 空间按 1024 位编址（128B 对界）。
	PA[33]为“1”	EP 的 IO 空间。PA[32]取值被忽略，因此有效空间只有 4GB。EP IO 空间按 32 位编址（4B 对界）。
2'b10	PA[33]为“0”	RC 配置空间。寄存器索引地址为 PA[23:7]，内部划分如下：  PA[23:21]：功能号； PA[16:7]：寄存器号； PA[6:0]：全 0； 其它：保留。  RC 配置空间按 1024 位编址（128B 对界）。

	PA[33]为“1”	EP 配置空间。内部地址编址为： PA[32]：保留； PA[31:24]：总线号； PA[23:19]：设备号； PA[18:16]：功能号； PA[15:12]：保留； PA[11:8]：扩展寄存器号； PA[7:2]：寄存器号。  EP 配置空间按 32 位编址（4B 对界）。
--	------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------

说明：

- 1) EP 的存储器空间划分在连续的 16GB 空间范围内；
- 2) 当访问 EP 的配置空间时，用 PA[32]来区分配置请求采用 TYPE0 还是 TYPE1 类型的包头

（PA[32]为 0 表示 TYPE0 类型配置访问，为 1 表示 TYPE1 类型）。TYPE0 用于“非桥设备”的配置请求，TYPE1 用于“交叉开关”或“桥设备”的配置请求。

### 1.2.9.1 PCI-E 设备的 IO 寄存器编址

对于 EP 的地址空间，共有以下 5 类子空间，其地址范围（包括申威 411 处理器编址和 PCI-E 标准编址的对应关系）、空间大小如表 1-7 所示。

表 1-7：EP 的地址空间划分

EP 地址空间类型	申威 411 处理器编址范围	PCI-E 标准编址范围
32 位存储器空间	0xC0,0000,0000~0xC0,FFFF,FFFF (4GB)	0x0,0000,0000~0x0,FFFF,FFFF
64 位存储器空间	0xC1,0000,0000~0xC3,FFFF,FFFF (12GB)	0x1,0000,0000~0x3,FFFF,FFFF
IO 空间	0xC6,0000,0000~0xC7,FFFF,FFFF (4GB)	0x0000,0000~0xFFFF,FFFF
TYPE0 类型配置空间	0xCA,0000,0000~0xCA,FFFF,FFFF (4GB)	0x0000,0000~0xFFFF,FFFF
TYPE1 类型配置空间	0xCB,0000,0000~0xCB,FFFF,FFFF (4GB)	0x0000,0000~0xFFFF,FFFF

说明：

- 1) EP 的 32 位存储器空间共 4GB 大小，发给 EP 的有效地址截取 PA[31:0]即可；
- 2) EP 的 64 位存储器空间共 12GB 大小，发给 EP 的有效地址截取 PA[33:0]即可；
- 3) EP 的 IO 空间范围为 8GB，但由于其最大只能扩展到 4GB（受 PCI-E 请求数据包中地址域的位宽所限），因此发给 EP 的有效地址只需截取 PA[31:0]即可，PA[32]被忽略；
- 4) 申威 411 处理器编址和 PCI-E 标准编址的转换由 PIU 实现。

### 1.2.9.2 PCI-E 接口控制器的 IO 寄存器编址

PIU 和 RC 内部的寄存器都是按 1024 位编址的。

其中，PIU 的寄存器控制又划分为 IOR0、IOR1 两部分。IOR0 部分寄存器的访问基址为 0xC4,0000,0000，其编址如表 1-8 所示；IOR1 部分寄存器的访问基址为 0xC5,0000,0000，其编址如表 1-9 所示。

表 1-8: PIU 内部 IOR0 部分寄存器编址

序号	名称	符号	索引地址 [15:7]	存取 特性
1	PIU 配置寄存器 0	PIUConfig0	0x00	RWS
2	EP DMA 请求基址寄存器	EPDMABAR	0x01	RW
3~ 66	IOMMU 段式代换段表寄存器 0~63	IOMMUSegItem~63	0x02~0x41 (由低到高对 应寄存器 0~63)	RW
67	IOMMU 地址代换异常控制寄存器	IOMMUExcpt_Ctrl	0x42	RW
68	MSI 地址寄存器	MSIAddr	0x43	RW
69~ 324	MSI 中断配置寄存器 0~255	MSIConfig0~255	0x44~0x143 (由低到高对 应寄存器 0~255)	RW
325	INTA 中断配置寄存器	INTAConfig	0x144	RW
326	INTB 中断配置寄存器	INTBConfig	0x145	RW
327	INTC 中断配置寄存器	INTCConfig	0x146	RW
328	INTD 中断配置寄存器	INTDConfig	0x147	RW
329	高级错误报告错 INT 中断配置寄存器	aerErrIntConfig	0x148	RW
330	高级错误报告错 MSI 中断配置寄存器	aerErrMsiConfig	0x149	RW
331	PME INT 中断配置寄存器	pmeIntConfig	0x14a	RW
332	PME MSI 中断配置寄存器	pmeMsiConfig	0x14b	RW
333	热插拔 INT 中断配置寄存器	hpIntConfig	0x14c	RW
334	热插拔 MSI 中断配置寄存器	hpMsiConfig	0x14d	RW
335	链路自动带宽调节 INT 中断配置寄存器	linkAutoBwIntConfig	0x14e	RW
336	链路带宽管理 INT 中断配置寄存器	bwMgtIntConfig	0x14f	RW

337	DMAW 请求/响应计数寄存器	DMAWReqCplCnt	0x150	RW
338	DMAR 请求/响应计数寄存器	DMARReqCplCnt	0x151	RW
339	MSI 中断请求/处理计数寄存器	MSIReqClrCnt	0x152	RW
340	INTX 中断请求/处理计数寄存器	INTXReqClrCnt	0x153	RW
341	页式代换访存请求/响应计数寄存器	PageMemReqCplCnt	0x154	RW
342~ 352	保留	---	0x155~0x15f	---
353	页式代换设备表基址寄存器	DTBaseAddr	0x160	RW
354	设备表 TLB 全刷新寄存器	DTLB_FlushAll	0x161	RW
355	指定设备号设备 TLB 表刷新寄存器	DTLB_FlushDev	0x162	RW
356	页表 TLB 全刷新寄存器	PTLB_FlushAll	0x163	RW
357	指定设备号页表 TLB 刷新寄存器	PTLB_FlushDev	0x164	RW
358	指定设备和虚地址页表 TLB 刷新寄存器	PTLB_FlushVAddr	0x165	RW
359	页表 Cache 全刷新寄存器	PCache_FlushAll	0x166	RW
360	指定设备和索引页表 Cache 刷新寄存器	PCache_FlushDev	0x167	RW
361	指定物理地址页表 Cache 刷新寄存器	PCache_FlushPAddr	0x168	RW
362	超时配置寄存器	Timeout_Config	0x169	RW
363	IOMMU 地址代换异常状态寄存器	IOMMUExcpt_Status	0x16a	RW
364	IOMMU 页式地址代换地址寄存器 1	IOMMUPage_PAddr1	0x16b	RO
365	IOMMU 页式地址代换地址寄存器 2	IOMMUPage_PAddr2	0x16c	RO
366	IOMMU 页式地址代换地址寄存器 3	IOMMUPage_PAddr3	0x16d	RO

表 1-9: PIU 内部 IOR1 部分寄存器编址

序号	名称	符号	索引地址 [15:7]	存取 特性
1	PIU 配置寄存器 1	PIUConfig1	0x00	RWS
2	PIO 请求/响应计数器	PIOReqCplCnt	0x01	RW
3	RC 请求/响应计数器	RCReqCplCnt	0x02	RW
4	设备 NP 类请求/响应计数器	EPNPREqCplCnt	0x03	RW
5	IOR0 请求/响应计数器	IOR0ReqCplCnt	0x04	RW
6	IOR1 请求/响应计数器	IOR1ReqCplCnt	0x05	RW
7~ 11	LTSSM 链路训练最近状态寄存器 0~4	NewLTSSMState0~4	0x06~0xa	RO

12~16	LTSSM 链路训练最旧状态寄存器 0~4	OldLTSSMState0~4	0x0b~0xf	RO
17	错误登记寄存器	ErrReport	0x10	RW
18	错误使能寄存器	ErrEnable	0x11	RW
19	DMA 响应 ECC 单错计数器	DMACpl_SErrCnt	0x12	RWC
20	PIO 请求 ECC 单错计数器	PIOREq_SErrCnt	0x13	RWC
21	MCU 请求 ECC 单错计数器	MCUReq_SErrCnt	0x14	RWC
22	ECC 单错计数阈值寄存器	ECCSErrLimit	0x15	RW
23	RC 控制寄存器	RCCControl	0x16	RW
24	RC 电源管理寄存器	RCPowerManage	0x17	RW
25	RC 调试信息寄存器 0	RCDebugInf0	0x18	RO
26	RC 调试信息寄存器 1	RCDebugInf1	0x19	RO
27	RC 电气特性控制寄存器	RCElectroControl	0x1a	RW
28	RC 电气特性状态寄存器	RCElectroStatus	0x1b	RO
29	RC 复位寄存器	RCHotRst	0x1c	RW
30	MSG 控制寄存器 0	MSGControl0	0x1d	RW
31	MSG 控制寄存器 1	MSGControl1	0x1e	RW
32	链路复位延迟控制寄存器	LinkDownDelay	0x1f	RW
33	PHY 控制寄存器	PMAControl	0x20	RWS
34	页表 Cache 数据单错计数寄存器	PCData_SErrCounter	0x21	RWC
35	页表 Cache TAG 偶校验错计数寄存器	PCTag_PErrCounter	0x22	RWC

### 1.2.9.3 PCI-E 根部件的 IO 寄存器编址

RC 内部的 IO 寄存器的访问基址为 0xC8,0000,0000，其编址及访问方式详见 5.2 节。该节描述的 RC 内部寄存器的偏移以字节为单位且 4 字节对齐，以便与 PCI-E 规范保持一致。

RC 内部 IO 寄存器的偏移与软件访问地址偏移量的关系为：寄存器偏移值左移 5 位即得到访问地址中的偏移量。如偏移为 0x38 的寄存器，其软件访问地址的偏移量为 0x700，访问地址为 0xC8,0000,0700。

## 1.3 IO 寄存器的访问机制

IO 寄存器编址在申威 411 处理器的 IO 空间上，软件可以通过 IO 空间的 Load、Store 指令显式

访问 IO 寄存器。维护接口也可以通过维护命令显式访问 IO 寄存器。对维护接口内的 IO 寄存器而言，维护接口的权限通常比软件高，其它 IO 寄存器的属性则对维护接口和软件访问权限相同。

申威 411 处理器保证每个核心对 IO 空间的读写操作不能推测进行，这就保证了软件显式读写 IO 寄存器之间的顺序。维护命令只能串行执行，这也保证了维护接口显式读写 IO 寄存器之间的顺序。申威 411 处理器不保证软件对 IO 寄存器的显式读写与维护接口对 IO 寄存器的显式读写之间的顺序，需要软件和系统人员在高层进行控制。

申威 411 处理器不保证显式写 IO 寄存器和隐式读 IO 寄存器之间的顺序关系，需要软件通过同步方式保证显式写完成之后，IO 寄存器的新值能对后续请求的处理产生影响。如对存控的配置寄存器进行写操作前，首先软件必须保证没有访存操作正在执行，显示写执行完毕后，才能开始后续的访存操作，如果软件无法同步 16 个核心和 PCI-E0、PCI-E1、维护接口的访存行为，则对相关 IO 寄存器的配置只能在系统引导（Boot）之前通过维护接口或者初始化加载到 ICache 中的程序进行配置，配置完成后不允许再修改。

## 2 存储器接口的 IO 寄存器

### 2.1 一致性处理部件的 IO 寄存器

#### 2.1.1 CPM\_MC0Disable / CPM\_MC1Disable

CPM\_MC0Disable 和 CPM\_MC1Disable 是一致性处理部件的 MC 不使能配置寄存器，维护和核心可读写，复位为默认配置值。

表 2-1: CPM\_MC\*Disable 寄存器域的描述

名称	范围	类型	描述
MCDisable	[0]	WR,0	禁止存储器访问标志，为“1”表示不允许接收访问主存的请求，如果接收到访问主存的请求，存控将直接返回 ReadDataError，初始值为“0”。该标志不限制对 MC 的 IO 访问。 要求软件在 SROM 加载程序一开始，就将该位写为“1”，等存控进行 Data training 结束后，再将该位写为“0”。
—	其它	—	保留。

#### 2.1.2 CPM\_CrossMode

CPM\_CrossMode 是一致性处理部件的双 MC 地址交叉模式配置寄存器，维护和核心可读写，复位为默认配置值。单个 MC 工作时，该寄存器不起作用，CPM 固定将按“最高位地址交叉”模式处理请求。

表 2-2: CPM\_CrossMode 寄存器域的描述

名称	范围	类型	描述
----	----	----	----



CrossMode	[1:0]	WR,0	两个 MC 的地址交叉模式，分为 4 种： <ul style="list-style-type: none"> <li>• 0：按奇/偶 Cache 块(Addr[7])划分两个 MC；（默认值）</li> <li>• 1：按奇/偶页号(Addr[13])划分两个 MC；</li> <li>• 2：按 1MB 地址(Addr[20])划分两个 MC；</li> <li>• 3：按最高地址划分两个 MC，即两个 MC 连续编址。该位地址与 MC#0 的容量有关，1GB/2GB/4GB/8GB 时分别为 Addr[30]/Addr[31]/Addr[32]/Addr[33]；</li> </ul>
—	其它	—	保留。

### 2.1.3 CPM\_FRQCmprEn

CPM\_FRQCmprEn 是一致性处理部件的访存一次请求队列（FRQ）压缩使能配置寄存器，维护和核心可读写，复位为默认配置值。

表 2-4：CPM\_FRQCmprEn 寄存器域的描述

名称	范围	类型	描述
FRQCmprEn	[0]	WR,1	<ul style="list-style-type: none"> <li>• '0'：不允许对 FRQ 进行队列压缩；</li> <li>• '1'：允许对 FRQ 进行有限的队列压缩；（默认值）</li> </ul>
—	其它	—	保留。

### 2.1.4 CPM\_PerfCtrl

CPM\_PerfCtrl 是一致性处理部件中多个性能计数器的控制寄存器，维护和核心可读写，复位为默认配置值。

表 2-5：CPM\_PerfCtrl 寄存器域的描述

名称	范围	类型	描述
PerfCtrl	[1:0]	WR,2	根据写入值控制 CPM 各个性能计数器的操作： <ul style="list-style-type: none"> <li>• 0：清除性能计数器；</li> <li>• 1：启动性能计数器；（默认值）</li> <li>• 2：停止性能计数器；</li> <li>• 其他值：停止性能计数器</li> </ul>
—	其它	—	保留。

CPM 性能计数器的使用流程：

- 1) 写“性能计数器控制寄存器”为 0，清除性能计数器。如果刚复位结束，可省略此步骤；
- 2) 写“性能计数器控制寄存器”为 1，启动性能计数器。如果刚复位结束，可省略此步骤；
- 3) 写“性能计数器控制寄存器”为 2，停止性能计数。

## 2.1.5 CPM\_CoreSleep

CPM\_CoreSleep 是一致性处理部件中用于核心睡眠功能的核心睡眠使能寄存器，维护和核心可读写，复位为默认配置值。核心睡眠之前，需要置该寄存器有效。核心和维护对该寄存器的读操作会返回不同的值。

表 2-5: CPM\_CoreSleep 寄存器域的描述

名称	范围	类型	描述
CoreSleep	[0]	WR,0	某个核心写该寄存器（任意值）时，CPM 将相应核心的“核心睡眠使能标志”置为‘1’，CPM 停止向该核心发送二次请求； 注意：硬件对该位进行自清除。由硬件在核心唤醒时隐式清除的，核心被唤醒后，软件 <b>不需要也“不能”</b> 向该寄存器写“0”，否则反而会将该寄存器被置‘1’，导致一致性处理出错。 某个核心读该寄存器时，CPM 将返回该核心的“核心睡眠使能标志”（64’h0 或 64’h1）； 维护对该寄存器的 IO 写操作不起作用；维护对该寄存器的 IO 读操作将返回 4 个核心的“核心睡眠使能标志”（读数据[3:0]分别对应 4 个核心）。
—	其它	—	保留。

## 2.1.6 CPM\_ProbeOK

CPM\_ProbeOK 是一致性处理部件中用于核心睡眠功能的二次请求完成状态查询寄存器，维护和核心只读。核心和维护对该寄存器的读操作会返回不同的值。

表 2-6: CPM\_ProbeOK 寄存器域的描述

名称	范围	类型	描述
ProbeOK	[0]	RO	记录某个核心的二次请求是否执行完毕，为‘1’表示 CPM 发向该核心的二次请求已经完成；某个核心读该寄存器时，CPM 将返回该核心的“CPM 外部 Probe 完成标志”（64’h0 或 64’h1）；维护读该寄存器时，CPM 将返回 4 个核心的“CPM 外部 Probe 完成标志”（读数据[3:0]分别对应 4 个核心）。
—	其它	—	保留。

## 2.1.7 CPM\_SerrThres

CPM\_SerrTh 是单错计数阈值。

表 2-7: CPM\_SerrThres 寄存器域的描述

名称	范围	类型	描述
SerrThres	[15:0]	RW,0xffff	CPM 单错预警阈值。如果单错统计计数器的计数值 大于该阈值，则 CPM 向 MCU 报 SERR 错误。
—	其它	—	保留。

## 2.1.8 CPM\_SerrCnt

CPM\_SerrCnt 是 CPM 内部检测到的各种单错的累计计数值。

表 2-8: CPM\_SerrCnt 寄存器域的描述

名称	范围	类型	描述
TData_SerrCnt	[15:0]	RWC,0	对 TData 阵列的输出数据进行 ECC 单错计数，到达最大值时停止计数。软件写该寄存器时清 0。
CPM_SerrCnt	[47:32]	RWC,0	对 CPM 内部检测到的 ECC 单错进行计数，不含 TData 阵列的 ECC 单错。到达最大值时停止计数。软件写该寄存器时清 0。
—	其它	—	保留。

## 2.1.9 CPM\_Fault

CPM\_Fault 是一致性处理部件故障状态寄存器，包括 CPM 发现的 ECC 单错、不可纠错、硬件 错误。该寄存器按位写'1'清'0'，写'0'保持相应位保持不变。

表 2-9: CPM\_Fault 寄存器域的描述

名称	范围	类型	描述
	[58]	RW1C,0	CPM-IPU 接口响应目标核心号错；（核心断连时，IPU 仍向核心返回响应）
	[57]	RW1C,0	EvictECacheBlk 刷新请求路号错；（路号超过芯片配置的 TCache 路数）。CPM 不向 MCU 报错，只作为查询状 态。(EvictECacheBlk 请求可能是推测路径上的)
CPM 硬件故障			

CPM_Fault	[56:38]	RW1C, 0	<p>[56]: IOQ 收到无数据回答; (IOQ 只会向核心发送“取数释放缓冲”请求)</p> <p>[55]: IPU-&gt;CPM 响应接口协议错 (响应包携带的头尾标志有错);</p> <p>[54]: IPU-&gt;CPM 请求接口协议错 (请求包携带的头尾标志有错);</p> <p>[53]: CPM-IPU 接口 RQ2 队列溢出;</p> <p>[52]: TMAF2 满溢出; [51]: TMAF1 满溢出; [50]: 二次请求队列 (SRQ) 溢出; [49]: TData 端口发生读写冲突; [48]: TTag 端口发生读写冲突;</p> <p>[47]: 二次请求目标错; (向断连核心发送非置无效类二次请求)</p> <p>[46]: 请求命中多个 TTag 条目;</p> <p>[45]: 请求命中 TTag 状态错 (脏独占 Local 核心向 CPM 发送指令流读请求或可 Cache 读写请求); [44]: TCache 淘汰请求地址非法 (含 EvictECacheBlk 和 TCache 挤占淘汰的地址);</p> <p>[43]: 一次请求类型错;</p> <p>[42]: 多个 TMAF 条目全地址相同;</p> <p>[41]: MAQ#1 满溢出;</p> <p>[40]: CPM-MC#1 读响应协议错 (主存读数据不是 4 个 flit, 或 IO 读数据不是 1 个 flit, 或 4 个主存读数据携带的 CPM ID 号不一致);</p> <p>[39]: MAQ#0 满溢出;</p> <p>[38]: CPM-MC#0 读响应协议错;</p>
CPM 偶校验错 (不可纠错)			
CPM_ParityErr	[37:26]	RW1C, 0	<p>[37]: DATA 修复地址 Latch 偶校验错;</p> <p>[36]: TMAD2 偶校验错;</p> <p>[35]: MC#1 读控制偶校验错;</p> <p>[34]: MC#0 读控制偶校验错;</p> <p>[33:30]: MPE#3~ MPE#0 回答控制偶校验错;</p> <p>[29:26]: MPE#3~ MPE#0 请求控制偶校验错;</p>

CPM ECC 多错（不可纠错）			
CPM_MErr	[25:13]	RW1C, 0	[25]: IO 写数据 ECC 多错; [24]: IPU 响应控制 flit 多错; [23]: IPU 请求控制 flit 多错; [22]: IPU 回答数据 ECC 多错; [21]: TMAD1 ECC 多错; [20:17]: MPE#3~ MPE#0 回答数据 ECC 多错; [16]: MC#1 读数据 ECC 多错; [15]: MC#0 读数据 ECC 多错; [14]: TTag 阵列 ECC 多错; [13]: TData 阵列 ECC 多错;
CPM ECC 单错（可纠错）			
CPM_SErr	[12:0]	RW1C, 0	[12]: IO 写数据 ECC 单错; [11]: IPU 响应控制 flit 单错; [10]: IPU 请求控制 flit 单错; [9]: IPU 回答数据 ECC 单错; [8]: TMAD1 ECC 单错; [7:4]: MPE#3~ MPE#0 回答数据 ECC 单错; [3]: MC#1 读数据 ECC 单错; [2]: MC#0 读数据 ECC 单错; [1]: TTag 阵列 ECC 单错; [0]: TData 阵列 ECC 单错;
—	其它	—	保留，不可写，读出数据固定为 0。

### 2.1.10 CPM\_FaultEn

CPM\_FaultEn 是一致性处理部件的故障使能寄存器，按位屏蔽 CPM 内的各种错误是否报告给

MCU，不屏蔽 ECC 单错。

表 2-19: CPM\_FaultEn 寄存器域的描述

名称	范围	类型	描述
CPMFaultEn	[58:13]	RW,全 1	CPM 内各个报错信号的使能标志，默认为使能有效，按位对应 CPM 故障状态寄存器 CPM_Fault[58:13];

—	其它	—	保留，不可写，读出数据固定为 0
---	----	---	------------------

### 2.1.11 CPM\_TReqCnt

CPM\_TReqCnt 是一致性处理部件记录可 Cache 请求数量的计数器，核心和维护只读。该计数器的启动、停止和清除操作需要通过 IO 写“CPM 性能计数器控制寄存器 (CPM\_PerfCtrl)”来控制。

表 2-11: CPM\_TReqCnt 寄存器域的描述

名称	范围	类型	描述
TReqCnt	[31:0]	RO	记录核心发向 CPM 的可 Cache 请求 (可 Cache 读、可 Cache 写、置脏、原子操作) 的数量, 用于统计 TCache 命中率。该计数值到达最大值时, 可 Cache 请求计数器、TCache 直接命中计数器、TCache 脱靶计数器停止计数。
—	其它	—	保留, 读出数据固定为 0

### 2.1.12 CPM\_TCHitCnt

CPM\_TCHitCnt 是一致性处理部件记录 TCache 命中次数的计数器，核心和维护只读。该计数器的启动、停止和清除操作需要通过 IO 写“CPM 性能计数器控制寄存器 (CPM\_PerfCtrl)”来控制。

表 2-12: CPM\_TCHitCnt 寄存器域的描述

名称	范围	类型	描述
TCHitCnt	[31:0]	RO	记录核心的可 Cache 请求直接命中 TCache (从 TCache 读取数据) 的次数。
—	其它	—	保留, 读出数据固定为 0

### 2.1.13 CPM\_TCMissCnt

CPM\_TCMissCnt 是一致性处理部件记录 TCache 脱靶次数的计数器，核心和维护只读。该计数器的启动、停止和清除操作需要通过 IO 写“CPM 性能计数器控制寄存器 (CPM\_PerfCtrl)”来控制。

表 2-13: CPM\_TCMissCnt 寄存器域的描述

名称	范围	类型	描述
TCMissCnt	[31:0]	RO	记录核心的可 Cache 请求发生 TCache 脱靶 (从主存读取数据) 的次数。 TCache 命中率 = $1 - (\text{TCache 脱靶计数} / \text{可 Cache 请求计数})$
—	其它	—	保留, 读出数据固定为 0

### 2.1.14 CPM\_AddrCfltCnt

CPM\_TCMissCnt 是一致性处理部件记录一次请求在一致性流水线上发生地址冲突的次数（一次请求与 TMAF 条目的全地址相同，或与 FRQ 年老条目的 TCache 索引地址相同），核心和维护只读。该计数器的启动、停止和清除操作需要通过 IO 写“CPM 性能计数器控制寄存器（CPM\_PerfCtrl）”来控制。

表 2-14: CPM\_AddrCfltCnt 寄存器域的描述

名称	范围	类型	描述
AddrCfltCnt	[23:0]	RO	记录一致性流水线上的请求发生 TCache 索引冲突的次数。该计数值到达最大值时停止计数，维持最大值不变；
—	其它	—	保留，读出数据固定为 0

### 2.1.15 CPM\_FRQCmprCnt

CPM\_FRQCmprCnt 是一致性处理部件记录 FRQ 压缩性能的计数器，核心和维护只读。该计数器的启动、停止和清除操作需要通过 IO 写“CPM 性能计数器控制寄存器（CPM\_PerfCtrl）”来控制。

表 2-15: CPM\_FRQCmprCnt 寄存器域的描述

名称	范围	类型	描述
FRQCmprCnt	[23:0]	RO	记录 FRQ 尾指针回溯的条目数。该计数值到达最大值时停止计数，维持最大值不变；
—	其它	—	保留，读出数据固定为 0

### 2.1.16 CPM\_TMAFMax

CPM\_TMAFMax 是一致性处理部件用于记录 TMAF1/TMAF2 的最大使用情况，核心和维护只读。该计数器的启动、停止和清除操作需要通过 IO 写“CPM 性能计数器控制寄存器（CPM\_PerfCtrl）”来控制。

表 2-16: CPM\_TMAFMax 寄存器域的描述

名称	范围	类型	描述
TMAF1Max	[7:0]	RO	TMAF1 的最大使用编号（0~23）；
TMAF2Max	[15:8]	RO	TMAF2 的最大使用编号（0~15）；
—	其它	—	保留，读出数据固定为 0



## 2.2 存储控制器内部的 IO 寄存器

存储控制器的 IO 寄存器中，有部分信息域用于控制各种命令的发送间隔。如无特殊说明，这些信息域对应计数器的工作时钟均为“DDR3 存储器接口输出时钟”，其复位后的初值，是按照 JEDEC DDR3 标准中，DDR3-1600（对应的“DDR3 存储器接口输出时钟”频率为 800MHz）、8Gb 存储器颗粒要求的典型配置值，折算为“DDR3 存储器接口输出时钟”的周期数。在使用申威 411 处理器时，应该根据主存的实际工作频率和颗粒容量，参照存储器颗粒的相关手册进行配置。

每个存控对应的 DDR3 接口物理层电路中包含一套存储器地址/命令/数据控制接口，用于控制 DDR3 接口的命令、地址和数据信息。地址/命令/数据控制接口包括 1 组地址/命令接口（简称 AC）和 9 组数据接口（简称 Datx8），每组 Datx8 对应 8 位数据。AC/Datx8 子模块统称 PHY，AC/Datx8 的公共控制模块统称 PUB。

### 2.2.1 CfgTR0

CfgTR0 为时序寄存器 0，可读写，该寄存器用于控制 DDR3 刷新时序。

表 2-3: CfgTR0 寄存器域的描述

名称	范围	类型	描述
CfgREFI	[18:9]	RW,195	两次刷新命令之间的平均间隔周期数。（REFI 计数器使用维护时钟进行计数，该时钟频率为 25MHz 时，初值“195”
CfgRFC	[8:0]	RW,280	REF 命令到 REF/ACT 命令之间的最小间隔周期数。（该默认配置按 8Gb 的颗粒进行设置，即 350ns）
—	其它	—	保留。

注：CfgRFC 的设置与存储器颗粒的容量有关，具体见下表：

表 2-4: CfgRFC 与存储器颗粒的关系

存储器颗粒容量	CfgRFC (ns)
512Mb	90
1Gb	110
2Gb	160
4Gb	300
8Gb	350

## 2.2.2 CfgTR1

CfgTR1 为时序寄存器 1，可读写，该寄存器用于控制 DDR3 Speed Bins 时序。

表 2-5: CfgTR1 寄存器域的描述

名称	范围	类型	描述
CfgRAS	[23:18]	RW,28	ACT 到 PRE 命令的最小间隔周期数。
CfgRC	[17:12]	RW,39	ACT 到 ACT/REF 命令的最小间隔周期数。
CfgRP	[11:8]	RW,9	PRE 命令最小间隔周期数。
CfgRCD	[7:4]	RW,9	ACT 到读写命令 (CAS) 之间的最小间隔周期数。
CfgAA	[3:0]	RW,9	读命令到其第一个数据上数据总线的最小时间间隔。(相当于 11.25ns@800MHz)。注: 该延时等同于 CL (即 tCfgAA/tCK)。SW1610 具体实现由 CL 进行延时控制;
—	其它	—	保留。

## 2.2.3 CfgTR2

CfgTR2 为时序寄存器 2，可读写，该寄存器用于控制 DDR3 命令和地址时序。

表 2-6: CfgTR2 寄存器域的描述

名称	范围	类型	描述
CfgDQS_WW	[63:61]	RW,2	不同 Rank 之间写到写切换的额外延迟。
CfgDQS_RR	[60:58]	RW,2	不同 Rank 之间读到读切换的额外延迟。
CfgDQS_WR	[57:54]	RW,2	不同 Rank 之间写到读切换的额外延迟。
CfgCCDR	[53:51]	RW,4	RD 到 RD 的最小间隔周期数。
CfgDQS_RW	[50:48]	RW,2	不同 Rank 之间读到写切换的额外延迟。
CfgRTW	[47:44]	RW,6	读到写切换延时控制。实际切换延时计数方式如下: (1) Rank0->Rank0: 4+(RL-WL)+2+CfgRTW; (2) Rank0->Rank1: 4+(RL-WL)+CfgDQS; (3) Rank1->Rank1: 同 (1); (4) Rank1->Rank0: 同 (2)。
CfgFAW	[43:38]	RW,32	4 个活动 Bank 的窗口时间周期。
CfgRRD	[37:34]	RW,6	ACT 到 ACT 命令之间的最小间隔周期数 (不同 Bank)。

CfgCCDW	[33:31]	RW,4	WR 到 WR 的最小间隔周期数。
CfgMOD	[30:26]	RW,12	模式寄存器更新延迟周期数。
CfgMRD	[25:23]	RW,4	模式寄存器设置时间，即 2 条 MRS 命令间的最小间隔周
CfgWR	[22:18]	RW,12	写恢复 (Write Recovery) 时间 (周期数)。
CfgWTR	[17:14]	RW,6	同一个 Bank 内的写命令到读命令之间的最小间隔周期。
CfgRTP	[13:10]	RW,6	同一个 Bank 内的读命令到 PRE 命令之间的最小间隔周
CfgDLLK	[9:0]	RW,512	DLL 锁定所需的最短时间 (周期数)。
—	其它	—	保留。

### 2.2.4 CfgTR3

CfgTR3 为时序寄存器 3，可读写，该寄存器用于控制端接校准 (ZQ Calibration) 和初始化时序。表 2-7: CfgTR3 寄存器域的描述

名称	范围	类型	描述
CfgXPR	[34:26]	RW,288	复位结束后，从 CKE 到第一个有效的命令之间的间隔。
CfgZQCS	[25:19]	RW,64	正常操作期间短校准时间 (周期数)。
CfgZQoper	[18:10]	RW,256	正常操作期间完全校准时间 (周期数)。
CfgZQinit	[9:0]	RW,512	上电和复位过程的校准时间 (周期数)。
—	其它	—	保留。

### 2.2.5 CfgTR4

CfgTR4 为时序寄存器 4，可读写，该寄存器用于控制自刷新时序。

表 2-8: CfgTR4 寄存器域的描述

名称	范围	类型	描述
CfgCKSRX	[30:27]	RW,8	自刷新、PowerDown 退出或者复位退出前时钟必须有效最少时间。
CfgCKSRE	[26:23]	RW,8	进入自刷新或 PowerDown 后时钟仍需要保持有效的最少时间。
CfgCKESR	[22:19]	RW,7	自刷新期间 CKE 为低的最少周期数。
CfgXSDLL	[18:9]	RW,512	退出自刷新到需要锁定的 DLL 命令之间的最小间隔周期数。
CfgXS	[8:0]	RW,288	退出自刷新到不需要锁定的 DLL 命令间的最小间隔周期数。

—	其它	—	保留。
---	----	---	-----

## 2.2.6 CfgTR5

CfgTR5 为时序寄存器 5，可读写，该寄存器用于控制 Power Down 时序。

表 2-9: CfgTR5 寄存器域的描述

名称	范围	类型	描述
CfgWRAPDEN	[36:31]	RW,35	WRA 命令到进入 PowerDown 的最小间隔周期。
CfgWRPDEN	[30:25]	RW,34	WR 命令到进入 PowerDown 的最小间隔周期。
CfgRDPDEN	[24:20]	RW,22	RD/RDA 命令到 PowerDown 进入的最小间隔周期。
CfgPD	[19:11]	RW,6	进入 PowerDown 到退出时间最小间隔周期数。该配置值不能小于 CfgCKE。
CfgCKE	[10:8]	RW,6	CKE 的最小脉冲宽度(周期数)。
CfgXPDLL	[7:3]	RW,20	退出 Precharge PowerDown (DLL frozen) 到需要一个锁定的 DLL 的命令之间的最小间隔周期数。
CfgXP	[2:0]	RW,6	退出 PowerDown 状态时, DLL 打开到一个有效命令的最小间隔周期数, 或者是从退出 Precharge PowerDown (DLL frozen) 到一个不需要锁定的 DLL 命令间的最小间隔周期
—	其它	—	保留。

## 2.2.7 CfgTR6

CfgTR6 为时序寄存器 6，可读写，该寄存器用于控制片上端接 (ODT) 时序。

表 2-10: CfgTR6 寄存器域的描述

名称	范围	类型	描述
CfgAOFPD	[9:7]	RW,1	异步 ODT 关闭延时 (周期数)。未实现异步 ODT;
CfgAONPD	[6:4]	RW,1	异步 ODT 打开延时 (周期数)。未实现异步 ODT;
CfgODTH8	[3:0]	RW,6	ODT 为高的时间 (周期数)。
—	其它	—	保留。

## 2.2.8 CfgTR7

CfgTR7 为时序寄存器，可读写，该寄存器用于控制其它时序。

表 2-11: CfgTR7 寄存器域的描述

名称	范围	类型	描述
CfgZQI	[49:28]	RW, 2500000	正常运行期间，ZQCL/S 命令的产生间隔周期数。 (ZQI 计数器使用维护时钟进行计数)
CfgRSTH	[27:13]	RW, 15000	初始化阶段复位信号为高电平的时间。(RSTH 计数器使用维护时钟进行计数)
CfgRSTL	[12:0]	RW, 6000	初始化阶段复位信号为低电平的时间。(RSTL 计数器使用维护时钟进行计数)
—	其它	—	保留。

## 2.2.9 CfgDec

CfgDec 记录了当前存控中部分配置的摘要内容。

表 2-12: CfgDec 寄存器域的描述

名称	范围	类型	描述
IsMPR	[57]	RO,0	为“1”时，指示 MPR 处于打开状态。
IsDynODTOn	[56]	RO,0	为“1”时，指示动态 ODT 处于打开状态。
IsRttWRRZQD4	[55]	RO,0	RttWR 配置。
IsRttWRRZQD2	[54]	RO,0	RttWR 配置。
IsSRTNormal	[53]	RO,1	SRT 为 normal 模式。
由于代码描述中位宽不匹配，造成[53]~[57]丢失，读取不到			
IsASREnable	[52]	RO,0	ASR 使能。
IsCWL10	[51]	RO,0	为“1”时，指示 CWL 配置为“10”。
IsCWL9	[50]	RO,0	为“1”时，指示 CWL 配置为“9”。
IsCWL8	[49]	RO,0	为“1”时，指示 CWL 配置为“8”。
IsCWL7	[48]	RO,0	为“1”时，指示 CWL 配置为“7”。
IsCWL6	[47]	RO,1	为“1”时，指示 CWL 配置为“6”。
IsCWL5	[46]	RO,0	为“1”时，指示 CWL 配置为“5”。
IsFullArray	[45]	RO,1	全阵列自刷新。
IsHalfArrayL	[44]	RO,0	低段半阵列自刷新。

IsQuarArrayL	[43]	RO,0	低段四分之一阵列自刷新。
Is1by8ArrayL	[42]	RO,0	低段八分之一阵列自刷新。
Is3by4Array	[41]	RO,0	四分之三阵列自刷新。
IsHalfArrayH	[40]	RO,0	高段半阵列自刷新。
IsQuarArrayH	[39]	RO,0	高段四分之一阵列自刷新。
Is1by8ArrayH	[38]	RO,0	高段八分之一阵列自刷新。
IsOutEnable	[37]	RO,1	输出使能。
IsTDQSEnable	[36]	RO,0	TDQS 使能。
IsSynODTOn	[35]	RO,1	同步 ODT 打开。
IsRttNOMRZQD4	[34]	RO,1	RttNOMR 配置为 1/4。
IsRttNOMRZQD2	[33]	RO,0	RttNOMR 配置为 1/2。

IsRttNOMRZQD6	[32]	RO,0	RttNOMR 配置为 1/6。
IsRttNOMRZQD12	[31]	RO,0	RttNOMR 配置为 1/12。
IsRttNOMRZQD8	[30]	RO,0	RttNOMR 配置为 1/8。
IsWLEnable	[29]	RO,0	Write Leveling 使能。
IsOutputRZQD6	[28]	RO,1	输出电阻选择为 1/6。
IsOutputRZQD7	[27]	RO,0	输出电阻选择为 1/7。
IsAL0	[26]	RO,1	为“1”时，指示 AL 配置为“0”。
IsALCL_1	[25]	RO,0	为“1”时，指示 AL 配置为“CL-1”。
IsALCL_2	[24]	RO,0	为“1”时，指示 AL 配置为“CL-2”。
IsDLLEnable	[23]	RO,1	DLL 使能。
IsFastExit	[22]	RO,1	快速退出 Precharge PowerDown。
IsWR5	[21]	RO,0	为“1”时，指示 WR 配置为“5”。
IsWR6	[20]	RO,0	为“1”时，指示 WR 配置为“6”。
IsWR7	[19]	RO,0	为“1”时，指示 WR 配置为“7”。
IsWR8	[18]	RO,0	为“1”时，指示 WR 配置为“8”。
IsWR10	[17]	RO,0	为“1”时，指示 WR 配置为“10”。
IsWR12	[16]	RO,1	为“1”时，指示 WR 配置为“12”。
IsDLLReset	[15]	RO,1	DLL 复位。
IsTestMode	[14]	RO,0	测试模式。
IsCL14	[13]	RO,0	为“1”时，指示 CL 配置为“14”。
IsCL13	[12]	RO,0	为“1”时，指示 CL 配置为“13”。
IsCL12	[11]	RO,0	为“1”时，指示 CL 配置为“12”。

IsCL11	[10]	RO,0	为“1”时，指示 CL 配置为“11”。
IsCL10	[9]	RO,0	为“1”时，指示 CL 配置为“10”。
IsCL9	[8]	RO,0	为“1”时，指示 CL 配置为“9”。
IsCL8	[7]	RO,0	为“1”时，指示 CL 配置为“8”。
IsCL7	[6]	RO,1	为“1”时，指示 CL 配置为“7”。
IsCL6	[5]	RO,0	为“1”时，指示 CL 配置为“6”。
IsCL5	[4]	RO,0	为“1”时，指示 CL 配置为“5”。
IsInterleave	[3]	RO,1	为“1”时，指示突发类型为交错。
IsBL8	[2]	RO,1	为“1”时，指示突发长度为“8”。
IsBL8orBC4	[1]	RO,0	为“1”时，指示突发长度为“8”或者“4”。
IsBC4	[0]	RO,0	为“1”时，指示突发长度为“4”。
—	其它	—	保留。

## 2.2.10 CfgCR

CfgCR 为控制寄存器。该寄存器必须先于其它存储控制器的 IO 寄存器进行配置。

表 2-13: CfgCR 寄存器域的描述

名称	范围	类型	描述
UPDResCntEn	[48]	RW,0	Update 状态机从 pUpdST_WAIT 跳转到 pUpdST_UPD 时，是否判断发出去的 MEM 读都已返回响应使能，为 0，表示不对此判断，为 1，表
RankSelCnt	[47:42]	RW,7	双 Rank 时，Rank 间的切换间隔。
UPDWaitTime	[41:36]	RW,12	配置 Update Req 到 Update Ack 的间隔。最大配置 63 拍。
AtomUPD	[35]	RW,1	该位为“1”，表示 Update 处理遵守与其他原子序列的关系。
UPDForbiddenAll	[34]	RW,1	该位为“1”，表示 Update 请求需要禁止所有
UpdateEn	[33]	RW,0	对于 PHY 发起的 Update 请求，存控的确认使能。如果该位为“1”，则予以确认。
gr_EnableWrap	[32]	RW,0	存控控制 Data Training 时起作用，该位为“1”则使能 Wrap 地址，否则屏蔽 Wrap 地址（清

ForceResponse	[31]	RW,0	DQS Gate Training 的超时强制响应，主要用于调试。
RDIMM	[30]	WR,0	RDIMM 指示。
DisableREF	[28]	RW,0	关闭刷新功能。
2T	[27]	RW,1	2T 模式使能。
AddressMirror	[26]	RW,0	地址镜像使能。
DTUWrSEL	[25:23]	RW,0	MPR 模式下，用于选择第 (0~7) 个 144 位写数
DTURdSEL	[22:20]	RW,0	MPR 模式下，用于选择第 (0~7) 个 144 位读数
ForbiddenReceive	[16]	RW,1	禁止将 DDR3 物理通路接收到的读响应数据转发给一致性处理部件。该功能用于消除 DDR3 接口
ForbiddenArbWR	[15]	RW,0	禁止仲裁 WR 命令。用于 DEBUG。
保留	[14:12]	—	—
FPGAStart	[11]	RW,0	FPGA 启动指示。(用于非 Synopsys PHY)
ForbiddenArbRD	[10]	RW,0	禁止仲裁 RD 命令。用于 DEBUG。
DisablePD	[9]	RW,0	关闭 PD 功能。
DisableSR	[8]	RW,0	关闭 SR 功能。
DisableZQ	[7]	RW,0	关闭 ZQ 功能。
DisableECC	[6]	RW,0	关闭 ECC 校验。
ForbiddenArbACT	[5]	RW,0	禁止仲裁 ACT 命令。用于 DEBUG。
ForbiddenArbPRE	[4]	RW,0	禁止仲裁 PRE 命令。用于 DEBUG。
FastReq	[3]	RW,0	快速产生请求。该功能仅用于验证。芯片正常使用时，该位必须配置为 0。
DFTBypassDTU	[2]	RW,1	旁路数据通路信号调节。该信号只在存控进行 Data Training 状态有意义。如果需要存控进行数据通路信号调节时，将该为改为“0”。
DFTZQCLSEL	[1]	RW,0	选择 ZQCL 命令。
DFTFastInit	[0]	RW,1	快速初始化(在 CTL 进行的 SDRAM 初始化过程中，控制 Sdram_Reset 在 High 或 Low 等待的时



## 2.2.11 MC\_CTRL

MC\_CTRL 为存储控制器内部控制寄存器，该寄存器中包含存储控制器的 ODT 控制、主存配置 和错误检查使能信息。

表 2-14: MC\_CTRL 寄存器域的描述

名称	范围	类型	描述
BANKHASH	[26]	WR,0	BANK 地址 hash 选择。 首先根据 MAPCFG 选择地址映射模式，在根据该位决定是否对地址进行 Hash。
MAPCFG	[25:23]	WR,0	地址映射模式选择。
MEMCAP	[22:21]	WR,0	主存容量，与 Rank 类型选择有关，具体如下： RANKSEL = “0” 时 “0”：1GB； “1”：2GB； “2”：4GB； “3”：8GB。 RANKSEL = “1” 时 “0”：2GB； “1”：4GB； “2”：8GB； “3”：保留。
DVTYPE	[20]	WR,0	颗粒类型： “0”：x8 颗粒； “1”：x16 颗粒。
WRODT1	[19:18]	RW,0x2	对 Rank1 进行写操作时，Rank1 和 Rank0 的 ODT 控制使能。高位对应 Rank1，低位对应 Rank0。
RDODT1	[17:16]	RW,0x0	对 Rank1 进行读操作时，Rank1 和 Rank0 的 ODT 控制使能。高位对应 Rank1，低位对应 Rank0。
WRODT0	[15:14]	RW,0x1	对 Rank0 进行写操作时，Rank1 和 Rank0 的 ODT 控制使能。高位对应 Rank1，低位对应 Rank0。
RDODT0	[13:12]	RW,0x0	对 Rank0 进行读操作时，Rank1 和 Rank0 的 ODT 控制使能。高位对应 Rank1，低位对应 Rank0。

ActPri	[11]	WR,0x0	存储器命令 (ACT 和非 ACT) 仲裁控制, 该位为 “1”, 表示 ACT 优先。
RANKSEL	[10]	WR,0x0	Rank 类型选择: “0” : 单 Rank; “1” :

ArbCFG	[9:4]	WR,0	读写请求仲裁控制。 [5:4]表示仲裁类型：2'b00 表示选择与上一个请求相同属性；2'b01 表示读优先；2'b10 表示轮转；2'b11 表示按固定时隙； [3]表示按固定时隙，读写时隙比例，该位为“1”表示读写比例是 2:1，否则读写比例是 1:1； [2:0]表示按固定时隙时的时间单位： $2^{*(Value+1)}$ ； 表示固定时隙的时间单位时，有效配置是 0x0~0x5。
WRDATMERR_EN	[3]	WR,0	写数据多错检查使能，该位为“1”时，如果写主存数据发现不可纠正多错，则产生
WRDATMSERR_EN	[2]	WR,0	写数据可纠正错记录使能，该位为“1”时，如果写主存数据发现可纠正错，则产生
MEMMERR_EN	[1]	WR,0	主存多错检查使能，该位为“1”时，如果读主存时发现不可纠正多错，则产生 MEMMERR。
MEMSERR_EN	[0]	WR,0	主存可纠正错记录使能，该位为“1”时，如果读主存时发现可纠正错，则产生 MEMSERR。
—	其它	—	保留。

## 2.2.12 INIT\_CTRL

INIT\_CTRL 是存控初始化流程控制寄存器，可读写。

表 2-15: INIT\_CTRL 寄存器域的描述

名称	范围	类型	描述
DATATRAN_MC	[3]	RW,0	该位为“1”表示由用户软件控制存储器控制进行 Data Training。该位与 WAITPUB_DONE 结合使用，一起自
WAITPUB_DONE	[2]	RW,0	表示通过触发 PUB 已经完成 SDRAM 初始化和 Data Training。用户配置该寄存器进入下一个状态；该位
SDRAMINIT_MC	[1]	RW,0	该位为“1”表示 SDRAM 初始化由用户软件控制存储控制器进行，否则由用户触发 PUB 自动完成。推荐使用触发 PUB 自动完成的方式。 该位与 PHYINIT_DONE 结合使用，一起自清除。
PHYINIT_DONE	[0]	RW,0	表示 PHY 已经自动完成初始化。用户配置该寄存器

			入下一个状态；该位会自清除。
—	其它	—	保留。

### 2.2.13 ClosePage

ClosePage 是页面管理策略配置寄存器，可读写。默认是 OpenPage 策略。

表 2-16: ClosePage 寄存器域的描述

名称	范围	类型	描述
ClosePage	[0]	RW,0	该位为“1”表示采用 ClosePage 策略，否则采用 OpenPage 策略。
—	其它	—	保留。

### 2.2.14 DTUR

DTUR 为数据通路信号调节 (Data Training) 读命令寄存器，进行数据通路信号调节时，对于该寄存器的写将触发一次存储器读。通过该寄存器可以配置读地址、Bank 和 Rank，通过读 DTURD0~DTURD2 获得读数据。

表 2-17: DTUR 寄存器域的描述

名称	范围	类型	描述
DTURdCAAddr	[31:21]	RW,0	数据通路信号调节时，读请求的列地址。
DTURdRAAddr	[20:5]	RW,0	数据通路信号调节时，读请求的行地址。
DTURdBank	[4:2]	RW,0	数据通路信号调节时，读请求的 Bank 号。
DTURdRank	[1]	RW,0	数据通路信号调节时，读请求的 Rank 号。
DTURdValid	[0]	RW,0	数据通路信号调节读请求有效。该位具有自清除功能。
—	其它	—	保留。

### 2.2.15 DTUW

DTUW 为数据通路信号调节 (Data Training) 写命令寄存器，进行数据通路信号调节时，对于该寄存器的写将触发一次存储器写。通过该寄存器可以配置写地址、Bank 和 Rank，通过 DTUWD0~DTUWD2 指定写数据。

表 2-18: DTUW 寄存器域的描述

名称	范围	类型	描述
----	----	----	----

DTUWrCAddr	[31:21]	RW,0	数据通路信号调节时，写请求的列地址。
------------	---------	------	--------------------



DTUWrRAddr	[20:5]	RW,0	数据通路信号调节时，写请求的行地址。
DTUWrBank	[4:2]	RW,0	数据通路信号调节时，写请求的 Bank 号。
DTUWrRank	[1]	RW,0	数据通路信号调节时，写请求的 Rank 号。
DTUWrValid	[0]	RW,0	数据通路信号调节写请求有效。该位可自清除。
—	其它	—	保留。

### 2.2.16 DTUWD0

DTUWD0 为数据通路信号调节 (Data Training) 写数据寄存器 0。

表 2-19: DTUWD0 寄存器域的描述

名称	范围	类型	描述
DTUWD0	[7:0]	RW,0	数据通路信号调节写数据对应的 ECC。
—	其它	—	保留。

### 2.2.17 DTUWD1

DTUWD1 为数据通路信号调节 (Data Training) 写数据寄存器 1。

表 2-20: DTUWD1 寄存器域的描述

名称	范围	类型	描述
DTUWD1	[63:0]	RW,0	数据通路信号调节写数据。

### 2.2.18 DTURD0

DTURD0 为数据通路信号调节 (Data Training) 读数据寄存器 0。

表 2-21: DTURD0 寄存器域的描述

名称	范围	类型	描述
DTURD0	[15:0]	RW	数据通路信号调节读数据对应的 ECC。
—	其它	—	保留。

### 2.2.19 DTURD1

DTURD1 为数据通路信号调节 (Data Training) 读数据寄存器 1。

表 2-22: DTURD1 寄存器域的描述

名称	范围	类型	描述
----	----	----	----



DTURD1	[63:0]	RW	数据通路信号调节读数据。
--------	--------	----	--------------

### 2.2.20 DTUCTL

DTUCTL 为数据通路信号调节 (Data Training) 测试结束控制，用于初始化流程。

表 2-23: DTUCTL 寄存器域的描述

名称	范围	类型	描述
DTUEnd	[0]	RW,0	数据通路信号调节测试完成。对该位写 1，可以强制结束数据通路信号调节。具有自清除功能。
—	其它	—	保留。

### 2.2.21 DTUPREA

DTUPREA 为数据通路信号调节 (Data Training) PREA 命令寄存器，进行数据通路信号调节测试时，对于该寄存器的写将触发一次 PREA 命令。通过该寄存器可以配置 PREA 命令的目标 Rank。

表 2-24: DTUPREA 寄存器域的描述

名称	范围	类型	描述
DTUPREARank	[1]	RW,0	数据通路信号调节期间产生 PREA 命令对应的 Rank。
DTUPREAValiD	[0]	RW,0	数据通路信号调节期间产生 PREA 命令。对该位写
—	其它	—	保留。

### 2.2.22 WL\_CTRL

WL\_CTRL 寄存器用于非全流程的 Write Leveling 控制，可读写；

表 2-25: WL\_CTRL 寄存器域的描述

名称	范围	类型	描述
WLRankValid	[2]	RW,0	该位为“1”，表示 MRS 命令和 ODT 都是对指定 Rank 的控制。当 IOR: CfgMR1[WL]为“1”时，该位有

WLRank	[1]	RW,0	当 IOR: CfgMR1[WL]为“1”，且 WLRankValid 为“1”时该位有意义，该位为“0”，表示指定
			MRS 命令只发向该位指定的 Rank。
WLODTON	[0]	RW,0	当 IOR: CfgMR1[WL]为“1”，且 WLRankValid 为“1”时，该位有意义，该位为“0”，表示打开 ODT，否则表示关闭 ODT。注：此时存储控制器发出的
—	其它	—	保留。

### 2.2.23 MEMSERR\_P

MEMSERR\_P 用于记录已纠正校验错的相关信息。复位时为全“0”。

表 2-26: MEMSERR\_P 寄存器域的描述

名称	范围	类型	描述
Valid	[8]	RW1C,0	发生主存数据单错错标志。
ECC	[7:0]	RO,0	发生主存数据单错错时的诊断码。
—	其它	—	保留。

当存储器访问发生主存数据单错（MEMSERR），且 IOR: MEMSERR\_P[Valid]为“0”时，置 Valid 位为“1”，并在 IOR: MEMSERR\_P 中登记发生可纠错的 64 位数据的 16 位诊断码，用于判断单错发生的位置，该诊断码只记录第 1 个已纠正校验错的信息。

### 2.2.24 CfgErr

CfgErr 用于记录当前配置合法性检查结果。

表 2-27: CfgErr 寄存器域的描述

名称	范围	类型	描述
ErrorASRSRT	[10]	RO,0	ASR 和 SRT 同时有效。
ErrorMPR	[9]	RO,0	MPR 配置错。正常操作模式必须 Disable MPR。
ErrorWLeveling	[8]	RO,0	Write Leveling 配置错。正常操作模式必须 Disable Write Leveling。Write Leveling 必须关闭动态 ODT。

ErrorDlLODT	[7]	RO,0	DLL 与 ODT 组合配置错。 DLL Disable 是必须关闭 ODT。
ErrorRBT	[5]	RO,0	MRS 配置错。突发类型不能配置为顺序模式。
ErrorBL8	[4]	RO,0	突发长度不是“8”。
ErrorMBZ	[3]	RO,0	必须为“0”的配置实际没有配“0”。 MR0[7]: 测试模式必须为零; MR1[8]和 MR1[10]必须为零; MR1[11]必须为零; MR1[12]在非 Write Leveling 时必须为零; MR2[8]和 MR2[2:0]必须为零; MR3[1:0]必须为零。
ErrorRsv	[2]	RO,0	保留位配置错。 CfgMR0~CfgMR3 的配置信息使用了保留值。
ErrorCfg	[1]	RO,0	模式寄存器配置错 (总的出错标志)。
ErrorCfgFPGA	[0]	RO,1	FPGA 配置错。RL 配置不为 6, 或者 WL 配置不为 6, 或者 DLL 使能则报 FGPA 配置错;
—	其它	—	保留。

## 2.2.25 FSMSTAT

FSMSTAT 用于记录存储控制器内部的状态机状态。

表 2-28: FSMSTAT 寄存器域的描述

名称	范围	类型	描述
----	----	----	----

SMSDRAM	[6:3]	RO,0	<p>SDRAM 初始化状态机（只有在用户软件控制存储控制器进行 SDRAM 进行初始化时有意义）。</p> <p>0: 空闲状态；</p> <p>1: Reset_N 为低；</p> <p>2: Reset_N 为高；</p> <p>3: 等待 XPR 满足；</p> <p>4/11: 发送 MR2 命令（Rank0/Rank1）；</p> <p>5/12: 发送 MR3 命令（Rank0/Rank1）；</p> <p>6/13: 发送 MR1 命令（Rank0/Rank1）；</p> <p>7/14: 发送 MR0 命令（Rank0/Rank1）；</p> <p>8/15: 发送 ZQ 命令（Rank0/Rank1）；</p> <p>9: DTU 测试；</p> <p>10: 初始化结束状态</p>
SMINIT	[2:0]	RO,0	<p>存控初始化状态机。</p> <p>0: PHY 初始化状态；</p> <p>1: 存控进行 SDRAM 初始化状态；</p> <p>2: PUB 进行 SDRAM 初始化状态； 3: 存控进行 Data Training 状态； 4: 正常访问状态；</p> <p>其它: 保留。</p>
—	其它	—	保留。

## 2.2.26 RC

RC 为 RDIMM 配置寄存器。外接 RDIMM 存储条时，需要写该寄存器触发 WRC 命令，对 RC 进行配置。

表 2-29: RC 寄存器域的描述

名称	范围	类型	描述
ReqWRCValid	[8]	RW,0	RC 配置请求的有效位。
ReqWRCAddr	[7:4]	RW,0	RC 配置地址。
ReqWRCDData	[3:0]	RW,0	RC 配置数据。
—	其它	—	保留。

## 2.2.27 RCMRD

RCMRD 用于配置 RDIMM 的扩展模式寄存器加载等待时间。外接 RDIMM 存储条时，使用该寄存器配置 MRD 计数初值。

表 2-30: RCMRD 寄存器域的描述

名称	范围	类型	描述
CfgRCMRD	[4:0]	RW,8	RC 配置时的 tMRD。
—	其它	—	保留。

## 2.2.28 6U

6U 用于配置 RDIMM 的 RC 寄存器加载等待时间。外接 RDIMM 存储条时，使用该寄存器设置 RC 配置延时。

表 2-31: 6U 寄存器域的描述

名称	范围	类型	描述
Cfg6U	[12:0]	RW,4800	RC 配置时的 6us 延迟。
—	其它	—	保留。

## 2.2.29 PUB\_INTERFACE

PUB\_INTERFACE 为地址/命令/数据接口信号控制寄存器，可读写。

表 2-32: PUB\_INTERFACE 寄存器域的描述

名称	范围	类型	描述
PUB1_dram_clk_disable	[31:29]	RW,0	关闭时钟。
PUB1_data_byte_disable	[28:20]	RW,0	关闭 Datx8。
PUB1_init_start	[19]	RW,0	接口上产生 dfi_inti_start 信号。
PUB1_ret_en_i	[18]	RW,0	配置 ret_en_i。
PUB1_ret_en	[17]	RW,0	配置 ret_en_i。
PUB1_poc_dis	[16]	RW,0	IO 的 Power ON Clear Disable。
PUB0_dram_clk_disable	[15:13]	RW,0	关闭时钟。
PUB0_data_byte_disable	[12:4]	RW,0	关闭 Datx8。
PUB0_init_start	[3]	RW,0	接口上产生 dfi_inti_start 信号。

PUB0_ret_en_i	[2]	RW,0	配置 ret_en_i。
PUB0_ret_en	[1]	RW,0	配置 ret_en_i。
PUB0_poc_dis	[0]	RW,0	IO 的 Power ON Clear Disable。
—	其它	—	保留。

### 2.2.30 POWERCTRL

POWERCTRL 为存储器低功耗控制寄存器，通过设置该寄存器可以开启和关闭 DDR3 存储器的浅睡眠和深睡眠功能，以实现低功耗控制。

表 2-33: POWERCTRL 寄存器域的描述

名称	范围	类型	描述
DSLEEP	[3:2]	RW,0	存储器进入深睡眠状态的时间阈值。该计数器使用维护时钟进行计数，该时钟频率为 25MHz 时。0: 关闭; 1: 1ms; 2: 2ms; 3: 3ms。
			该时间与维护时钟频率相关。
SSLEEP	[1:0]	RW,0	存储器进入浅睡眠状态的时间阈值。该计数器使用维护时钟进行计数，该时钟频率为 25MHz 时。0: 关闭; 1: 1us; 2: 2us; 3: 3us。 该时间与维护时钟频率相关。
—	其它	—	保留。

### 2.2.31 PUB.DTO\_CTL

PUB.DTO\_CTL 为 PUB DTO/ATO 信号控制寄存器，可读写。

表 2-34: PUB.DTO\_CTL 寄存器域的描述

名称	范围	类型	描述
PUB0_dto_sr	[20:17]	RW,4'b1010	斜率控制。

PUB0_dto_lb	[16:15]	RW,2'b0	自环测试控制。
PUB0_dto_te	[14:13]	RW,2'b0	ODT 使能控制。
PUB0_dto_pdr	[12:11]	RW,2'b11	无效输入。
PUB0_dto_pdd	[10:9]	RW,2'b0	无效输出。
PUB0_dto_iom	[8:3]	RW,6'b010010	IO 类型控制。
PUB0_dto_oe	[2:1]	RW,2'b11	输出使能。
PUB0_ato_ae	[0]	RW,1'b0	模拟信号控制。
—	其它	—	保留。

### 2.2.32 CfgMR0

MR0 为 DDR3 模式寄存器 0；写该寄存器会触发存储控制器向存储器发一条 MRS 命令，配置

DDR3 存储器颗粒的 MR0 寄存器。

表 2-35: CfgMR0 寄存器域的描述

名称	范围	类型	描述
PPD	[12]	RW,1	进入带 Precharge 的关电状态时，DLL 控制方式：“0”：慢速退出（DLL off）；“1”：快速退出（DLL On）。
WR	[11:9]	RW,6	自动 Precharge 的时候，Write Recovery 时间，配置值与周期数的对应关系如下： “0”：“16”； “1”：“5”； “2”：“6”； “3”：“7”； “4”：“8”； “5”：“10”； “6”：“12”； “7”：“14”。
DLL	[8]	RW,1	DLL 复位，该位为“1”表示 DLL 复位。
TM	[7]	RW,0	测试模式，该位为“1”表示测试模式。

CL	[6:4],[2]	RW,10	CAS 延迟。配置值与周期数的对应关系如下： “0”：保留； “2”：“5”； “4”：“6”； “6”：“7”； “8”：“8”； “10”：“9”； “12”：“10”； “14”：“11”(DDR3-1610 的可选配置)； “1”：“12”； “3”：“13”； “5”：“14”。
RBT	[3]	RW,1	突发类型。 “0”：顺序模式； “1”：缠绕描述。
BL	[1:0]	RW,0	突发长度，译码为“8”。
—	其它	—	保留。

### 2.2.33 CfgMR1

MR1 为 DDR3 模式寄存器 1；写该寄存器会触发存储控制器向存储器发一条 MRS 命令，配置

DDR3 存储器颗粒的 MR1 寄存器。

表 2-36: CfgMR1 寄存器域的描述

名称	范围	类型	描述
Qoff	[12]	RW,0	输出缓冲不使能。
TDQS	[11]	RW,0	TDQS 使能。
MBZ	[10],[8]	RW,0	必需配置为“0”。
Level	[7]	RW,0	Write Leveling 使能。
AL	[4:3]	RW,0	附加延时： “0”：AL=0； “1”：AL=CL-1； “2”：AL=CL-2； “3”：保留。



Rtt_Nom	[9],[6],[2]	RW,1	Rtt_Nom 阻值选择，其中 RZQ 为 240 欧姆： “0”：关闭 Rtt_Nom； “1”：RZQ/4； “2”：RZQ/2； “3”：RZQ/6； “4”： RZQ/12； “5”：RZQ/8； “6”：保留； “7”：保留。
DIC	[5],[1]	RW,0	输出驱动电阻控制： “0”：RZQ/6； “1”：RZQ/7； “2”：RZQ/TBD； “3”：RZQ/TBD。
DLL	[0]	RW,0	DLL 使能： “0”：表示使能； “1”：表示不使
—	其它	—	保留。

## 2.2.34 CfgMR2

MR2 为 DDR3 模式寄存器 2；写该寄存器会触发存储控制器向存储器发一条 MRS 命令，配置

DDR3 存储器颗粒的 MR2 寄存器。

表 2-37: CfgMR2 寄存器域的描述

名称	范围	类型	描述
Rtt_WR	[10:9]	RW,0	动态 ODT 端接控制： “0”：关闭动态 ODT； “1”：RZQ/4； “2”：RZQ/2；
MBZ	[8]	RW,0	必需配置为“0”。
SRT	[7]	RW,0	自刷新温度范围。
ASR	[6]	RW,0	自刷新。

CWL	[5:3]	RW,3	CAS 写延迟。配置值，与周期数的对应关系如下： “0”：5； “1”：6； “2”：7； “3”：8； “4”：9； “5”：10； “6”：11； “7”：12。
PASR	[2:0]	RW,0	部分阵列自刷新。
—	其它	—	保留。

### 2.2.35 CfgMR3

MR3 为 DDR3 模式寄存器 3；写该寄存器会触发存储控制器向存储器发一条 MRS 命令，配置

DDR3 存储器颗粒的 MR3 寄存器。

表 2-38: CfgMR3 寄存器域的描述

名称	范围	类型	描述
MPR	[2]	RW,0	MPR。
MPRLoc	[1:0]	RW,0	MPR 地址。
—	其它	—	保留。

### 2.2.36 PERF\_CTRL

该寄存器主要包含存控对课题发出的总请求数、页命中与否计数。

名称	范围	类型	描述
RWCntClr	[2]	RW,0	读写命令计数器清除控制
PerfCtrl	[1:0]	RW,0	存控性能计数器控制信息， 0: 默认值，不做任何操作动作； 1: 表示清除计数器； 2: 表示启动计数器； 3: 表示停止计数器计数；

## 2.2.37 MC\_PERF0

该寄存器

名称	范围	类型	描述
ReqCnt_ALL	[63:32]	RO,0	存控仲裁出去的读写请求数
PDHCnt	[31:0]	RO,0	存控仲裁出去的页直接命中的读写命令数

## 2.2.38 MEMSERR\_CNT

存储控制器会对存储器返回的数据进行 ECC 校验，该寄存器记录 ECC 单错的次数。寄存器 MEMSERR\_TH 控制该寄存器的计数，当指定的时间窗口内单错计数达到阈值，则向系统报错。同时也会对 ECC 多错进行记录。

表 2-39: MEMSERR\_CNT 寄存器域的描述

名称	范围	类型	描述
MEMMErr	[33]	RW1C,0	Mem 读数据多错标志： [33]：表示 Mem 读数据发生 ECC 多错。 软件写该寄存器时，如果数据部分 bit 33 为 1，则清除该寄存器的[33]位。 存储控制器不通知维护接口读主存数据发生多错，不重新生成 ECC，原数据直接返回给目标方（各核心或各设备方），由目标方进行报多错。
CntMEMSErr	[32:0]	RW1C,0	Mem 读数据单错信息： [32]：表示 Mem 读数据发生 ECC 单错预警； [31:0]：表示 Mem 读数据时间窗口内发生 ECC 单错的次数。
			软件写该寄存器时，如果数据部分 bit0 为 1，则清除该寄存器的[31:0]。如果数据部分 bit 32 为 1，则清除该寄存器的[32]
—	其它	—	保留。

## 2.2.39 MEMSERR\_TH

存储控制器会对存储器返回的数据和来自 CPM 的写数据进行 ECC 校验，该寄存器控制 MEMSERR\_CNT 和 WRDATASERR\_CNT 的计数，当指定的时间窗口内单错计数达到阈值，则向维护接口报错。

表 2-40: MEMSERR\_TH 寄存器域的描述

名称	范围	类型	描述
SErrThreshold	[10:6]	RW,0x1F	主存单错计数阈值。
SErrWdsThresholdEn	[5]	RW,1	主存单错时间窗口阈值使能。
SErrWdsThreshold	[4:0]	RW,0x1F	主存单错时间窗口阈值。
—	其它	—	保留。

## 2.2.40 WRDATASERR\_CNT

存储控制器会对来自一致性处理部件的写数据进行 ECC 校验，对读写请求的控制信息进行偶校验，该寄存器记录 ECC 单错的次数，同时记录 ECC 多错和偶校验错标志。寄存器 MEMSERR\_TH 控制该寄存器的计数，当指定的时间窗口内单错计数达到阈值，则向维护接口报错，同时也会对 ECC 多错和偶校验错进行记录，并报给维护接口。

表 2-41: WRDATASERR\_CNT 寄存器域的描述

名称	范围	类型	描述
WRDATAMErr	[34:33]	RW1C,0	CPM 下行通路多错标志： [34]：表示下行异步 FIFO 写数据通路发生 ECC 多错； [33]：表示下行异步 FIFO 控制信息通路发生偶校验错。 软件写该寄存器时，如果数据部分 bit 34 为 1，则清除该寄存器的[34]位。如果数据部分 bit 33 为 1，则清除该寄存器的[33]位。
CntWRDATASErr	[32:0]	RW1C,0	CPM 下行通路单错信息：
			[32]：表示下行通路发生 ECC 单错预警； [31:0]：表示下行通路在时间窗口内发生 ECC 单错的次数。 软件写该寄存器时，如果数据部分 bit 0 为 1，则清除该寄存器的[31:0]。如果数据部分 bit 32 为
—	其它	—	保留。

### 2.2.41 MC\_PERF1

名称	范围	类型	描述
PHCnt	[63:32]	RO,0	存控仲裁出去的页命中的读写命令数
PMCnt	[31:0]	RO,0	存控仲裁出去的页丢失的读写命令数

### 2.2.42 RWCNTOFREF

名称	范围	类型	描述
RdWrCntOfREF_Max	[63:48]	RO,0	一个刷新周期内仲裁的读写命令数的峰值
ConRdCntOfREF_Max	[47:32]	RO,0	一个刷新周期内连续仲裁读命令数的峰值
ConWrCntOfREF_Max	[31:16]	RO,0	一个刷新周期内连续仲裁写命令数的峰值
RdWrTurnCntOfREF_Max	[15:0]	RO,0	一个刷新周期内读写命令切换次数的峰值

## 2.3 存储器地址/命令/数据接口相关 IO 寄存器

AC 和 Datx8 的复位由其内部的 PLL 模块生成，在复位之前，从 IO 寄存器中读出的数据为不定值。复位结束后，PHY 会自动开始初始化过程，该过程完成后，某些寄存器中读到的数据不再是复位值。

### 2.3.1 RIDR

RIDR 是存储器地址/命令/数据接口版本标识寄存器。

表 2-42: RIDR 寄存器域的描述

名称	范围	类型	描述
PHYMJR	[23:20]	RO,0x1	AC/Datx8 (统称 PHY) 大规模修改的标号。
PHYMDR	[19:16]	RO,0x0	AC/Datx8 (统称 PHY) 中规模修改的标号。
PHYMNR	[15:12]	RO,0x0	AC/Datx8 (统称 PHY) 小规模修改的标号。
PUBMJR	[11:8]	RO,0x2	AC/Datx8 的总控部分 (统称 PUB) 大规模修改的标号。
PUBMDR	[7:4]	RO,0x1	AC/Datx8 的总控部分 (统称 PUB) 中规模修改的标号。
PUBMNR	[3:0]	RO,0x0	AC/Datx8 的总控部分 (统称 PUB) 小规模修改的标号。

—	其它	—	保留。
---	----	---	-----

### 2.3.2 PIR

PIR 是存储器 PHY 初始化控制寄存器。配置 PIR 相应的各位，将触发相应的初始化流程，相应位在完成初始流程后进行自清除。

表 2-43: PIR 寄存器域的描述

名称	范围	类型	描述
INITBYP	[31]	RW,0x0	初始化旁路：该位为“1”表示初始化阶段被旁路（包括 PHY 的初始化、DRAM 初始化和 PHY 的 Training）或停止初始化。用户可配置 PIR 寄存器(INIT 位和其它相关位)进
ZCALBYP	[30]	RW,0x0	阻抗校准旁路：该位为“1”表示初始化阶段跳过阻抗校准或停止阻抗校准。用户可配置 PIR 寄存器（INIT 位和 ZCAL 位）进行人工触发。该位会自清除。
DCALBYP	[29]	RW,0x0	DDL 校准 Bypass：该位为“1”表示初始化阶段跳过 DDL 校准或停止 DDL 校准。用户可配置 PIR 寄存器（INIT 位和
LOCKBYP	[28]	RW,0x0	PLL Lock Bypass：该位为“1”表示初始化阶段跳过 PLL 锁定或停止 PLL 锁定。用户可配置 PIR 寄存器（INIT 配置成 1、PLLINIT 配置成 1）进行人工触发。该位会自清
CLRSR	[27]	RW,0x0	清除状态寄存器：对该位写“1”将清除所有的状态寄存器，包括 PGSR 和 DXnGSR；该位会自清除。
CTLDINIT	[18]	RW,0x0	Controller DRAM 初始化：该位为“1”表示 DRAM 初始化由 Controller 完成，否则 DRAM 初始化由 PUB 完成（内
PLLBYP	[17]	RW,0x0	PLL Bypass；该位为“1”，AC 和 DATX8 的 PLL 处于旁路模式。
ICPC	[16]	RW,0x0	初始化完成端口（dfi_init_complete）配置。该位为“0”，该完成端口在完成 PHY 初始化（DLL 锁定且完成阻抗校准）后有效；该位为“1”，该完成端口在完成 PHY 初始化、SDRAM 初始化和 PHY Training 后有效。

WREYE	[15]	RW,0x0	Write Data Eye Training: 该位为“1”表示需要由 PUB 进行 Write Data Eye Training。该位会进行自清除。
RDEYE	[14]	RW,0x0	Read Data Eye Training: 该位为“1”表示需要由 PUB 进行 Read Data Eye Training。该位会进行自清除。
WRDSKW	[13]	RW,0x0	Write Data Bit Deskew: 该位为“1”表示需要由 PUB 进行 Write Data Bit Deskew。该位会进行自清除。
RDDSKW	[12]	RW,0x0	Read Data Bit Deskew: 该位为“1”表示需要由 PUB 进行 Read Data Bit Deskew。该位会进行自清除。
WLADJ	[11]	RW,0x0	Write Leveling Adjust: 该位为“1”表示需要由 PUB 进行 Write Leveling Adjust。该位会进行自清除。
QSGATE	[10]	RW,0x0	Read DQS Gate Training: 该位为“1”表示需要由 PUB 进行 Read DQS Gate Training。该位会进行自清除。
WL	[9]	RW,0x0	Write Leveling: 该位为“1”表示需要由 PUB 进行 Write Leveling。该位会进行自清除。
DRAMINIT	[8]	RW,0x0	DRAM 初始化: 该位为“1”表示需要进行 SDRAM 初始化。该位会进行自清除。
DRAMRST	[7]	RW,0x0	DRAM 复位: 该位为“1”表示需要进行 DRAM 复位 (仅限于 DDR3, 驱动复位信号), 且维持 200us。可单独触发复位或者作为 SDRAM 初始化过程的一部分, 对于后者 Reset 信号要保持 200us 后才可以开始 SDRAM 初始化流
PHYRST	[6]	RW,0x0	PHY 复位: 该位为“1”表示要复位 AC 和 DATX8 (驱动 AC 和 DATX8 的复位信号)。该位会进行自清除。
DCAL	[5]	RW,0x0	DDL 校准: 该位为“1”表示需要进行 DDL 校准。该位会进行自清除。
PLLINT	[4]	RW,0x0	PLL 初始化: 该位为“1”表示需要进行 PLL 初始化。该过程包括正确的驱动 Power-Down、Reset 和 gear shift
ZCAL	[1]	RW,0x0	阻抗校准: 该位为“1”表示需要进行输出阻抗和输入端接电阻校准。该步和 PHY 的初始化(PLL 初始化+DDL 校准+PHY 的复位)同步进行。该位会进行自清除。

INIT	[0]	RW,0x0	初始化触发位：对该位写“1”则触发初始化，包括 PHY 初始化、DRAM 初始化和 PHY Training。具体的初始化步骤根据[15:1]位进行独立使能（不使能则被跳过）。该
—	其它	—	保留。

### 2.3.3 PGCR0-1

PGCR0-1 是 PUB 的通用控制寄存器。

表 2-44: PGCR0 寄存器域的描述

名称	范围	类型	描述
CKEN	[31:26]	RW,0x2a	CK 使能，用于控制给 SDRAM 的 SDR_CLK 时钟。每个 CK 两位控制，具体含义如下： 2'b00: 保持为 0； 2' 01: 反向输出； 2'b10: 正常输出； 2'b11: 保持为 1。
DTOSEL	[18:14]	RW,0x0	数字测试输出（phy_dto）控制选择； 5'b00000: DATX8 0 PLL 数字测试输出； 5'b00001: DATX8 1 PLL 数字测试输出； 5'b00010: DATX8 2 PLL 数字测试输出； 5'b00011: DATX8 3 PLL 数字测试输出； 5'b00100: DATX8 4 PLL 数字测试输出； 5'b00101: DATX8 5 PLL 数字测试输出； 5'b00110: DATX8 6 PLL 数字测试输出； 5'b00111: DATX8 7 PLL 数字测试输出； 5'b01000: DATX8 8 PLL 数字测试输出； 5'b01001: AC PLL 数字测试输出； 5'b01010~5'b01111: 保留； 5'b10000: DATX8 0 DDL 数字测试输出； 5'b10001: DATX8 1 DDL 数字测试输出；



			5'b10010: DATX8 2 DDL 数字测试输出; 5'b10011: DATX8 3 DDL 数字测试输出; 5'b10100: DATX8 4 DDL 数字测试输出; 5'b10101: DATX8 5 DDL 数字测试输出; 5'b10110: DATX8 6 DDL 数字测试输出; 5'b10111: DATX8 7 DDL 数字测试输出; 5'b11000: DATX8 8 DDL 数字测试输出; 5'b11001: AC PLL 数字测试输出; 5'b11010~5'b11111: 保留。
OSCWDL	[13:12]	RW,0x3	DLL 环振测试模式 Write-Leveling Delay Line 选择, 处于 inactive 的 Delay Line 的 delay value 设置为零。 2'b00: no WL LDCL is active; 2'b01: DDR WL LCDL is active; 2'b10: SDR WL LCDL is active; 2'b11: Both WL LCDL is active。
OSCDIV	[11:9]	RW,0x7	DLL 环振测试模式分频输出。 0: 1 分频; 1: 256 分频; 2: 512 分频; 3: 1024 分频; 4: 2048 分频; 5: 4096 分频; 6: 8192 分频; 7: 65536 分频。
OSCEN	[8]	RW,0x0	环振模式使能。
DLTST	[7]	RW,0x0	Delay Line Test Start, 对该位写“1”表示触发 DLL 环振测试。
DLTMODE	[6]	RW,0x0	Delay Line Test Mode, 该位为“1”表示处于 DLL 环振测试模式。
RDBVT	[5]	RW,0x1	Read Data BDL VT 补偿使能。 对应
WDBVT	[4]	RW,0x1	Write Data BDL VT 补偿使能。 对应
RGLVT	[3]	RW,0x1	Read DQS Gating LCDL VT 补偿使能。

			对应 DXnLCDLR2[RxDQSD]。
RDLVT	[2]	RW,0x1	Read DQS LCDL VT 补偿使能。 对应
WDLVT	[1]	RW,0x1	Write DQ LCDL VT 补偿使能。 对应
WLLVT	[0]	RW,0x1	Write Leveling LCDL VT 补偿使能。 对应
—	其它	—	保留。

表 2-45: PGCR1 寄存器域的描述

名称	范围	类型	描述
LBMODE	[31]	RW,0x0	Loopback Mode: 该位为“1”表示处于 LoopBack 测试模式。
LBGDQS	[30:29]	RW,0x0	Loopback DQS Gating 选择。 2'b00: DQS Gate 总是开启; 2'b01: DQS Gate 由 PUB 触发; 2'b10: DQS Gate 由软件触发;
LBDQSS	[28]	RW,0x0	LoopBack DQS Shift 选择。 0: PUB 把 Read DQS LCDL 设置为 0, DQS 已经被 Write Path 移位 90 度; 1: Read DQS 移位通过软件配置。
IOLB	[27]	RW,0x0	IO Loop-Back 选择。 0: 在 output buffer 之后 (pad) 环回, 必须控制输出使能; 1: 在 output buffer 之前 (core sice) 环回, 可以不关
INHVT	[26]	RW,0x0	禁止 VT 调节, 该位写“1”则禁止 VT 调节。在直接配置 delay Line 寄存器时应该配置该位为“1”。
PHYHRST	[25]	RW,0x1	PHY 高速逻辑复位; 对该位写“0”会复位 AC 和 DTAX8

ZCKSEL	[24:23]	RW,0x2	电阻时钟驱动选择；用于选择用于阻抗校准控制逻辑的时钟和 CFG_CLK 的频率关系； 0: 1 分频； 1: 8 分频； 2: 32 分频； 3: 64 分频。
DLDLMT	[22:15]	RW,0x1	延时线上 PVT 调节阈值（单个方向上），超出这个阈值将启动 VT 调节。该位设置为零，则 Disable PVT 调节。
FDEPTH	[14:13]	RW,0x2	滤波深度，指定初始化期间或 DCAL 被触发后周期测量时的滤波深度。 0: 2； 1: 4； 2: 8； 3: 16。
LPFDEPTH	[12:11]	RW,0x0	低通滤波深度，指定 MDL 周期测量时的滤波深度。 0: 2； 1: 4； 2: 8； 3: 16。
LPFEN	[10]	RW,0x1	低通滤波使能。该位为“1”，则 MDL 周期测量的低通滤波使能。
MDLEN	[9]	RW,0x0	Master Delay Line 使能。该位为“1”，AC 的 Master delay line 在初始周期测量后会执行后续的周期测
IODDRM	[8:7]	RW,0x0	IO DDR Mode。
WLSELT	[6]	RW,0x0	Write Leveling Select Type: 0: 90 度的 Setup margin、90 度的 hold margin； 1: 135 度的 Setup margin、45 度的
WLRANK	[5:4]	RW,0x3	Write Level Rank，该位表示该存控支持的 Rank 数； 2'b00= 1 Rank； 2'b01= 2 Rank； 2'b10= 3 Rank； 2'b11= 4 Rank。

WLSTEP	[2]	RW,0x0	Write Leveling 步长控制。 0: 32 step; 1: 1
WLFULL	[1]	RW,0x1	Write Leveling Full: 该位为“1”，完整的 Write Leveling 算法被执行（完整的 Write Leveling 包括遍历所有 Rank 的 DDRAM 的设置）
PDDISDX	[0]	RW,0x1	Power-Down Disable Byte Enable: 该位为“1”，被 Disable 的字节的 DLL 和 IO 会 PowerDown。
—	其它	—	保留。

### 2.3.4 PGSR0-1

PGSR0-1 是 PUB 的通用状态寄存器。

表 2-46: PGSR0 寄存器域的描述

名称	范围	类型	描述
APLOCK	[31]	RO	AC PLL 锁定信号。
DTERR	[30:28]	RO	Data Training 错误码，反映出错步骤。
WEERR	[27]	RO	Write Eye Training 错误。
REERR	[26]	RO	Read Eye Training 错误。
WDERR	[25]	RO	Write Bit Deskew 错误。
RDERR	[24]	RO	Read Bit Deskew 错误。
WLAERR	[23]	RO	Write Leveling PipeLine 调节错误。
QSGERR	[22]	RO	DQS Gate 训练错误。
WLERR	[21]	RO	Write Leveling 错误。
DIERR	[20]	RO	DRAM 初始化错误。
ZCERR	[19]	RO	阻抗校准错误。
DCERR	[18]	RO	DLL 校准错误。
PLERR	[17]	RO	PLL 锁定错误。
IERR	[16]	RO	初始化错误。
WEDONE	[11]	RO	Write Eye Training 完成。
REDONE	[10]	RO	Read Eye Training 完成。
WDDONE	[9]	RO	Write Data Deskew 完成。
RDDONE	[8]	RO	Read Data Deskew 完成。
WLADONE	[7]	RO	Write Leveling PipeLine 完成。

QSGDON	[6]	RO	DQS Gate 训练完成。
WLDONE	[5]	RO	Write Leveling 完成。
DIDONE	[4]	RO	DRAM 初始化完成。
ZCDONE	[3]	RO	阻抗校准完成。
DCDONE	[2]	RO	DLL 校准完成。
PLDONE	[1]	RO	PLL 锁定完成。
IDONE	[0]	RO	初始化完成。
—	其它	—	保留。

表 2-47: PGSR1 寄存器域的描述

名称	范围	类型	描述
DLTCODE	[24:1]	RO	延迟线测试结果，返回 AC 延迟线测量（DLL 环振测试）得到的周期数。
DLTDONE	[0]	RO	延迟线测试结束，表示已经完成了 AC 的延迟线测量（DLL 环振测试）。
—	其它	—	保留。

### 2.3.5 PLLCR

PLLCR 是 PLL 控制寄存器。

表 2-48: PLLCR 寄存器域的描述

名称	范围	类型	描述
BYP	[31]	RW,0x0	PLL 旁路。
PLLRST	[30]	RW,0x0	PLL 复位，会驱动 pll_rst。对该写“1”将复位 pll。该位不会自清除，需要写零结束复位。
PLLPD	[29]	RW,0x0	PLL Power Down，会驱动 pll_pwrdsn。对该写“1”将 Power Down PLL。该位不会被自清除，需要写零退出。

FRQSEL	[19:18]	RW,0x0	<p>PLL 频率选择, 根据 PHY 的不同, 该位信息配置有所不同。</p> <p>对于可以工作在 2133Mbps 的 PHY:</p> <p>2'b00: PLL 参考时钟在 335MHz~533MHz;</p> <p>2'b00: PLL 参考时钟在 225MHz~385MHz;</p> <p>2'b10: 保留;</p> <p>2'b11: PLL 参考时钟在 166MHz~275MHz。对于不可以工作在 2133Mbps 的 PHY:</p>
			<p>2'b00: PLL 参考时钟在 166MHz~300MHz;</p> <p>2'b10: 保留;</p> <p>2'b11: 保留。</p>
CPPC	[16:13]	RW,0xe	Charge Pump proportional Current Control, 调节压控振荡器的控制电流。
CPIC	[12:11]	RW,0x0	Charge Pump integrating Current Control, 调节压控振荡器的控制电流。
GSHIFT	[10]	RW,0x0	Gear Shift 使能, 如果使能则快速锁定模式。
ATOEN	[9:6]	RW,0x0	<p>PLL 模拟测试使能。</p> <p>4'b0000: ALL PLL 模拟测试输出为三态;</p> <p>4'b0001: AC PLL 模拟测试输出;</p> <p>4'b0010: DATX8 0 PLL 模拟测试输出;</p> <p>4'b0011: DATX8 1 PLL 模拟测试输出;</p> <p>4'b0100: DATX8 2 PLL 模拟测试输出;</p> <p>4'b0101: DATX8 3 PLL 模拟测试输出;</p> <p>4'b0110: DATX8 4 PLL 模拟测试输出;</p> <p>4'b0111: DATX8 5 PLL 模拟测试输出;</p> <p>4'b1000: DATX8 6 PLL 模拟测试输出;</p> <p>4'b1001: DATX8 7 PLL 模拟测试输出;</p> <p>4'b1010: DATX8 8 PLL 模拟测试输出;</p> <p>4'b1011~4'b1111: 保留。PLL 数字测试使能由 PGCR0[DTSEL]进行控制。</p>

ATC	[5:2]	RW,0x0	PLL 模拟测试控制选择（通过 pll_ato 输出）； 4'b0000：保留； 4'b0001：vdd_ckin； 4'b0010：rbbf； 4'b0011：vdd_cko； 4'b0100：vp_cp； 4'b0101：vpfil； 4'b0111：gd； 4'b1000： vcntrl_atb； 4'b1001： vref_atb； 4'b1010： vpsf_atb； 4'b1011~4'b1111：保留。
DTC	[1:0]	RW,0x0	PLL 数字测试控制选择（通过 Pll_dto[1]输出） 2'b00：不使能，输出为零； 2'b01：PLL x1 clock； 2'b10：PLL 参考时钟； 2'b11：PLL feedback clock； 通过 PGCR0[DTOSEL]选择 AC 或 DATX8 的 PLL 或 DDL 测试输出到 DTO[1:0]。对于 DDL 测试只有 DTO[0]有意义。对于 PLL 测试，两位都有意义，且 DTO[0]为反馈时钟，DTO[1]根据 DTC 可控制。
—	其它	—	保留。

### 2.3.6 PTR0-4

PTR 是延时控制寄存器。

表 2-49：PTR0 寄存器域的描述

名称	范围	类型	描述
tPLLPD	[31:21]	RW, 0x190	PLL Power Down 时间。在 Power Down 状态维持的时间，即 CFG 复位结束到 PLL PowerDown 结束的时间。该时间要大于等于 1us。默认值是 25MHz（CFG 时钟）

tPLLGS	[20:6]	RW, 0x7D0	PLL Gear Shift Time。从 PLL Reset pin 无效到 PLL gear Shift pin 无效的时间。该时间要大于等于 40us，默认
tPHYRST	[5:0]	RW, 0x10	PHY Rest Time。PHY Reset 在结束 PHY 校准之后复位信号必须维持的时间。默认值是 25MHz（CFG 时钟）的 400ns。
—	其它	—	保留。

表 2-50: PTR1 寄存器域的描述

名称	范围	类型	描述
tPLLLOCK	[31:16]	RW, 0x2ee0	PLL 锁定时间。从 PLL 复位结束到 PLL 锁定的时间。该时间要大于等于 30us。默认值是 25MHz（CFG 时钟）的 56us。
tPLLRST	[12:0]	RW, 0x578	PLL 复位时间；从 PLL PowerDown 结束到 PLL 复位结束的时间。该时间要大于等于 3.5us。默认值是
—	其它	—	保留。

表 2-51: PTR2 寄存器域的描述

名称	范围	类型	描述
tWLDLYS	[19:15]	RW,0x10	Write Leveling Delay Setting Time。Write Leveling Delay 设置到 DQS 为高的节拍数（SDR 时钟）。
tCALH	[14:10]	RW,0xf	Calibration Hold Time: Cal_en 有效到 Cal_clk_en 有效的节拍数（SDR 时钟）。
tCALS	[9:5]	RW,0xf	Calibration Setup Time: Cal_clk_en 无效到 cal_en 有效的节拍数（SDR 时钟）。
tCALON	[4:0]	RW,0xf	Calibration ON Time: Cal_clk_en 有效的节拍数（SDR 时钟）。
—	其它	—	保留。

表 2-52: PTR3 寄存器域的描述

名称	范围	类型	描述
tDINIT1	[28:20]	RW, 384	SDRAM 初始化流程，CKE 为高到第一个命令的时间，默认值是 800Mhz（DDR3-1600）的 480ns。



tDINIT0	[19:0]	RW, 533334	SDRAM 初始化流程, Reset 无效但 CKE 仍需要为低的时间, 默认值是 800Mhz (DDR3-1600) 的
—	其它	—	保留。

表 2-53: PTR4 寄存器域的描述

名称	范围	类型	描述
tDINIT3	[27:18]	RW, 683	SDRAM 初始化流程, ZQ 命令到第一个命令的时间, 默认值是 800Mhz (DDR3-1600) 的 853ns。
tDINIT2	[17:0]	RW, 213334	SDRAM 初始化流程, Rest 维护有效的的时间, 默认值是 800Mhz (DDR3-1600) 的 266us。
—	其它	—	保留。

### 2.3.7 ACMDLR

ACMDLR 为地址和命令接口的主延迟线 (Master Delay Line)。

表 2-54: ACMDLR 寄存器域的描述

名称	范围	类型	描述
MDLD	[23:16]	RW,0x0	MDL 延迟: 采用折半查找的方法, 对 DDR 周期进行测量, 该配置用于指定测量过程中 MDL 延迟值。
TPRD	[15:8]	RW,0x0	目标周期值。主延迟线周期测量值 (主延迟线更新功能使能时, 该值会周期性的更新)。该值用于延迟线校准, 用作比例系数的分子。其它延时线 (BDL) 根据该调节因子 (TPRD/IPRD) 去计算延时线更新后的值, 如果偏差超
IPRD	[7:0]	RW,0x0	初始周期值。主延迟线初始测量得到的周期值。该值用于延迟线校准, 用作比例系数的分母。
—	其它	—	保留。

### 2.3.8 ACBDLR

ACBDLR 为地址和命令接口的位延迟线 (Bit Delay Line)。

表 2-55: ACBDLR 寄存器域的描述

名称	范围	类型	描述
----	----	----	----

ACBD	[23:18]	RW,0x0	AC Bit Delay。
CK2BD	[17:12]	RW,0x0	CK2 Bit Delay。
CK1BD	[11:6]	RW,0x0	CK1 Bit Delay。
CK0BD	[5:0]	RW,0x0	CK0 Bit Delay。
—	其它	—	保留。

### 2.3.9 ACIOCR

ACIOCR 为地址和命令接口的 IO 配置寄存器。

表 2-56: ACIOCR 寄存器域的描述

名称	范围	类型	描述
ACSR	[31:30]	RW,0x0	AC Slew Rate 选择。 2'b00: 非常快速 (DDR3 的默认配置) ; 2'b01: 快速; 2'b10: 中速; 2'b11: 慢速。
RSTIOM	[29]	RW,0x1	SDRAM Reset IO Mode 选择。 0: 选择 SSTL; 1: 选择

RSTPDR	[28]	RW,0x1	SDRM Reset Power Down Receiver。如果设置为“1”，Reset_N 输入驱动处于 Power-Down。
RSTPDD	[27]	RW,0x0	SDRM Reset Power Down Driver。如果设置为“1”，Reset_N 输出驱动处于 Power-Down。
RSTODT	[26]	RW,0x0	SDRM Reset ODT 控制。如果设置为“1”，则 Reset_N 设置端接电阻。
RANKPDR	[25:22]	RW,0xf	RANK Power Down Receiver。如果设置为“1”，则 CKE[3:0]、ODT[3:0]、CS[3:0]输入驱动处于 Power-Down，其中 RANKPDR[0]对应 Rank0，RANKPDR[1]对应 Rank1，
CSPDD	[21:18]	RW,0x0	CS Power Down Driver。如果设置为“1”，则 CS[3:0]输出驱动处于 Power-Down。
RANKODT	[17:14]	RW,0x0	Rank ODT 控制。如果设置为“1”，则 CKE[3:0]、ODT[3:0]、CS[3:0]设置端接电阻，其中 RANKODT[0]对应 Rank0，RANKODT[1]对应 Rank1，依次类推。
CKPDR	[13:11]	RW,0x7	CK Power Down Receiver。如果设置为“1”，CK[2:0]输入驱动处于 Power-Down。
CKPDD	[10:8]	RW,0x0	CK Power Down Driver。如果设置为“1”，CK[2:0]输出驱动处于 Power-Down。
CKODT	[7:5]	RW,0x0	CK ODT 控制。如果设置为“1”，则 CK[2:0]设置端接电阻。
ACPDR	[4]	RW,0x1	AC Power Down Receiver。如果设置为“1”，RAS_N、CAS_N、WE_N、BA[2:0]和 A[15:0]输入驱动处于 Power-Down。
ACPDD	[3]	RW,0x0	AC Power Down Driver。如果设置为“1”，RAS_N、CAS_N、WE_N、BA[2:0]和 A[15:0]输出驱动处于 Power-Down。
ACODT	[2]	RW,0x0	AC ODT 控制。如果设置为“1”，则 RAS_N、CAS_N、WE_N、BA[2:0]和 A[15:0]设置端接电阻。
ACOE	[1]	RW,0x1	AC 输出使能。
ACIOM	[0]	RW,0x0	AC IO Mode 选择。 0: 选择 SSTL; 1: 选择 CMOS。
—	其它	—	保留。

## 2.3.10 DXCCR

DXCCR 是 DatX8 部分的通用配置寄存器。

表 2-57: DXCCR 寄存器域的描述

名称	范围	类型	描述
DXSR	[14:13]	RW,0x0	该位控制所有 DATX8 中的 DQ、DM、DQS、DQS_N 的 Slew rate。 2'b00: 非常快速 (DDR3 的默认配置); 2'b01: 快速; 2'b10: 中速; 2'b11: 慢速。
DQSNRES	[12:9]	RW,0xc	对 DQS# 的上拉、下拉电阻调节。 [3]: 0 表示上拉电阻调节, 1 表示下拉电阻调节。 [2:0] 表示具体值, 具体如下: 0: 没有片上电阻; 1: 688 欧姆; 2: 611 欧姆; 3: 550 欧姆; 4: 500 欧姆; 5: 458 欧姆; 6: 393 欧姆; 7: 344 欧姆。
DQSRES	[8:5]	RW,0x4	对 DQS 的上拉、下拉电阻调节。 [3]: 0 表示上拉电阻调节, 1 表示下拉电阻调节。 [2:0] 表示具体值, 具体如下: 0: 没有片上电阻; 1: 688 欧姆; 2: 611 欧姆; 3: 550 欧姆; 4: 500 欧姆; 5: 458 欧姆; 6: 393 欧姆; 7: 344 欧姆。

DXPDR	[4]	RW,0x0	Data Power Down Receiver。该位控制所有 DATX8 中的 DQ、DM、DQS、DQS_N；该位和每个独立的 DATX8 中的 IOM 配置信息共同起作用（与 DXnGCR[DXPDR] “或”
DXPDD	[3]	RW,0x0	Data Power Down Driver。该位控制所有 DATX8 中的 DQ、DM、DQS、DQS_N；该位和每个独立的 DATX8 中的 IOM。
MDLEN	[2]	RW,0x1	Master Delay Line 使能。该位控制所有 DATX8 的 Master delay Line 校准；该位和每个独立的 DATX8 中的 IOM 配置信息共同起作用（与
DXIOM	[1]	RW,0x0	Data IO Mode 选择。该位控制所有 DATX8 中的 DQ、DM、DQS、DQS_N。 0: 选择 SSTL； 1: 选择 CMOS。该位和每个独立的 DATX8 中的 IOM 配置信息共同起作用（与 DXnGCR[DXIOM] “或” 操作）。
DXODT	[0]	RW,0x0	Data ODT 控制。该位控制所有 DATX8 中的 DQ、DM、DQS、DQS_N 的 ODT。该位和每个独立的 DATX8 中的 ODT 配置
—	其它	—	保留。

### 2.3.11 DSGCR

DSGCR 是 DDR System 通用控制寄存器。

表 2-58: DSGCR 寄存器域的描述

名称	范围	类型	描述
CKEOT	[31]	RW,0x1	CKE 输出使能。
RSTOE	[30]	RW,0x1	Reset_N 输出使能。
ODTOE	[29]	RW,0x1	ODT 输出使能。
CKOE	[28]	RW,0x1	CK 输出使能。
ODTPDD	[27:24]	RW,0x0	ODT Power Down Driver, 用于关闭 ODT 引脚的输出功
CKEPDD	[23:20]	RW,0x0	CKE Power Down Driver, 用于关闭 CKE 引脚的输出功

SDRMODE	[19]	RW,0x0	SDR 模式或 HDR 模式（申威 411 处理器使用 HDR 模式）。0：表示 HDR 模式；1：表示 SDR 模式。
RRMODE	[18]	RW,0x0	Rist-to-Rise 模式。
DQSGX	[6]	RW,0x0	DQS Gate 左右各延长 1 拍。
CUAEN	[5]	RW,0x0	Controller update Acknowledge 使能。该位为“1”，PHY 应该对来自 controller 的 Update 请求予以确认信号。
LPPLLPD	[4]	RW,0x1	Low Power PLL Power Down。与 DFI Low Power 有关；如果设置为“1”，则要响应 DFI low power 请求，使 PLL 处
LPIOPD	[3]	RW,0x1	Low Power IO Power Down。与 DFI Low Power 有关，如果设置为“1”，则要响应 DFI low power 请求，使 IO 处于
ZUEN	[2]	RW,0x1	阻抗 Update 使能。该位为“1”，当发起的 Update 请求时，
BDisEN	[1]	RW,0x1	Byte Disable Enable。该位为“1”，PHY 会响应 controller 的 Byte Disable 请求，否则被忽略，此时只能通过 DXnGCR
PUREN	[0]	RW,0x1	PHY Update 请求使能。如果使能，当 DDL VT 漂移时，PHY
—	其它	—	保留。

### 2.3.12 DCR

DCR 是 DRAM 配置寄存器。

表 2-59: DCR 寄存器域的描述

名称	范围	类型	描述
RDIMM	[30]	RW,0x0	RDIMM 指示。
UDIMM	[29]	RW,0x0	UDIMM 地址镜像指示。
DDR2T	[28]	RW,0x0	2T 模式指示。
NOSRA	[27]	RW,0x0	该位为“1”，则同一拍命令不能同时发射到两个 Rank 上。对于带镜像的双 Rank UDIMM，该位必须设置为
BYTEMASK	[17:10]	RW,0x1	在 DQS Gate 训练时，选择相应的 bit，MPR 操作时该位要和 PDQ 一致。

MPRDQ	[7]	RW,0x0	MPR 模式读时，指定非主要 DQ pin 上的值。 0：表示非主要 DQ 上是 0； 1：表示非主要 DQ 上与主要 DQ 上一致。
-------	-----	--------	------------------------------------------------------------------------

PDQ	[6:4]	RW,0x0	MPR 模式读时, 指定主要 DQ 管脚, 即从 DQ[7:0]中进行 8
DDR8BANK	[3]	RW,0x1	8 Bank 指示。
DDRMD	[2:0]	RW,0x3	SDRAM 类型指示。 3'b000: LPDDR; 3'b001: DDR; 3'b010: DDR2; 3'b011: DDR; 3'b100: LPDDR2;
—	其它	—	保留。

### 2.3.13 DTPR0-2

DTPR 是 DRAM 延时参数寄存器。

表 2-60: DTPR0 寄存器域的描述

名称	范围	类型	描述
tRC	[31:26]	RW,0x18	ACT 命令到 ACT 命令之间的间隔(同 Bank), 有效取值是 0x2~0x2a。 根据 DDR3-1600 要求, 需要修改为 0x27。
tRRD	[25:22]	RW,0x4	ACT 命令到 ACT 命令之间的间隔(不同 Bank), 有效取值是 1~8。 根据 DDR3-1600 要求, 需要修改为 6。
tRAS	[21:16]	RW,0x12	ACT 命令到 PRE 命令之间的间隔, 有效取值是 0x2~0x1f。 根据 DDR3-1600 要求, 需要修改为 0x1c。
tRCD	[15:12]	RW,0x6	ACT 命令到 CAS 命令之间的间隔, 有效取值是 2~11。 根据 DDR3-1600 要求, 需要修改为 9。
tRP	[11:8]	RW,0x6	PRE 命令到 ACT 命令的间隔, 有效取值是 2~11。 根据 DDR3-1600 要求, 需要修改为 9。
tWTR	[7:4]	RW,0x3	WR 命令到 RD 命令之间的间隔, 有效取值是 1~6。 根据 DDR3-1600 要求, 需要修改为 6。
tRTP	[3:0]	RW,0x3	RD 命令到 PRE 命令之间的间隔, 有效取值是 2~6。 根据 DDR3-1600 要求, 需要修改为 6。
—	其它	—	保留。

表 2-61: DTPR1 寄存器域的描述



名称	范围	类型	描述
----	----	----	----



tWLO	[29:26]	RW,0x6	Write Leveling 输出延时，即从 DQS 的上升沿到通过 DQ 返回的时间。
tWLMRD	[25:20]	RW,0x28	Write Leveling Mode 到第 1 个 DQS 上升沿的间隔。
tRFC	[19:11]	RW,83	Refresh-to-Refresh 或 Refresh-to-active 的最小时间。根据 DDR3-1600 和相应的颗粒类型修改该
tFAW	[10:5]	RW,18	四个 ACT 命令的窗口，有效取值 2~31。根据 DDR3-1600 要求，需要修
tMOD	[4:2]	RW,0x0	MRS 命令与后续指令(非 MRS)之间的间隔； 0: 12; 1: 13; 2: 14; 3: 15; 4: 16; 5: 17; 其它保留。
tMRD	[1:0]	RW,0x2	两个 MRS 命令之间的间隔，实际延时值是该配置值加
—	其它	—	保留。

表 2-62: DTPR2 寄存器域的描述

名称	范围	类型	描述
tCCD	[31]	RW,0x0	0 表示 4 拍；1 表示 5 拍。
tRTW	[30]	RW,0x0	Read to Write 的总线切换间隔。0: 标准的读写切换时间；
tRTODT	[29]	RW,0x0	Read ODT Delay: 0: ODT 在 Read 后导后立即开启； 1: ODT 在 Read 后导的下一个时钟才可以开启。 如果 ODT 使能，且该值配置为“1”，读到写的延时需要加 1。
tDLLK	[28:19]	RW, 512	DDL 锁定时间，有效取值 2~1023。
tCKE	[18:15]	RW,0x3	CKE 的最小脉冲宽度，有效取值 2~15。根据 DDR3-1600 要求，需要修
tXP	[14:10]	RW,0x8	Power Down 退出延时，有效取值 2~31。

tXS	[9:0]	RW, 200	Self Refresh 退出的延时, 有效取值 2~1023。根据 DDR3-1600 要求, 需要修改
—	其它	—	保留。

### 2.3.14 MR0-3

MR 是 SDRAM 的模式寄存器。

表 2-63: MR0 寄存器域的描述

名称	范围	类型	描述
RSVD	[15:13]	RW,0x0	JEDEC 保留(零);
PD	[12]	RW,0x0	Power Down Control: 0: Slow Exit (DLL off) ; 1: Fast Exit (DLL on) 。
WR	[11:9]	RW,0x5	Write Recovery 延 时: 0: 16; 1: 5; 2: 6; 3: 7; 4: 8; 5: 10; 6: 12; 7: 14。
DR	[8]	RW,0x0	DLL Reset; 该位自清除;
TM	[7]	RW,0x0	0: 普通模式; 1: 测试模式;

CL	[6:4],[2]	RW,0xa	读延时: 2: 5; 4: 6; 6: 7; 8: 8; 10: 9; 12: 10; 14: 11; 1: 12; 3: 13;
----	-----------	--------	----------------------------------------------------------------------------------------------

			5: 14。
BT	[3]	RW,0x0	突发类型（申威 411 处理器要求该位配置为“1”）： 0: 顺序模式；1: Wrap 模式。
BL	[1:0]	RW,0x2	突发长度： 0: 表示固定 8； 1: 表示 4 或 8； 2: 表示固定 4； 3: 保留。申威 411 处理器要求该位配置为“0”。
—	其它	—	保留。

表 2-64: MR1 寄存器域的描述

名称	范围	类型	描述
RSVD	[15:13]	RW,0x0	JEDEC 保留(零)；
QOFF	[12]	RW,0x0	输出使能。 0=所有输出功能正常； 1=所有 SDRAM 输出关闭。
TDQS	[11]	RW,0x0	DQS 端接。 1=提供附加的端接电阻输出； 0=没有附加端接电阻。
RSVD	[10],[8]	RW,0x0	为 JEDEC 保留(零)。
LEVEL	[7]	RW,0x0	Write Leveling 使能信号。置“1”表示 SDRAM 进行 Write Leveling。
AL	[4:3]	RW,0x0	Posted CAS 附加延迟。允许 Read 和 Write 命令比正常更早的让 SDRAM 执行。 2'b00=不使能； 2'b01=CL-1； 2'b10=CL-2； 2'b11=保留。

RTT	[9],[6],[2]	RW,0x0	片上端接：为 SDRAM 选择有效的电阻。 3'b000=无效； 3'b001 = RZQ/4； 3'b010 = RZQ/2； 3'b011 = RZQ/6； 3'b100 = RZQ/12； 3'b101 = RZQ/8； 其它值保留。
DIC	[5],[1]	RW,0x0	输出驱动阻抗控制。 2'b00=RZQ/6； 2'b01=RZQ/7； 2'b10~2'b11=保留。
DE	[0]	RW,0x0	DLL 使能： 1=不使能； 0=使能。
—	其它	—	保留。

表 2-65: MR2 寄存器域的描述

名称	范围	类型	描述
RSVD	[15:11],[7]	RW,0x0	JEDEC 保留(零)。
RTTWR	[10:9]	RW,0x0	动态 ODT 选择。 2'b00 = 关闭动态 ODT； 2'b01 = RZQ/4； 2'b10 = RZQ/2； 2'b11 = 保留。
ASR	[6]	RW,0x0	该位为“1”表示 SDRAM 根据工作温度自动提供 Self-Refresh 电源管理，否则必须根据温度范围通过 SRT 设置。

CWL	[5:3]	RW,0x0	CAS 写延迟, 从 SDRAM 寄存器写命令到写数据有效的周期数。 3'b000 = 5 ( $t_{CK} > 2.5ns$ ); 3'b001 = 6 ( $2.5ns > t_{CK} > 1.875ns$ ); 3'b010 = 7 ( $1.875ns > t_{CK} > 1.5ns$ ); 3'b011 = 8 ( $1.5ns > t_{CK} > 1.25ns$ ); 3'b100 = 9 ( $1.25ns > t_{CK} > 1.07ns$ );
-----	-------	--------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------



			3'b101=10 (1.07ns > tCK > 0.935ns); 3'b110=11 (0.935ns > tCK > 0.833ns); 3'b111=12 (0.833ns > tCK > 0.75ns);
PARS	[2:0]	RW,0x0	指定部分阵列进行 Self-Refresh。进入 Self Refresh 后，除指定区域外，其它区域的数据将被丢失。  4 Bank 有效取值： 3'b000 = Full Array; 3'b001 = Half Array (BA[1]=0); 3'b010 = Quarter Array (BA[1:0]=2'b00); 3'b011 = 没有定义； 3'b100 = 3/4 Array (BA[1:0]=2'b00 除外); 3'b101 = Half Array (BA[1]=1); 3'b110 = Quarter Array (BA[1:0]=2'b11); 3'b111 = 没有定义。  8 bank 有效取值： 3'b000 = Full Array; 3'b001 = Half Array (BA[2]=0); 3'b010 = Quarter Array (BA[2:1]=2'b00); 3'b011 = 1/8 Array (BA[2:0]=3'b000); 3'b100 = 3/4 Array (BA[2:1]=2'b00 除外); 3'b101 = Half Array (BA[2]=1); 3'b110 = Quarter Array (BA[2:1]=2'b11); 3'b111 = 1/8 Array (BA[2:0]=3'b111);
—	其它	—	保留。

表 2-66: MR3 寄存器域的描述

名称	范围	类型	描述
RSVD	[15:3]	RW,0x0	JEDEC 保留(零)。
MPR	[2]	RW,0x0	多用途寄存器 (MPR) 使能信号。置“1”表示从 MPR 中读取数据，否则从 DRAM 阵列读取数据。
MPRLOC	[1:0]	RW,0x0	多用途寄存器 (MPR) 位置，选择 MPR 数据的位置。 00=为系统校准按预定义模式； 其它值不能使用。
—	其它	—	保留。

## 2.3.15 ODTCR

ODTCR 为 ODT 控制寄存器。

表 2-67: ODTCR 寄存器域的描述

名称	范围	类型	描述
WRODT3	[31:28]	RW,0x8	写 ODT: 当一个 Write 命令发送到某 RANK 时, 是否将各 RANK 的 ODT 使能。 WRODT0~3 分别对 RANK0~3 执行 Write 命令时对各 RANK 的 ODT 的设置。 每个域四个 bit 分别对应四个 RANK。 缺省值时在 Write 命令时只对正在执行 Write 的 RANK 的 ODT 使能;
WRODT2	[27:24]	RW,0x4	
WRODT1	[23:20]	RW,0x2	
WRODT0	[19:16]	RW,0x1	
RDODT3	[15:12]	RW,0x0	读 ODT: 当一个 Read 命令发送到某 RANK 时, 是否将 RANK 的 ODT 使能。 RDODT0~3 分别对 RANK0~3 执行 Read 命令时对各 RANK 的 ODT 的设置。 每个域四个 bit 分别对应四个 RANK。 缺省值时在 Read 命令时全部 ODT 不使能。
RDODT2	[11:8]	RW,0x0	
RDODT1	[7:4]	RW,0x0	
RDODT0	[3:0]	RW,0x0	

## 2.3.16 DTCR

DTCR 为 Data Training 配置寄存器。

表 2-68: DTCR 寄存器域的描述

名称	范围	类型	描述
RFSHDT	[31:28]	RW,0x0	训练期间 Refresh: 非 0 值表示在每个 Rank 训练之后 (除了最后一个 rank), 将指定数量的刷新命令发送到相应的 SDRAM。
RANKEN	[27:24]	RW,0xF	RANK 使能: 指定 RANK 进行数据训练 (Read DQS Gate 训练), 每位对应一个 RANK。
DTEXG	[23]	RW,0x0	数据训练 DQS Gate 扩展使能。置 “1” 表示 DQS Gate 训练时, 按 DSGCR[DQSGX]要求, 扩展 DQS Gate。
DTDSTP	[21]	RW,0x0	Debug 数据训练单步模式, 该位会自清除。

DTDEN	[20]	RW,0x0	数据训练 Debug 使能：置“1”表示以 Debug 模式进行数据训练。
-------	------	--------	---------------------------------------

DTDBS	[19:16]	RW,0x0	数据训练 Debug 字节选择。在数据训练 Debug 模式下，选择相应字节。
DTWBDDM	[12]	RW,0x1	DM 是否参与 Bit Deskew 调节的控制使能。
DTWDQM	[11:8]	RW,0x5	WDQ 训练的 Margin 设置。
DTCMPD	[7]	RW,0x1	数据训练数据比较。置“1”表示在 DQS Gate 训练时检查返回的数据是否正确。否则，只对返回数据的个数（DQS 边沿数量）进行检查。
DTMPR	[6]	RW,0x0	使用 MPR 进行数据训练。置“1”，指定使用 SDRAM 的 MPR 寄存器进行 DQS Gate 训练，否则，将先写入指定 SDRAM 空间，然后读回。
DTRANK	[5:4]	RW,0x0	数据训练 RANK 号：用于指定进行数据 bit deskew 和眼图训练的 Rank 号。
DTRPTN	[3:0]	RW,0x7	数据训练重复次数：指定数据训练时读、写次数，验证 DDR 的稳定性。
—	其它	—	保留。

### 2.3.17 DTAR0-3

DTAR 为 Data Training 地址寄存器。

表 2-69: DTAR0 寄存器域的描述

名称	范围	类型	描述
DTBANK	[30:28]	RW,0x0	数据训练 BANK 地址：用于数据训练时，指定 SDRAM 的 BANK 地址。
DTROW	[27:12]	RW,0x0	数据训练行地址：用于数据训练时，指定 SDRAM 行地址。
DTCOL	[11:0]	RW,0x0	数据训练列地址：用于数据训练时，指定 SDRAM 列地址。地址低三位必须为“3'b000”。
—	其它	—	保留。

表 2-70: DTAR1 寄存器域的描述

名称	范围	类型	描述
DTBANK	[30:28]	RW,0x0	数据训练 BANK 地址：用于数据训练时，指定 SDRAM 的 BANK 地址。
DTROW	[27:12]	RW,0x0	数据训练行地址：用于数据训练时，指定 SDRAM 行地址。

DTCOL	[11:0]	RW,0x0	数据训练列地址：用于数据训练时，指定 SDRAM 列地址。 地址低三位必须为“3'b000”。
-------	--------	--------	----------------------------------------------------

—	其它	—	保留。
---	----	---	-----

表 2-71: DTAR2 寄存器域的描述

名称	范围	类型	描述
DTBANK	[30:28]	RW,0x0	数据训练 BANK 地址: 用于数据训练时, 指定 SDRAM 的 BANK 地址。
DTROW	[27:12]	RW,0x0	数据训练行地址: 用于数据训练时, 指定 SDRAM 行地址。
DTCOL	[11:0]	RW,0x0	数据训练列地址: 用于数据训练时, 指定 SDRAM 列地址。地址低三位必须为“3'b000”。
—	其它	—	保留。

表 2-72: DTAR3 寄存器域的描述

名称	范围	类型	描述
DTBANK	[30:28]	RW,0x0	数据训练 BANK 地址: 用于数据训练时, 指定 SDRAM 的 BANK 地址。
DTROW	[27:12]	RW,0x0	数据训练行地址: 用于数据训练时, 指定 SDRAM 行地址。
DTCOL	[11:0]	RW,0x0	数据训练列地址: 用于数据训练时, 指定 SDRAM 列地址。地址低三位必须为“3'b000”。
—	其它	—	保留。

### 2.3.18 DTDR0-1

DTDR 为 Data Training 数据寄存器。

表 2-73: DTDR0 寄存器域的描述

名称	范围	类型	描述
DTBYTE3	[31:24]	RW,0XDD	数据训练的数据 (字节 0~字节 3)。在数据训练期间, 每个通道采用相同的数据。缺省单个周期序列步长为“1”。
DTBYTE2	[23:16]	RW,0X22	
DTBYTE1	[15:8]	RW,0XEE	
DTBYTE0	[7:0]	RW,0X11	
—	其它	—	保留。

表 2-74: DTDR1 寄存器域的描述

名称	范围	类型	描述
DTBYTE7	[31:24]	RW,0X77	数据训练的数据 (字节 4~字节 7)。在数据训练期间, 每个通道采用相同的数据。缺省单个周期序列步长为
DTBYTE6	[23:16]	RW,0X88	

DTBYTE5	[15:8]	RW,0XBB	“1”。
DTBYTE4	[7:0]	RW,0X44	
—	其它	—	保留。

### 2.3.19 DTEDR

DTEDR 为 Data Training 眼图数据寄存器。

表 2-75: DTEDR0 寄存器域的描述

名称	范围	类型	描述
DTWBMX	[31:24]	RW,0x0	数据训练写通路 BDL shift 最大值。
DTWBMN	[23:16]	RW,0x0	数据训练写通路 BDL shift 最小值。
DTWLMX	[15:8]	RW,0x0	数据训练 WDQ LCDL 最大值。
DTWLMN	[7:0]	RW,0x0	数据训练 WDQ LCDL 最小值。
—	其它	—	保留。

表 2-76: DTEDR1 寄存器域的描述

名称	范围	类型	描述
DTRBMX	[31:24]	RW,0x0	数据训练读通路 BDL shift 最大值。
DTRBMN	[23:16]	RW,0x0	数据训练读通路 BDL shift 最小值。
DTRLMX	[15:8]	RW,0x0	数据训练 RDQS LCDL 最大值。
DTRLMN	[7:0]	RW,0x0	数据训练 RDQS LCDL 最小值。
—	其它	—	保留。

### 2.3.20 DCUAR

DCUAR 为 DCU 地址寄存器。

表 2-77: DCUAR 寄存器域的描述

名称	范围	类型	描述
ATYPE	[11]	RW,0x0	访问类型。 0=写访问; 1=读访问。
INCA	[10]	RW,0x0	地址加 1。置“1”时,每次访问 Cache 后, WADDR 和 SADDR 的 cache 地址自动加 1。

CSEL	[9:8]	RW,0x0	Cache 选择：选择被访问的 Cache。 2'b00 = 命令 cache;
			2'b01 = 期望值数据 cache; 2'b10 = 读数据 cache; 2'b11 = 保留。
CSADDR	[7:4]	RW,0x0	Cache slice 地址：本次访问的 Cache Slice 的地址，对应行内的 Slice 地址；读写接口都是 32 位，如果 Cache 行宽度大于 32 位，则需要通过 Slice 地址进行访问。
CWADDR	[3:0]	RW,0x0	Cache WORD 地址：本次访问的 cache word 的地址，对应 Cache 的行地址。
—	其它	—	保留。

### 2.3.21 DCUDR

DCUDR 为 DCU 数据寄存器。

表 2-78: DCUDR 寄存器域的描述

名称	范围	类型	描述
CDATA	[31:0]	RW,0x0	Cache 数据，从 Cache 读出或写入的数据接口。

### 2.3.22 DCURR

DCURR 为 DCU 运行寄存器。

表 2-79: DCURR 寄存器域的描述

名称	范围	类型	描述
XCEN	[23]	RW,0x0	期望值比较使能。置“1”时，从 SDRAM 读出的数据将与期望值比较；
RCEN	[22]	RW,0x0	读 Cache 使能。置“1”时，从 SDRAM 读出的数据被写入到 Read Data Cache 中；
SCOF	[21]	RW,0x0	读数据采样满后停止标志。置“1”表示对读出数据进行采样时，用于存放采样数据的 Cache 已满，则停止采样。
SNOF	[20]	RW,0x0	在 N 次错误后停止运行标志。该标志与 NFAIL 一起使用。
NFAIL	[19:12]	RW,0x0	失败次数。在执行命令过程中，发生错误的次数达到规定值，且 SNOF 寄存器=1 则停止运行。



EADDR	[11:8]	RW,0x0	结束地址。执行命令 Cache 中命令的结束地址
SADDR	[7:4]	RW,0x0	启动地址。执行命令 Cache 中命令的起始地址
DINST	[3:0]	RW,0x0	DCU 指令。
			4'b0000=空操作； 4'b0001=开始执行命令 Cache 中指令； 4'b0010=停止执行命令 Cache 中命令； 4'b0011=停止循环执行命令 Cache； 4'b0100=复位所有 DCU 中运行计时器； 4'b0101~4'b 1111 保留。
—	其它	—	保留。

### 2.3.23 DCULR

DCULR 为 DCU 循环控制寄存器。

表 2-80: DCULR 寄存器域的描述

名称	范围	类型	描述
XLEADDR	[31:28]	RW,0x0	期望数据循环结束地址。
IDA	[17]	RW,0x0	DRAM 地址加 1。每个读/写命令执行后，DRAM 地址加 1。
LINF	[16]	RW,0x0	无限循环。置“1”将循环执行 Cache 中的命令，直到被 STOP 命令停止，否则执行 LCNT 指定的循环次数。
LCNT	[15:8]	RW,0x0	LOOP 次数。如果 LINF 寄存器不为“1”，循环执行指定的次数。
LEADDR	[7:4]	RW,0x0	LOOP 结束地址，指定循环执行的命令 Cache 结束地址。
LSADDR	[3:0]	RW,0x0	LOOP 开始地址，指定循环执行的命令 Cache 开始地址。
—	其它	—	保留。

### 2.3.24 DCUGCR

DCUGCR 为 DCU 通用配置寄存器。

表 2-81: DCUGCR 寄存器域的描述

名称	范围	类型	描述
----	----	----	----

RCSW	[15:0]	RW,0x0	读出数据比较启动。在连续读出 RCSW 个数据后，启动对读出数据的比较。
—	其它	—	保留。

### 2.3.25 DCUTPR

DCUTPR 为 DCU 延时参数寄存器。

表 2-82: DCUTPR 寄存器域的描述

名称	范围	类型	描述
tDCUT3	[31:24]	RW,0x0	DCU 通用时间参数 3。
tDCUT2	[23:16]	RW,0x0	DCU 通用时间参数 2。
tDCUT1	[15:8]	RW,0x0	DCU 通用时间参数 1。
tDCUT0	[7:0]	RW,0x0	DCU 通用时间参数 0。

### 2.3.26 DCUSR0-1

DCUSR 为 DCU 状态寄存器。

表 2-83: DCUSR0 寄存器域的描述

名称	范围	类型	描述
CFULL	[2]	RO	Capture 满，为“1”表示 capture cache 满。
CFAIL	[1]	RO	Capture 失败，为“1”表示至少有一个读出数据有错。
RDONE	[0]	RO	执行结束标志，在成功执行了 STOP 后，将置起该标志。
—	其它	—	保留。

表 2-84: DCUSR1 寄存器域的描述

名称	范围	类型	描述
LPCNT	[31:24]	RO	循环次数：已经循环执行命令的次数。主要用于因出错停止执行的情况。
FLCNT	[23:16]	RO	失败次数：读出失败的次数。
RDCNT	[15:0]	RO	读计数器：从 SDRAM 读出次数。

### 2.3.27 BISTR

BISTR 为 BIST 运行寄存器。

表 2-85: BISTRN 寄存器域的描述

名称	范围	类型	描述
----	----	----	----

BCCSEL	[26:25]	RW,0x0	<p>BIST 时钟周期选择。AC 的 LOOPBACK 测试时，从返回四个节拍中选择哪个写入 FIFO 中。</p> <p>2'b00 = Clock cycle 0, 4, 8, 12 等；          2'b01 = Clock cycle 1, 5, 9, 13 等；          2'b10 = Clock cycle 2, 6, 10, 14 等；          2'b11 = Clock cycle 3, 7, 11, 15 等。</p>
BCKSEL	[24:23]	RW,0x0	<p>BIST CK 选择。AC 的 LOOPBACK 测试时，从三个 CK 中选择哪个进行测试。取值范围如下：</p> <p>2'b00 = CK[0]；          2'b01 = CK[1]；          2'b10 = CK[2]；          2'b11 = 保留。</p>
BDXSEL	[22:19]	RW,0x0	<p>DATX8 的 BIST 测试使能时有意义，选择哪个 Datx8 进行 BIST 测试，有效取值范围 0~8，对应于 9 个 DatX8。</p>
BDPAT	[18:17]	RW,0x0	<p>BIST 测试数据模式选择：</p> <p>2'b00 = Walking 0；          2'b01 = Walking 1；          2'b10 = LFSR-based pseudo-random；          2'b11 = 用户自定义。</p>
BDMEN	[16]	RW,0x0	<p>BIST Data MASK 使能：该位为“1”表示 BIST 测试包括 DM 测试。</p>
BACEN	[15]	RW,0x0	<p>AC 的 BIST 测试使能。该位和 BDXEN 是互斥的，不能同时置“1”。</p>
BDXEN	[14]	RW,0x0	<p>DATX8 的 BIST 测试使能。该位和 BACEN 是互斥的，不能同时置“1”。</p>
BSNOF	[13]	RW,0x0	<p>BIST 测试发生 N 次报错时（次数由 NFAIL 指定），则停止执行。</p>
NFAIL	[12:5]	RW,0x0	<p>失败次数。如果 BSBOF 寄存器置“1”时，执行命令并发现读出数据出错次数达到 NFAIL+1 次，则停止测试。</p>
BINF	[4]	RW,0x0	<p>BIST 无限执行。置“1”表示无限执行 BIST 测试，直到被停止或遇到读出错误（满足退出条件）。置“0”表示按 BISTWCR[BWCNT]寄存器指定次数执行。</p>

BMODE	[3]	RW,0x0	BIST 模式: 0=自环模式, 地址、命令、数据 IO 上环回, 可以选择从 Pad 端或 Core 端换回; 1=DRAM 模式, 以正常访存方式。
BINIT	[2:0]	RW,0x0	BIST 指令, 选择 BIST 测试指令。 3'b000=空操作; 3'b001=启动 BIST 测试; 3'b010=停止 BIST 测试; 3'b011=清除所有运行相关的状态寄存器; 3'b100~3'b111 保留。
—	其它	—	保留。

### 2.3.28 BISTWCR

BISTWCR 为 BIST 字计数寄存器。

表 2-86: BISTWCR 寄存器域的描述

名称	范围	类型	描述
BWCNT	[15:0]	RW, 0X20	BIST Word 计数: BIST 生成数据计数。这个值应给是突发长度除以 2 的倍数。例如, BL=8, 则有效取值是 4、8、12、16、……。
—	其它	—	保留。

### 2.3.29 BISTMSKR0-2

BISTMSKR 为 BIST 屏蔽寄存器。

表 2-87: BISTMSKR0 寄存器域的描述

名称	范围	类型	描述
ODTMSK	[31:28]	RW,0x0	4 个 ODT 位屏蔽指示。
CSMSK	[27:24]	RW,0x0	4 个 CS_N 位屏蔽指示。
CKEMSK	[23:20]	RW,0x0	4 个 CKE 位屏蔽指示。
WEMSK	[19]	RW,0x0	WE_N 位屏蔽指示。
BAMSK	[18:16]	RW,0x0	3 位 BANK 地址屏蔽指示。

AMSK	[15:0]	RW,0x0	16 位地址屏蔽指示。
------	--------	--------	-------------

表 2-88: BISTMSKR1 寄存器域的描述

名称	范围	类型	描述
DMMSK	[31:28]	RW,0x0	DM 位屏蔽指示。
CASMSK	[1]	RW,0x0	CAS_N 位屏蔽指示。
RASMSK	[0]	RW,0x0	RAS_N 位屏蔽指示。
—	其它	—	保留。

表 2-89: BISTMSKR2 寄存器域的描述

名称	范围	类型	描述
DQMSK	[31:0]	RW,0x0	DQ 位屏蔽指示。

### 2.3.30 BISTLSR

BISTLSR 是 BIST 随机测试种子寄存器。

表 2-90: BISTLSR 寄存器域的描述

名称	范围	类型	描述
SEED	[31:0]	RW, 0X1234A BCD	随机 bist 测试数据时的种子 (LFSR)。

### 2.3.31 BISTAR0-2

BISTAR 是 BIST 地址寄存器。

表 2-91: BISTAR0 寄存器域的描述

名称	范围	类型	描述
BBANK	[30:28]	RW,0x0	BIST Bank 地址: 为 BIST 选择 SDRAM 的 BANK 地址。
BROW	[27:12]	RW,0x0	BIST 行地址: 为 BIST 选择 SDRAM 的行地址。
BCOL	[11:0]	RW,0x0	BIST 列地址: 为 BIST 选择 SDRAM 的列地址。
—	其它	—	保留。

表 2-92: BISTAR1 寄存器域的描述

名称	范围	类型	描述
BAINC	[15:4]	RW,0x0	BIST 地址增量: 为每个读/写访问指定地址的增量值。该值要满足对界要求。

BMRANK	[3:2]	RW,0x3	BIST 最大 RANK 数：为 BIST 指定最大 SDRAM 的 RANK 数。 0: 1 个 Rank; 1: 2 个 Rank; 2: 3 个 Rank; 3: 4 个 Rank。
BRANK	[1:0]	RW,0x0	BIST RANK 号：为 BIST 指定 RANK 号。 0: Rank 0; 1: Rank 1; 2: Rank 2; 3: Rank 3。
—	其它	—	保留。

表 2-93: BISTAR2 寄存器域的描述

名称	范围	类型	描述
BMBANK	[30:28]	RW, 0x7	BIST 最大 Bank 地址：为 BIST 指定 SDRAM 最大的 BANK 地址。
BMROW	[27:12]	RW, 0xFFF	BIST 最大行地址：为 BIST 指定 SDRAM 的最大行地址。
BMCOL	[11:0]	RW,0xFFF	BIST 最大列地址：为 BIST 指定 SDRAM 的最大列地址。
—	其它	—	保留。

### 2.3.32 BISTUDPR

BISTUDPR 是 BIST 用户制定测试向量寄存器。

表 2-94: BISTUDPR 寄存器域的描述

名称	范围	类型	描述
BUDP1	[31:16]	RW, 0Xffff	BIST 数据模式 1: 在 BIST 测试时, 该数据依次被发送到奇数的 DQ 引脚上。
BUDP0	[15:0]	RW, 0X0000	BIST 数据模式 0: 在 BIST 测试时, 该数据依次被发送到偶数的 DQ 引脚上。



### 2.3.33 BISTGSR

BISTGSR 是 BIST 通用状态寄存器。

表 2-95: BISTGSR 寄存器域的描述

名称	范围	类型	描述
CASBERR	[31:30]	RO	CAS 位错误: 记录 CAS 出错的次数。
RASBERR	[29:28]	RO	RAS 位错误: 记录 RAS 出错的次数。
DMBERR	[27:20]	RO	DM 位错错误: Data Mask 位错误次数。 DMBER[1:0]: 对应 Beat 0; DMBER[3:2]: 对应 Beat 1; DMBER[5:4]: 对应 Beat 2; DMBER[7:6]: 对应 Beat 3。
BDXERR	[2]	RO	BIST 数据错: 对数据通道测试时, 出现结果比较不等, 则报错。
BACERR	[1]	RO	BIST 地址/命令错: 对地址/命令通道测试时, 出现结果比较不等, 则报错。
BDONE	[0]	RO	BIST 测试完成标志。当重新进行 bist 测试时, 该位会自动清零;
—	其它	—	保留。

### 2.3.34 BISTWER

BISTWER 是 BIST 字错计数寄存器。

表 2-96: BISTWER 寄存器域的描述

名称	范围	类型	描述
DXWERR	[31:16]	RO	BIST 测试中, DATX8 发生错误的次数。
ACWERR	[15:0]	RO	BIST 测试中, AC 发生错误的次数。

### 2.3.35 BISTBER0-3

BISTBER 是位字错计数寄存器。

表 2-97: BISTBER0 寄存器域的描述

名称	范围	类型	描述
----	----	----	----

ABER	[31:0]	RO	地址 A[15:0]的各 bit 位错误计数器。每个地址线两位，例如 A[0]对应于 ABER[1:0]，A[1]对应 ABER[3:2]，依次类推。
------	--------	----	-----------------------------------------------------------------------------

表 2-98: BISTBER1 寄存器域的描述

名称	范围	类型	描述
ODTBER	[31:24]	RO	ODT[3:0]信号位错误计数器，每位信号对应两位。
CSBER	[23:16]	RO	CS_N[3:0]信号位错误计数器，每位信号对应两位。
CKEBER	[15:8]	RO	CKE[3:0]信号位错误计数器，每位信号对应两位。
WEBER	[7:6]	RO	WE_N 信号错误计数器。
BABER	[5:0]	RO	BANK[2:0]信号位错计数器，每位信号对应两位。

表 2-99: BISTBER2 寄存器域的描述

名称	范围	类型	描述
DQBER0	[31:0]	RO	DQS 偶数脉冲（第 0 个、第 2 个...）上数据对应的错误计数器。第 1 个 16 位对应上升沿（Beat 0、Beat 4...），第 2 个 16 位对应下降沿（Beat 1、Beat 5...）。每个 16 位对应 8 位数据，每个数据占用两位。

表 2-100: BISTBER3 寄存器域的描述

名称	范围	类型	描述
DQBER1	[31:0]	RO	DQS 奇数脉冲（第 1 个、第 3 个...）上数据对应的错误计数器。第 1 个 16 位对应上升沿（Beat 2、Beat 6...），第 2 个 16 位对应下降沿（Beat 3、Beat 7...）。每个 16 位对应 8 位数据，每个数据两位。

### 2.3.36 BISTWCSR

BISTWCSR 是 BIST 字计数统计寄存器。

表 2-101: BISTWER 寄存器域的描述

名称	范围	类型	描述
DXWCNT	[31:16]	RO	数据通道 WORD 统计，从数据通道上接收到的数量。
ACWCNT	[15:0]	RO	地址/命令通道 WORD 统计，从 ADDR/CMD 通道上接收到的数量。

### 2.3.37 BISTFWR0-2

BISTFWR 是 BIST 错误字寄存器。

表 2-102: BISTFWR0 寄存器域的描述

名称	范围	类型	描述
----	----	----	----

ODTWEBS	[31:28]	RO	4 位 ODT 信号线 WORD 错误状态。
CSWEBS	[27:24]	RO	4 位 CS# 信号线 WORD 错误状态。
CKEWEBS	[23:20]	RO	4 位 CKE 信号线 WORD 错误状态。
WEWEBS	[19]	RO	WE_N 信号线 WORD 错误状态。
BAWEBS	[18:16]	RO	3 位 Bank 地址线 WORD 错误状态。
AWEBS	[15:0]	RO	16 位地址线 WORD 错误状态。

表 2-103: BISTFWR1 寄存器域的描述

名称	范围	类型	描述
DMWEBS	[31:28]	RO	DM 信号 WORD 错误状态。DMWEBS[0]对应于第 1 个 Beat (对应 DQS 的上升沿), DMWEBS[1]对应于第 2 个 Beat (对应 DQS 的下降沿), 以此类推共四个 Beat。
CASWEBS	[1]	RO	CAS_N 信号线 WORD 错误状态。
RASWEBS	[0]	RO	RAS_N 信号线 WORD 错误状态。
—	其它	—	保留。

表 2-104: BISTFWR2 寄存器域的描述

名称	范围	类型	描述
DQWEBS	[31:0]	RO	通道内 8 位数据线 WORD 错误状态。第 1 个 8 位对应第 1 个 Beat (对应 DQS 的上升沿), 第 2 个 8 位对应第 2 个 Beat (对应 DQS 的下降沿), 依次类推共 4 个 Beat。8 位之内每一

### 2.3.38 ZQnCR0-1

ZQnCR 是 ZQ 控制寄存器。

表 2-105: ZQnCR0 寄存器域的描述

名称	范围	类型	描述
ZQPD	[31]	RW,0x0	ZQ PowerDown: 置“1”表示 PZQ 单元被 Power Down。
ZCALEN	[30]	RW,0x1	阻抗校准使能, 置“1”将通过 PIR 寄存器的 ZCAL 或 DFI 接口触发 ZQ 控制模块进行阻抗校准。
ZCALBYP	[29]	RW,0x0	旁路阻抗校准, 置“1”表示旁路阻抗校准。 ZCALBYP 优先级高于 ZCALEN。
ZDEN	[28]	RW,0x0	阻抗数据使能: 置“1”表示直接使用 ZDATA 数据域控制阻抗。

ZDATA	[27:0]	RW, 0x14a	阻抗数据，该数据直接驱动阻抗控制。 ZDATA[27:21]用于选择片上端接上拉阻抗；
			ZDATA[20:14]用于选择片上端接下拉阻抗； ZDATA[13:7]用于输出阻抗上拉阻抗； ZDATA[6:0]用于输出阻抗下拉阻抗。 片上端接控制如下： 0x0e: 120 欧姆； 0x13: 80 欧姆； 0x19: 60 欧姆； 0x26: 40 欧姆； 0x32: 30 欧姆； 输出阻抗控制如下： 0x2a: 45 欧姆； 0x30: 40 欧姆； 0x38: 34 欧姆。

表 2-106: ZQnCR1 寄存器域的描述

名称	范围	类型	描述
ZPROG	[7:0]	RW,0x7B	通过外部端接高精度 240 欧姆电阻比率来设置输出阻抗和片上端接，具体如下： ZPROG[7:4]对应片上端接电阻。 3: 120 欧姆，即外接电阻的 1/2； 7: 60 欧姆，即外接电阻的 1/4； 11: 40 欧姆，即外接电阻的 1/6。 ZPROG[3:0]对应输出阻抗。 11: 40 欧姆，即外接电阻的 1/6； 13: 34 欧姆，即外接电阻的 1/7。
—	其它	—	保留。

### 2.3.39 ZQnSR0-1

ZQnSR 是 ZQ 状态寄存器。

表 2-107: ZQnSR0 寄存器域的描述

名称	范围	类型	描述
ZDONE	[31]	RO	阻抗校准结束，为“1”表示已经完成了阻抗校准。

ZERR	[30]	RO	阻抗校准错，为“1”表示在阻抗校准过程中发生错误。
ZCTRL	[27:0]	RO	阻抗控制：阻抗控制的当前数值。
			ZDATA[27:21]用于选择片上端接上拉阻抗； ZDATA[20:14]用于选择片上端接下拉阻抗； ZDATA[13:7]用于输出阻抗上拉阻抗； ZDATA[6:0]用于输出阻抗下拉阻抗。
—	其它	—	保留。

表 2-108: ZQnSR1 寄存器域的描述

名称	范围	类型	描述
OPU	[7:6]	RO	片上端接上拉端接校准状态，各值含义如下： 2'b00=没有错； 2'b01=上溢错； 2'b10=下溢错； 2'b11=在校准处理中。
OPD	[5:4]	RO	片上端接下拉端接校准状态，各值含义如下： 2'b00=没有错； 2'b01=上溢错； 2'b10=下溢错； 2'b11=在校准处理中。
ZPU	[3:2]	RO	输出阻抗上拉校准状态，各值含义如下： 2'b00=没有错； 2'b01=上溢错； 2'b10=下溢错； 2'b11=在校准处理中。
ZPD	[1:0]	RO	输出阻抗下拉校准状态，各值含义如下： 2'b00=没有错； 2'b01=上溢错； 2'b10=下溢错； 2'b11=在校准处理中。
—	其它	—	保留。

### 2.3.40 DXnGCR

DXnGCR 是 Datx8 的通用控制寄存器。

表 2-109: DXnGCR 寄存器域的描述

名称	范围	类型	描述
CALBYP	[31]	RW,0x0	旁路校准: 置“1”表示在 Phy 初始化后关闭自动触发校准。
MDLEN	[30]	RW,0x1	主延迟线 (MDL) 使能: 置“1”表示在初始周期测量后, 开启 DATX8 的主延迟线周期校准。 该配置将与 DXCCR 寄存器的相关配置信息 (MDLEN) 进行共同控制 (“或”操作)。
WLRKEN	[29:26]	RW,0xF	Rank Write Leveling 使能。指定 RANK 进行 Write Leveling 处理。 对于没有指定的 RANK, Write Leveling 的结果被忽略, 并直接标识为完成。 WLRKEN[0]对应 Rank0; WLRKEN[1]对应 Rank1; WLRKEN[2]对应 Rank2; WLRKEN[3]对应 Rank3。
GSHIFT	[18]	RW,0x0	PLL 变速使能。置“1”时, PLL 处于快速 Lock 模式。 该配置将与 PLLCR 寄存器的相关配置信息 (GSHIFT) 进行共同控制 (“或”操作)。
PLLPD	[17]	RW,0x0	PLL PowerDown: 通过 PLL PowerDown 端口将 PLL 设置成 PowerDown 模式。该寄存器没有自清功能, 必须软件清除。 该配置将与 PLLCR 寄存器的相关配置信息 (PLLPD) 进行共同控制 (“或”操作)。
PLLRST	[16]	RW,0x0	PLL 复位。通过 PLL 复位端口对 DatX8 的 PLL 复位。该寄存器没有自清功能, 必须软件清除。 该配置将与 PLLCR 寄存器的相关配置信息 (PLLRST) 进行共同控制 (“或”操作)。
RTTOAL	[13]	RW,0x0	ODT 开启附加延迟, 表示在 Read 期间, 何时开启 ODT (DQS 根据 DQSODT、DQ/DM 根据 DQODT 配置)。 0=在 Read 数据前导之前 2 个周期开启 ODT; 1=在 Read 数据前导之前 1 个周期开启 ODT。
RTTOH	[12:11]	RW,0x1	ODT 输出保持时间: 表示在读数据后导之后, ODT 仍维持 0~3 个周期 (DQS 根据 DQSODT、DQ/DM 根据 DQODT 配置)。如果是动态 ODT 方式, RTTOH 之后会关闭 ODT。

DQRTT	[10]	RW,0x1	<p>DQ 动态 RTT 控制。</p> <p>1=表示 DQ/DM 的 ODT 在 Read 时根据 DQODT 动态控制，其它时候不使能；</p> <p>0=表示 DQ/DM 的 ODT 始终根据 DQODT 控制。</p>
DQSRTT	[9]	RW,0x1	<p>DQS 动态 ODT 控制。</p> <p>1=表示 DQS 的 ODT 在 Read 时根据 DQSODT 动态控制，其它时候不使能；</p> <p>0=表示 DQS 的 ODT 始终根据 DQSODT 控制。</p>
DSEN	[8:7]	RW,0x1	<p>写 DQS 使能：控制到 SDRAM 的写 DQS 是否使能或者 DQS 反相。有效取值范围如下：</p> <p>2'b00=DQS 始终驱动为低电平；</p> <p>2'b01=DQS 极性反相；</p> <p>2'b10=DQS 正常工作；</p> <p>2'b11=DQS 始终驱动为高电平。</p> <p>这些值只有在 DQS/DQS_N 输出使能时才有效，否则 DQS/DQS_N 为三态。</p>
DQSRPD	[6]	RW,0x0	<p>DQSR 单元 PowerDown：置“1”表示 PDQSR 单元设置成 PowerDown。</p> <p>该配置将与 DXCCR 寄存器的相关配置信息（DXPDR）进行共同控制（“或”操作）。</p>
DXPDR	[5]	RW,0x0	<p>DQS/DQ/DM 接收 PowerDown：置“1”表示 DQ、DM 和 DQS/DQS_N 的输入接收器设置成 PowerDown。</p> <p>该配置将与 DXCCR 寄存器的相关配置信息（DXPDR）进行共同控制（“或”操作）。</p>
DXPDD	[4]	RW,0x0	<p>DQS/DQ/DM 发送 PowerDown：置“1”表示 DQ、DM 和 DQS/DQS_N 的输出驱动器设置成 PowerDown。</p> <p>该配置将与 DXCCR 寄存器的相关配置信息（DXPDD）进行共同控制（“或”操作）。</p>
DXIOM	[3]	RW,0x0	<p>DQS/DQ/DM IO 模式选择：</p> <p>0=选择 SSTL 模式；</p> <p>1=选择 CMOS 模式。</p> <p>该配置将与 DXCCR 寄存器的相关配置信息（DXIOM）进行共同控制（“或”操作）。</p>



DQODT	[2]	RW,0x0	DQ 片上端接：置“1”表示打开 DQ、DM 信号 SSTL IO 的片上端接。该配置将与 DXCCR 寄存器的相关配置信息（DXODT）进行共同控制（“或”操作）。
QSODT	[1]	RW,0x1	DQS 片上端接：置“1”表示打开 DQS/DQS_N 信号 SSTL IO 的片上端接。该配置将与 DXCCR 寄存器的相关配置信息（DXODT）进行共同控制（“或”操作）。
DXEN	[0]	读/写	数据通道使能： 1=打开该数据通道； 0=关闭通道，此时初始化、数据训练和读写操作均被屏蔽。
—	其它	—	保留。

### 2.3.41 DXnGSR0-1

DXnGSR 是 Datx8 的通用状态寄存器。

表 2-110: DXnGSR0 寄存器域的描述

名称	范围	类型	描述
DTERR	[30]	RO	表示 Data Training 出错。
WBDWN	[29]	RO	表示 Write Bit Skew 算法没有找到左边沿，即 WDQ 的 LCDL 和 BDL 已经设置为最大值仍没有找到左边沿。
WEYEWN	[28]	RO	表示 Write Eye 算法没有找到左边沿或右边沿，即 WDQ 的 LCDL 和 BDL 已经设置为最大值仍没有找到左边沿，或者 WDQ 的 LCDL 和 BDL 已经设置为最小值仍没有找到右边沿。
QSGERR	[27:24]	RO	DQS Gating 训练错：为“1”表示在 DQS Gate 训练时发生错误，每个比特位对应一个 RANK。
GDQSPRD	[23:16]	RO	DQS Gate Read period: 读 DQS Gate 的 LCDL 延迟线测得的 DDR 时钟周期数，这个值需要进行 PVT 校准。
DPLOCK	[15]	RO	DATX8 PLL LOCK: 为“1”表示 DATX8 的 PLL 已经 LOCK。
WLPRD	[14:7]	RO	Write Leveling Period: Write Leveling LCDL 延迟线测得的 DDR 周期数，这个值需要进行 PVT 校准。
WLERR	[6]	RO	Write Leveling Error: 为“1”表示 DATX8 在 Write Leveling 中报错。

WLDONE	[5]	RO	Write Leveling Done: 为“1”表示 DATX8 已经完成 Write Leveling。
WLCAL	[4]	RO	Write Leveling 校准: 为“1”表示 DATX8 已经为 Write Leveling LCDL (DXnLCDLR1[RxWLD]) 完成周期测量。
GDQSCAL	[3]	RO	DQS Gate Read 校准: 为“1”表示 DATX8 已经为 DQS gate 读通路 LCDL (DXnLCDLR1[RxDQSGD]) 完成周期测量。
RDQSNCAL	[2]	RO	DQS_N 读校准: 为“1”表示 DATX8 已经为 DQS_N 读通路 LCDL (DXnLCDLR1[RDQSN]) 完成周期测量。
RDQSCAL	[1]	RO	DQS 读校准: 为“1”表示 DATX8 已经为 DQS 读通路 LCDL (DXnLCDLR1[RDQSD]) 完成周期测量。
WDQCAL	[0]	RO	DQ 写校准: 为“1”表示 DATX8 已经为 DQ 写通路 LCDL (DXnLCDLR1[WDQD]) 完成周期测量。
—	其它	—	保留。

表 2-111: DXnGSRI 寄存器域的描述

名称	范围	类型	描述
DLTCODE	[24:1]	RO	延迟线测试结果, 返回 DATX8 延迟线测量 (DLL 环振测试) 得到的周期数。
DLTDONE	[0]	RO	延迟线测试结束, 表示已经完成了 DATX8 的延迟线测量 (DLL 环振测试)。
—	其它	—	保留。

### 2.3.42 DXnBDLR0-4

DXnBDLR 是 Datx8 的位延时的匹配寄存器。

表 2-112: DXnBDLR0 寄存器域的描述

名称	范围	类型	描述
DQ4WBD	[29:24]	RW,0x0	DQ4 写延迟: DQ4 写通路 BDL 延迟值。
DQ3WBD	[23:18]	RW,0x0	DQ3 写延迟: DQ3 写通路 BDL 延迟值。
DQ2WBD	[17:12]	RW,0x0	DQ2 写延迟: DQ2 写通路 BDL 延迟值。
DQ1WBD	[11:6]	RW,0x0	DQ1 写延迟: DQ1 写通路 BDL 延迟值。
DQ0WBD	[5:0]	RW,0x0	DQ0 写延迟: DQ0 写通路 BDL 延迟值。
—	其它	—	保留。

表 2-113: DXnBDLR1 寄存器域的描述

名称	范围	类型	描述
----	----	----	----

DQSWBD	[29:24]	RW,0x0	写 DQS 延迟: 写 DQS 通路 BDL 延迟值。
DMWBD	[23:18]	RW,0x0	DM 写延迟: DM 写通路 BDL 延迟值。
DQ7WBD	[17:12]	RW,0x0	DQ7 写延迟: DQ7 写通路 BDL 延迟值。
DQ6WBD	[11:6]	RW,0x0	DQ6 写延迟: DQ6 写通路 BDL 延迟值。
DQ5WBD	[5:0]	RW,0x0	DQ5 写延迟: DQ5 写通路 BDL 延迟值。
—	其它	—	保留。

表 2-114: DXnBDLR2 寄存器域的描述

名称	范围	类型	描述
DSNRBD	[23:18]	RW,0x0	读 DQS_N 延迟: 读 DQS_N 通路 BDL 延迟值。
DSRBD	[17:12]	RW,0x0	读 DQS 延迟: 读 DQS 通路 BDL 延迟值。
DQOEBD	[11:6]	RW,0x0	DQ 输出使能延迟: DQ/DM 输出使能 BDL 延迟值。
DSOEBD	[5:0]	RW,0x0	DQS 输出使能延迟: DQS 输出使能 BDL 延迟值。
—	其它	—	保留。

表 2-115: DXnBDLR3 寄存器域的描述

名称	范围	类型	描述
DQ4RBD	[29:24]	RW,0x0	DQ4 读延迟: DQ4 读通路 BDL 延迟值。
DQ3RBD	[23:18]	RW,0x0	DQ3 读延迟: DQ3 读通路 BDL 延迟值。
DQ2RBD	[17:12]	RW,0x0	DQ2 读延迟: DQ2 读通路 BDL 延迟值。
DQ1RBD	[11:6]	RW,0x0	DQ1 读延迟: DQ1 读通路 BDL 延迟值。
DQ0RBD	[5:0]	RW,0x0	DQ0 读延迟: DQ0 读通路 BDL 延迟值。
—	其它	—	保留。

表 2-116: DXnBDLR4 寄存器域的描述

名称	范围	类型	描述
DMRBD	[23:18]	RW,0x0	DM 读延迟: DM 读通路 BDL 延迟值。
DQ7RBD	[17:12]	RW,0x0	DQ7 读延迟: DQ7 读通路 BDL 延迟值。
DQ6RBD	[11:6]	RW,0x0	DQ6 读延迟: DQ6 读通路 BDL 延迟值。
DQ5RBD	[5:0]	RW,0x0	DQ5 读延迟: DQ5 读通路 BDL 延迟值。
—	其它	—	保留。

### 2.3.43 DXnLCDLR0-2

DXnLCDLR 是 Datx8 的本地主延时的匹配寄存器。

表 2-117: DXnLCDLR0 寄存器域的描述

名称	范围	类型	描述
R3WLD	[31:24]	RW,0x0	Rank3 Write Leveling 延迟: 为 RANK3 指定 LCDL Write Leveling 延迟。该寄存器与 DXnGTR[R3WLSL]共同起作用。
R2WLD	[23:16]	RW,0x0	Rank2 Write Leveling 延迟: 为 RANK2 指定 LCDL Write Leveling 延迟。该寄存器与 DXnGTR[R2WLSL]共同起作用。
R1WLD	[15:8]	RW,0x0	Rank1 Write Leveling 延迟: 为 RANK1 指定 LCDL Write Leveling 延迟。该寄存器与 DXnGTR[R1WLSL]共同起作用。
R0WLD	[7:0]	RW,0x0	Rank0 Write Leveling 延迟: 为 RANK0 指定 LCDL Write Leveling 延迟。该寄存器与 DXnGTR[R0WLSL]共同起作用。

表 2-118: DXnLCDLR1 寄存器域的描述

名称	范围	类型	描述
RDQSNL	[23:16]	RW,0x1f	读 DQS_N 延迟: 为读 DQS_N 指定 LCDL 延迟, 其默认值是 1600MHz (DDR3-1600 的 DDR 时钟) 的 1/2 拍。
RDQSD	[15:8]	RW,0x1f	读 DQS 延迟: 为读 DQS 指定 LCDL 延迟, 其默认值是 1600MHz (DDR3-1600 的 DDR 时钟) 的 1/2 拍。
WDQD	[7:0]	RW,0x1f	写数据延迟: 为写数据指定 LCDL 延迟, 其默认值是 1600MHz (DDR3-1600 的 DDR 时钟) 的 1/2 拍。
—	其它	—	保留。

表 2-119: DXnLCDLR2 寄存器域的描述

名称	范围	类型	描述
R3DQSGD	[31:24]	RW,0x6e	RANK3 读 DQS Gating 延迟: 为 Rank3 读 DQS Gating 指定 LCDL 延迟。该寄存器与 DXnGTR[R3DGSL]共同起作用, 其默认值是 800MHz (DDR3-1600 的 SDR 时钟) 的一拍。
R2DQSGD	[23:16]	RW,0x6e	RANK2 读 DQS Gating 延迟: 为 Rank2 读 DQS Gating 指定 LCDL 延迟。该寄存器与 DXnGTR[R2DGSL]共同起作用, 其默认值是 800MHz (DDR3-1600 的 SDR 时钟) 的一拍。
R1DQSGD	[15:8]	RW,0x6e	RANK1 读 DQS Gating 延迟: 为 Rank1 读 DQS Gating 指定 LCDL 延迟。该寄存器与 DXnGTR[R1DGSL]共同起作用, 其默认值是 800MHz (DDR3-1600 的 SDR 时钟) 的一拍。

R0DQSGD	[7:0]	RW,0x6e	RANK0 读 DQS Gating 延迟: 为 Rank0 读 DQS Gating 指定 LCDL 延迟。该寄存器与 DXnGTR[R0DGSL]共同起作用, 其默认值是 800MHz (DDR3-1600 的 SDR 时钟) 的一拍。
---------	-------	---------	--------------------------------------------------------------------------------------------------------------------------

### 2.3.44 DXnMDLR

DXnMDLR 是 Datx8 的主延时寄存器。

表 2-120: DXnMDLR 寄存器域的描述

名称	范围	类型	描述
MDLD	[23:16]	RW,0x0	MDL 延迟: 采用折半查找的方法, 对 DDR 周期进行测量, 该配置用于指定测量过程中 MDL 延迟值。
TPRD	[15:8]	RW,0x0	目标周期值。主延迟线周期测量值 (主延迟线更新功能使能时, 该值会周期性的更新)。该值用于延迟线校准, 用作比例系数的分子。其它延时线 (LCDL 或 BDL) 根据该调节因子 (TPRD/ IPRD) 去计算延时线更新后的值, 如果偏差超过阈值, 则更新其它延迟线 (PHY 发起 Update 操作)。
IPRD	[7:0]	RW,0x0	初始周期值。主延迟线初始测量得到的周期值。该值用于 延迟线校准, 用作比例系数的分母。
—	其它	—	保留。

### 2.3.45 DXnGTR

DXnGTR 为 Datx8 通用延时寄存器。

表 2-121: DXnGTR 寄存器域的描述

名称	范围	类型	描述
R3WLSL	[19:18]	RW,0x1	Rank0~3 的 Write Leveling 延迟: 该值用于 Write Leveling 之后的 Pipe Line 调整。上电后, 缺省值为 01。该值会根据 Data Training 结果进行调整, 也可以直接写入来直接
R2WLSL	[17:16]	RW,0x1	
R1WLSL	[15:14]	RW,0x1	

R0WLSL	[13:12]	RW,0x1	配置该值。有效取值范围 0~2: 2'b00: 写延迟=WL-1; 2'b01: 写延迟=WL; 2'b10: 写延迟=WL+1; 2'b11: 保留。
R3DGSL	[11:9]	RW,0x0	Rank0~3 的 DQS Gating system 延迟: 为了补偿系统板级延迟、封装延迟、IO 延迟等各种延迟以获得有效的读数数据, 使用该寄存器延迟 DQS Gating, 最大可达 7 个 SDR Clock 周期。在上电后, 缺省值为 00。该值会根据 Data Training 结果进行调整, 也可以直接写入来直接配置该值。有效范围是 0~7。
R2DGSL	[8:6]	RW,0x0	
R1DGSL	[5:3]	RW,0x0	
R0DGSL	[2:0]	RW,0x0	
—	其它	—	保留。

## 2.4 DDR3PHY\_FPGA 的 IO 寄存器

说明:

- 1) DDR3PHY FPGA 采用 1 个 AC 和 18 个 DATX8 的方式;
- 2) DDR3PHY FPGA 内部设置的 IOR 采用 PUB0 的 CFG 接口进行访问, 其编址是利用 PUB 中的 保留地址;
- 3) 特殊注意: DDR3PHY FPGA 不能进行 PUB1 的 CFG 接口访问, 否则会挂死;

### 2.4.1 写延时参数调节寄存器

写通路延时控制寄存器; WDELAY\_SEL0 复位值是 0x44444444, WDELAY\_SEL1 复位值是

0x44444444, WDELAY\_SEL2 复位值是 0x44;

2G 只有 9 个 DataX8, 只需要设置 Datx0~ Datx8。

表 2-1: WDELAY\_SEL0 寄存器域的描述

名称	范围	类型	描述	默认值
DatX7SEL	[31:28]	RW		0x4
DatX6SEL	[27:24]	RW		0x4
DatX5SEL	[23:20]	RW		0x4
DatX4SEL	[19:16]	RW		0x4
DatX3SEL	[15:12]	RW		0x4

DatX2SEL	[11:8]	RW		0x4
DatX1SEL	[7:4]	RW		0x4
DatX0SEL	[3:0]	RW	该值表示 DS 和 D 的相对关系： 3 表示不做调整； 4~7 表示 D 增加延时； 1~2 表示 DS 增加延时；	0x4

表 2-2: WDELAY\_SEL1 寄存器域的描述

名称	范围	类型	描述	默认值
DatX15SEL	[31:28]	RW		0x4
DatX14SEL	[27:24]	RW		0x4
DatX13SEL	[23:20]	RW		0x4
DatX12SEL	[19:16]	RW		0x4
DatX11SEL	[15:12]	RW		0x4
DatX10SEL	[11:8]	RW		0x4
DatX9SEL	[7:4]	RW		0x4
DatX8SEL	[3:0]	RW	该值表示 DS 和 D 的相对关系； 3 表示不做调整； 4~7 表示 D 增加延时； 1~2 表示 DS 增加延时；	0x4

表 2-3: WDELAY\_SEL3 寄存器域的描述

名称	范围	类型	描述	默认值
保留	[31:8]	—	读该位将返回零；	—
DatX17SEL	[7:4]	RW		0x4
DatX16SEL	[3:0]	RW	该值表示 DS 和 D 的相对关系； 3 表示不做调整； 4~7 表示 D 增加延时； 1~2 表示 DS 增加延时；	0x4

## 2.4.2 读延时参数调节寄存器

读通路控制寄存器；RDELAY\_SEL0 复位值是 0x44444444，RDELAY\_SEL1 复位值是

0x44444444，RDELAY\_SEL2 复位值是 0x44；

2G 只有 9 个 DataX8，只需要设置 Datx0~ Datx8。

表 2-4: RDELAY\_SEL0 寄存器域的描述

名称	范围	类型	描述	默认值
DatX7SEL	[31:28]	RW		0x4
DatX6SEL	[27:24]	RW		0x4
DatX5SEL	[23:20]	RW		0x4



DatX4SEL	[19:16]	RW		0x4
DatX3SEL	[15:12]	RW		0x4
DatX2SEL	[11:8]	RW		0x4
DatX1SEL	[7:4]	RW		0x4
DatX0SEL	[3:0]	RW	该值表示 QS 和 Q 的相对关系； 3 表示不做调整； 4~7 表示 QS 增加延时； 1~2 表示 Q 增加延时；	0x4

表 2-5: RDELAY\_SEL1 寄存器域的描述

名称	范围	类型	描述	默认值
DatX15SEL	[31:28]	RW		0x4
DatX14SEL	[27:24]	RW		0x4
DatX13SEL	[23:20]	RW		0x4
DatX12SEL	[19:16]	RW		0x4
DatX11SEL	[15:12]	RW		0x4
DatX10SEL	[11:8]	RW		0x4
DatX9SEL	[7:4]	RW		0x4
DatX8SEL	[3:0]	RW	该值表示 QS 和 Q 的相对关系； 3 表示不做调整； 4~7 表示 QS 增加延时； 1~2 表示 Q 增加延时；	0x4

表 2-6: RDELAY\_SEL3 寄存器域的描述

名称	范围	类型	描述	默认值
保留	[31:8]	—	读该位将返回零；	—
DatX17SEL	[7:4]	RW		0x4
DatX16SEL	[3:0]	RW	该值表示 QS 和 Q 的相对关系； 3 表示不做调整； 4~7 表示 QS 增加延时； 1~2 表示 Q 增加延时；	0x4

### 2.4.3 读 Gate 延时参数调节寄存器

读 Gate 延时控制寄存器；QSENDELAY\_SEL0 寄存器复位值是 0x55555555，QSENDELAY\_SEL1

寄存器复位值是 0x55555555，QSENDELAY\_SEL2 寄存器复位值是 0x55；2G 只有 9 个 DataX8，只需要设置 Datx0~ Datx9。

表 2-7：QSENDELAY\_SEL0 寄存器域的描述

名称	范围	类型	描述	默认值
DatX7SEL	[31:28]	RW		0x5
DatX6SEL	[27:24]	RW		0x5
DatX5SEL	[23:20]	RW		0x5
DatX4SEL	[19:16]	RW		0x5
DatX3SEL	[15:12]	RW		0x5
DatX2SEL	[11:8]	RW		0x5
DatX1SEL	[7:4]	RW		0x5
DatX0SEL	[3:0]	RW	该值只调节 DQS Enable 的 pipe line;	0x5

表 2-8：QSENDELAY\_SEL1 寄存器域的描述

名称	范围	类型	描述	默认值
DatX15SEL	[31:28]	RW		0x5
DatX14SEL	[27:24]	RW		0x5
DatX13SEL	[23:20]	RW		0x5
DatX12SEL	[19:16]	RW		0x5
DatX11SEL	[15:12]	RW		0x5
DatX10SEL	[11:8]	RW		0x5
DatX9SEL	[7:4]	RW		0x5
DatX8SEL	[3:0]	RW	该值只调节 DQS Enable 的 pipe line;	0x5

表 2-9：QSENDELAY\_SEL3 寄存器域的描述

名称	范围	类型	描述	默认值
保留	[31:8]	—	读该位将返回零;	—
DatX17SEL	[7:4]	RW		0x5
DatX16SEL	[3:0]	RW	该值只调节 DQS Enable 的 pipe line;	0x5

## 2.4.4 读响应延时调节寄存器

复位值是  
0x0;

表 2-10: RESDELAY\_SEL 寄存器域的描述

名称	范围	类型	描述	默认值
保留	[31:4]	—	读该位将返回零;	—
RESDELAY_SEL	[3:0]	RW	控制所用的 DatX8, 调节给响应的时 间; 一般不调节该延时;	0x5

## 2.4.5 卸载点调节寄存器

复位值是  
0x0;

表 2-11: DOWNLOAD\_SEL 寄存器域的描述

名称	范围	类型	描述	默认值
保留	[31:4]	—	读该位将返回零;	—
DOWNLOAD	[3:0]	RW	控制所用的 DatX8, 调节卸载 点; 一般不调节该延时;	0x0

## 2.4.6 CfgCR 寄存器

该寄存器复位值为零; 如果是 RDIMM 配置, 必须配置该寄存器;

表 2-12: CfgCR 寄存器域的描述

名称	范围	类型	描述	默认值
保留	[31:12]	保留	保留	0x0
DIMM1_OBS_SEL	[11:8]	RW	DIMM1 OBS Select; 2G 只有一 个 DIMM, 即 9 个 Datx8。	0x0
DIMM0_OBS_SEL	[7:4]	RW	DIMM0 OBS Select	0x0
保留	[3:1]	保留	保留	0x0
RDIMM	[0]	RW	RDIMM 指示	0x0

## 2.4.7 CORE\_ENABLE 寄存器

该寄存器复位值为“1”；用于在 FPGA 环境中屏蔽不在位的核心的信号；

表 2-13: CfgCR 寄存器域的描述

名称	范围	类型	描述	默认值
保留	[31:4]	保留	保留	0x0
CORE_ENABLE	[3:0]	RW	CORE_ENABLE, 每位对应一个核心, 相应位为“1”表示核心在位, 否则在 CPM 入口把相应核心的信号屏蔽掉; 2G 只有两个核心, 即【1: 0】有意义。	0x1

## 2.4.8 UPDATE 寄存器

PUB0\_UPDATE 寄存器复位值是 0x0, PUB1\_UPDATE 寄存器复位值是 0x0;

2G 只有 1 个 DIMM, 只需要设置 PUB0\_UPDATE。该寄存器只为模拟 Update 时 MC 的行为。

表 2-14: PUB0\_UPDATE 寄存器域的描述

名称	范围	类型	描述	默认值
保留	[31:1]	—	读该位将返回零;	—
Update	[0]	RW	对该寄存器写 1, 将发起 Update 请求; 直到收到 MC 返回的 Ack 信号, 通过对该位写“0”结束 Update;	0x0

表 2-15: PUB1\_UPDATE 寄存器域的描述

名称	范围	类型	描述	默认值
保留	[31:1]	—	读该位将返回零;	—
Update	[0]	RW	对该寄存器写 1, 将发起 Update 请求; 直到收到 MC 返回的 Ack 信号, 通过对该位写“0”结束 Update;	0x0

## 2.4.9 MR 寄存器

MR0~MR3 说明见 PUB 的 IOR 说明；其编址与 PUB 的编址相同；

## 2.4.10 读 Gate 宽度调整寄存器

读 Gate 宽度调整寄存器，用于调整 DQS Gate 的宽度，可以向前或向后调整 1 拍；

读 Gate 延时调节寄存器和该寄存器共同作用，首先由读 Gate 延时调节寄存器确定延时的节拍

（调节粒度是 DDRCLK 周期），再根据该寄存器前后扩展调节；

注意以下配置组合没有意义：

- a)、QSEN\_DEALY 配置 0x9 或 0xa，则该寄存器配置向后扩展无意义； b) QSEN\_DEALY 配置 0x0 或 0x1，则该寄存器配置向前扩展无意义； 2G 只有 9 个 DataX8，只需要设置 Datx0~ Datx8。

表 2-16: QSENWIDTH\_SEL 寄存器域的描述

名称	范围	类型	描述	默认值
保留	[31:18]	保留		
DATX8	[17:16]	RW,0x0		0x0
DATX7	[15:14]	RW,0x0		0x0
DATX6	[13:12]	RW,0x0		0x0
DATX5	[11:10]	RW,0x0		0x0
DATX4	[9:8]	RW,0x0		0x0
DATX3	[7:6]	RW,0x0		0x0
DATX2	[5:4]	RW,0x0		0x0
DATX1	[3:2]	RW,0x0		0x0
DATX0	[1:0]	RW,0x0	DATX0 读 Gate 宽度调整： 00: 保持原来； 01: 向后扩展 1 个 SDRCLK； 10: 向前扩展 1 个 SDRCLK； 11: 前后各扩展 1 个 SDRCLK；	0x0

表 2-17: QSENWIDTH\_SEL 寄存器域的描述

名称	范围	类型	描述	默认值
保留	[31:18]	保留		
DATX17	[17:16]	RW,0x0		0x0
DATX16	[15:14]	RW,0x0		0x0
DATX15	[13:12]	RW,0x0		0x0
DATX14	[11:10]	RW,0x0		0x0

DATX13	[9:8]	RW,0x0		0x0
DATX12	[7:6]	RW,0x0		0x0
DATX11	[5:4]	RW,0x0		0x0
DATX10	[3:2]	RW,0x0		0x0
DATX9	[1:0]	RW,0x0	DATX9 读 Gate 宽度调整; 00: 保持原来; 01: 向后扩展 1 个 SDRCLK; 10: 向前扩展 1 个 SDRCLK; 11: 前后各扩展 1 个 SDRCLK;	0x0

## 2.4.11 写 Gate 延时参数调节寄存器

写 Gate 延时控制寄存器; DSENDELAY\_SEL0 寄存器复位值是 0x55555555, DSENDELAY\_SEL1

寄存器复位值是 0x55555555, DSENDELAY\_SEL2 寄存器复位值是 0x55; 2G 只有 9 个 DataX8, 只需要设置 Datx0~ Datx8。

表 2-18: DSENDELAY\_SEL0 寄存器域的描述

名称	范围	类型	描述	默认值
DatX7SEL	[31:28]	RW		0x5
DatX6SEL	[27:24]	RW		0x5
DatX5SEL	[23:20]	RW		0x5
DatX4SEL	[19:16]	RW		0x5
DatX3SEL	[15:12]	RW		0x5
DatX2SEL	[11:8]	RW		0x5
DatX1SEL	[7:4]	RW		0x5
DatX0SEL	[3:0]	RW	该值只调节 DQS Enable 的 pipe line;	0x5

表 2-19: DSENDELAY\_SEL2 寄存器域的描述

名称	范围	类型	描述	默认值
DatX15SEL	[31:28]	RW		0x5
DatX14SEL	[27:24]	RW		0x5
DatX13SEL	[23:20]	RW		0x5
DatX12SEL	[19:16]	RW		0x5

DatX11SEL	[15:12]	RW		0x5
DatX10SEL	[11:8]	RW		0x5

DatX9SEL	[7:4]	RW		0x5
DatX8SEL	[3:0]	RW	该值只调节 DQS Enable 的 pipe line;	0x5

表 2-20: DSENDELAY\_SEL3 寄存器域的描述

名称	范围	类型	描述	默认值
保留	[31:8]	—	读该位将返回零;	—
DatX17SEL	[7:4]	RW		0x5
DatX16SEL	[3:0]	RW	该值只调节 DQS Enable 的 pipe line;	0x5

## 2.4.12 写 Gate 宽度调整寄存器

写 Gate 宽度调整寄存器，用于调整 DQS Gate 的宽度，可以向前或向后调整 1 拍；

写 Gate 延时调节寄存器和该寄存器共同作用，首先由读 Gate 延时调节寄存器确定延时的节拍

（调节粒度是 DDRCLK 周期），再根据该寄存器前后扩展调节；

注意以下配置组合没有意义：

- a)、DSEN\_DEALY 配置 0x9 或 0xa，则该寄存器配置向后扩展无意义；
  - b) DSEN\_DEALY 配置 0x0 或 0x1，则该寄存器配置向前扩展无意义；
- 2G 只有 9 个 DataX8，只需要设置 Datx0~ Datx8。

表 2-21: DSENWIDTH\_SEL 寄存器域的描述

名称	范围	类型	描述	默认值
保留	[31:18]	保留		
DATX8	[17:16]	RW,0x0		0x0
DATX7	[15:14]	RW,0x0		0x0
DATX6	[13:12]	RW,0x0		0x0
DATX5	[11:10]	RW,0x0		0x0
DATX4	[9:8]	RW,0x0		0x0
DATX3	[7:6]	RW,0x0		0x0
DATX2	[5:4]	RW,0x0		0x0
DATX1	[3:2]	RW,0x0		0x0



DATX0	[1:0]	RW,0x0	DATX0 读 Gate 宽度调整; 00: 保持原来; 01: 向后扩展 1 个 SDRCLK;	0x0
			10: 向前扩展 1 个 SDRCLK; 11: 前后各扩展 1 个 SDRCLK;	

表 2-22: DSENWIDTH\_SEL 寄存器域的描述

名称	范围	类型	描述	默认值
保留	[31:18]	保留		
DATX17	[17:16]	RW,0x0		0x0
DATX16	[15:14]	RW,0x0		0x0
DATX15	[13:12]	RW,0x0		0x0
DATX14	[11:10]	RW,0x0		0x0
DATX13	[9:8]	RW,0x0		0x0
DATX12	[7:6]	RW,0x0		0x0
DATX11	[5:4]	RW,0x0		0x0
DATX10	[3:2]	RW,0x0		0x0
DATX9	[1:0]	RW,0x0	DATX9 读 Gate 宽度调整; 00: 保持原来; 01: 向后扩展 1 个 SDRCLK; 10: 向前扩展 1 个 SDRCLK; 11: 前后各扩展 1 个 SDRCLK;	0x0

### 2.4.13 Write Leveling 调整寄存器

2G 只有 9 个 DataX8，只需要设置 Datx0~Datx8，即只有 Rank0 和 Rank1 的 DIMM0 有意

义。表 2-23: Rank0 DIMM0 的 Write Leveling 调节 (R0\_DIMM0\_WL\_SEL)

名称	范围	类型	描述	默认值
保留	[31:18]	保留	保留	0x0
Rank0 DIMM0 108 WL PipeLine	[17:0]	RW	Rank0 DIMM0 的 WL PipeLine 调节，每个 Datx8 对应两位，[1:0]表示 DataX0。 两位 具体含义如下： 0: WL-1 1: WL 2: WL+1 3: 保留	0x15555

表 2-24: Rank0 DIMM1 的 Write Leveling 调节 (R0\_DIMM1\_WL\_SEL)

名称	范围	类型	描述	默认值
保留	[31:18]	保留	保留	0x0
Rank0 DIMM1 WL PipeLine	[17:0]	RW	Rank0 DIMM1 的 WL PipeLine 调节, 每个 Datx8 对应两位, [1:0]表示 DataX0。 两位 具体含义如下: 0: WL-1 1: WL 2: WL+1 3: 保留	0x15555

表 2-25: Rank1 DIMM0 的 Write Leveling 调节 (R1\_DIMM0\_WL\_SEL)

名称	范围	类型	描述	默认值
保留	[31:18]	保留	保留	0x0
Rank1 DIMM0 WL PipeLine	[17:0]	RW	Rank1 DIMM0 的 WL PipeLine 调节, 每个 Datx8 对应两位, [1:0]表示 DataX0。 两位 具体含义如下: 0: WL-1 1: WL 2: WL+1 3: 保留	0x15555

表 2-26: Rank1 DIMM1 的 Write Leveling 调节 (R1\_DIMM1\_WL\_SEL)

名称	范围	类型	描述	默认值
保留	[31:18]	保留	保留	0x0
Rank1 DIMM1 WL PipeLine	[17:0]	RW	Rank1 DIMM1 的 WL PipeLine 调节, 每个 Datx8 对应两位, [1:0]表示 DataX0。 两位 具体含义如下: 0: WL-1 1: WL 2: WL+1 3: 保留	0x15555

## 3 系统接口的 IO 寄存器

### 3.1 系统接口 IO 寄存器列表

#### 3.1.1 CORE\_SLEEP\_STAT

CORE\_SLEEP\_STAT 为睡眠中断状态寄存器，该寄存器用于记录睡眠中断请求处理状态。 表 3-1: CORE\_SLEEP\_STAT 寄存器域的描述

名称	范围	类型	描述
Sleep	[19:16]	RO,0	Sleep [i]为“1”表示处于睡眠过程中或正在睡眠。
Sleepy	[3:0]	RO,0	Sleepy [i]为“1”表示即将睡眠。
—	其它	—	保留。

对于每一个核心 i 而言，这两位的含义如下：

表 3-2: 状态位描述

Sleep[i]	Sleepy[i]	核心 i 的状态
0	0	核心处于运行状态。
0	1	核心处于运行状态，即将睡眠。
1	0	核心处于睡眠（维护睡眠）/唤醒过程中。
1	1	核心处于睡眠过程中（核间睡眠）。

#### 3.1.2 DEVINT\_MIS

DEVINT\_MIS 为 PCI-E0 中断、维护中断、PCI-E1、短时钟和系统错误中断、不可屏蔽中断丢失 寄存器。该寄存器记录是否发生过因中断目标睡眠或 INTEN 寄存器为全 0 而产生的中断丢失。维护和 核心可读，写清“0”。

表 3-3: DEVINT\_MIS 寄存器域的描述

名称	范围	类型	描述
SHT_MISS	[24]	RWC,0	短时钟中断丢失标志。
NMI_MISS	[21]	RWC,0	不可屏蔽中断丢失标志

SYSERR_MISS	[20]	RWC,0	故障中断丢失标志。
PCIE1_MISS	[16:12]	RWC,0	PCI-E1 中断丢失标志。同 PCI-E0。
PCIE0_MISS	[8:4]	RWC,0	PCI-E0 中断丢失标志，从高到低对应 MSI、INTD、INTC、INTB、INTA 中断。
MCU_MISS	[0]	RWC,0	维护中断丢失标志。
—	其它	—	保留。

### 3.1.3 IPU\_CTRL

IPU\_CTRL 设置 PCI-E DMA 读 NB 计数控制等。维护和核心可读写。

表 3-4: IPU\_CTRL 寄存器域的描述

名称	范围	类型	描述
PCI-E1_wrcpl_mode	[12]	RW,1	PCI-E1 的控制信号，意义同 PCI-E0_wrcpl_mode。
PCI-E1DMAR_NBCNTEN	[8]	RW,0	PCI-E1 的控制信号，意义同 PCI-E0DMAR_NBCNTEN。
PCI-E0_wrcpl_mode	[4]	RW,1	为 1，即使用脉冲信号来作为 PCI-E0 NB 计数的减 1 指示； 为 0，则此时 IPU 使用正常数据通路返回的 PCI-E0 写结束来作为 PCI-E0 NB 计数的减 1 指示。
PCI-E0DMAR_NBCNTEN	[0]	RW,0	PCI-E0 DMA 读的 NB 计数控制。当该位为“1”时，DMA 读请求发出时，NB 计数器加“1”，收到读响应，NB 计数器减“1”；该位为“0”，DMA 读请求和响应不进行 NB 计数。默认为“0”。
—	其它	—	保留。

### 3.1.4 SHORT\_CLK\_START

SHORT\_CLK\_START 为短时钟中断计数起始值寄存器。该寄存器用于指定短时钟中断计数器的初值。

表 3-5: SHORT\_CLK\_START 寄存器域的描述

名称	范围	类型	描述
SHORT_CLK_START	[63:0]	RW, 64'h1000,0000,0000,0000	短时钟中断计数起始值，最小取值范围 大于等于 8。

### 3.1.5 SHORT\_CLK\_MSK

SHORT\_CLK\_MSK 为以短时钟中断目标核心屏蔽寄存器，可读写，该寄存器用于指定不向哪些核心发送短时钟中断。

表 3-6: SHORT\_CLK\_MSK 寄存器域的描述

名称	范围	类型	描述
SHORT_CLK_MSK	[3:0]	RW,0	短时钟中断目标核心屏蔽位。
—	其它	—	保留。

### 3.1.6 MT\_INDX

MT\_INDX 为维护中断映射寄存器。该寄存器用于把维护中断映射至 64 位中断向量中的第几位。

表 3-7: MT\_INDX 寄存器域的描述

名称	范围	类型	描述
MT_INDX	[7:0]	RW,0x24	维护中断向量号。
—	其它	—	保留。

### 3.1.7 SHORT\_INDX

SHORT\_INDX 为短时钟中断映射寄存器。该寄存器用于把短时钟中断映射至 64 位中断向量中的第几位。

表 3-8: SHORT\_INDX 寄存器域的描述

名称	范围	类型	描述
SHORT_INDX	[7:0]	RW,0x25	短时钟中断向量号。
—	其它	—	保留。

### 3.1.8 INTEN

INTEN 为中断使能寄存器，软件和维护可读写。当某一位为 0 时，向对应核心发送的中断均被 丢弃，并进行记录。对该核心的核间中断、异步消息中断、维护睡眠中断悬挂在 IPU，直到该核心 可以接受中断，再进行处理。

表 3-9: INTEN 寄存器域的描述

名称	范围	类型	描述
----	----	----	----

INTEN	[3:0]	RW,0x0	中断使能位，每一位对应一个核心。如果对应位为 1，则允许向该核心发送中断请求；否则禁止中断请求。注意：每个核心的对该寄存器的写只能修改自己对应的位，维护可对所有位进
—	其它	—	保留。

### 3.1.9 LOCKEN

LOCKEN 设置在 IPU 中，用于控制核心 IO 访问，维护和核心可读写。

表 3-10: LOCKEN 寄存器域的描述

名称	范围	类型	描述
LOCKEN	[3:0]	RW,0x0	为“1”表示 IO 访问需要等 IPU 里的 Lock 标志为 0 才可以发往目标；为“0”不需要关心 IPU 里的
—	其它	—	保留。

### 3.1.10 FMT\_ERR

FMT 记录收到片上网络或者设备的消息包的格式是否有错误，以及包头和数据 ECC 单错和多错的情况，维护和核心可读，写清“0”。

表 3-11: FMT\_ERR 寄存器域的描述

名称	范围	类型	描述
PCIE1_CPL_ERR	[28]	RWC,0x0	收到 PIU1 的响应发生包格式错误。
PCIE1_REQ_ERR	[24]	RWC,0x0	收到 PIU1 的请求发生包格式错误。
PCIE0_CPL_ERR	[20]	RWC,0x0	收到 PIU0 的响应发生包格式错误。
PCIE0_REQ_ERR	[16]	RWC,0x0	收到 PIU0 的请求发生包格式错误。
MCU_CPL_ERR	[12]	RWC,0x0	收到 MCU 的响应发生包格式错误。
MCU_REQ_ERR	[11:8]	RWC,0x0	MCU_REQ_ERR[3]: IO 读越界； MCU_REQ_ERR[2]: IO 写越界； MCU_REQ_ERR[1]: 收到 MCU 的 IO 写请求发生 ECC 多错； MCU_REQ_ERR[0]: 收到 MCU 的请求发生包格式错误。

CPM_CPL_ERR	[4]	RWC,0x0	CPM_CPL_ERR[0]: 收到 CPM 的响应发生包格式错误。
CPM_REQ_ERR	[3:0]	RWC,0x0	CPM_REQ_ERR[3]: IO 读越界; CPM_REQ_ERR[2]: IO 写越界; CPM_REQ_ERR[1]: 收到 CPM 的 IO 写请求发生数据 ECC 多错; CPM_REQ_ERR[0]: 收到 CPM 的请求发生包格式错误。
—	其它	—	保留。

### 3.1.11 SYSERR\_INDX

SYSERR\_INDX 为系统故障中断映射寄存器。该寄存器用于把系统故障中断映射至 64 位中断向量中的第几位。

表 3-12: SYSERR\_INDX 寄存器域的描述

名称	范围	类型	描述
SYSERR_INDX	[7:0]	RW,0x26	系统故障中断向量号。
—	其它	—	保留。

### 3.1.12 NMI\_INDX

NMI\_INDX 为不可屏蔽中断映射寄存器。该寄存器用于把不可屏蔽中断映射至 64 位中断向量中的第几位。

表 3-13: NMI\_INDX 寄存器域的描述

名称	范围	类型	描述
NMI_INDX	[7:0]	RW,0x28	不可屏蔽中断向量号。
—	其它	—	保留。

### 3.1.13 IPU\_STAT

IPU\_STAT 为缓冲读写控制错状态寄存器，维护和核心可读，写清 0。

表 3-14: IPU\_STAT 寄存器域的描述

名称	范围	类型	描述
NBOverFlow	[19:18]	RWC,0	NB 计数器溢出标志： 高位为 PIU1，低位为 PIU0。
OMCBetyRdErr	[15]	RWC,0	OMCB 空时读错误。
OMRBetyRdErr	[12]	RWC,0	OMRB 空时读错误。
IMCBFullWrErr	[9]	RWC,0	IMCB 满时写入错误。
IMRBFullWrErr	[6]	RWC,0	IMRB 满时写入错误。
—	其它	—	保留。

### 3.1.14 SERR\_CNTTH

IPU 单错预警阈值，维护和核心可读写。

表 3-15: SERR\_CNTTH 寄存器域的描述

名称	范围	类型	描述
SerrCntTH	[31:0]	RW,0xffff,ffff	IPU 单错预警阈值。如果单错统计计数器的计数值大于该阈值且对应的单错报错使能位打开，则 IPU 向 MCU 报 SERR 错误。单错统计计数器共有：MCUSERR_CNT、PIU1SERR_CNT、PIU0SERR_CNT 和 FXSerrCnt。
—	其它	—	保留。

### 3.1.15 MCUSERR\_CNT

该寄存器统计：IPU 接收到 MCU 方向请求和响应包的包头和数据发生 ECC 单错的次数。  
维护和核心可读，写清“0”。

表 3-16: MCUSERR\_CNT 寄存器域的描述

名称	范围	类型	描述
MCUSERR_CNT	[31:0]	RWC,0x0	每发生一次 ECC 单错，计数器加“1”。
—	其它	—	保留。



### 3.1.16 PIU0SERR\_CNT

该寄存器统计：IPU 接收到 PIU0 方向请求和响应包的包头和数据发生 ECC 单错的次数。维护 和核心可读，写清“0”。

表 3-17: PIU0SERR\_CNT 寄存器域的描述

名称	范围	类型	描述
PIU0SERR_CNT	[31:0]	RWC,0x0	每发生一次 ECC 单错，计数器加“1”。
—	其它	—	保留。

### 3.1.17 PIU1SERR\_CNT

该寄存器统计：IPU 接收到 PIU1 方向请求和响应包的包头和数据发生 ECC 单错的次数。维护 和核心可读，写清“0”。

表 3-18: PIU1SERR\_CNT 寄存器域的描述

名称	范围	类型	描述
PIU1SERR_CNT	[31:0]	RW,0x0	每发生一次 ECC 单错，计数器加“1”。
—	其它	—	保留。

### 3.1.18 FXSERR\_CNT

该寄存器统计：IPU 接收到 CPM 方向请求和响应包的包头和数据发生 ECC 单错的次数。维护 和核心可读，写清“0”。

表 3-19: FXSERR\_CNT 寄存器域的描述

名称	范围	类型	描述
FXSERR_CNT	[31:0]	RWC,0x0	每发生一次 ECC 单错，计数器加“1”。
	其它	—	保留。

### 3.1.19 ERRRPT\_EN

该寄存器和 FMT\_ERR 的报错位一一对应，ERRRPT\_EN 中的某 bit 为“1”表示 FMT\_ERR 中 相应的 bit 报错使能。维护和核心可读写，复位为全“0”。

表 3-20: ERRRPT\_EN 寄存器域的描述

名称	范围	类型	描述
PCIE1_CPL_ERREn	[28]	RW,0x0	收到 PIU1 的响应发生包格式错误报错使能。
PCIE1_REQ_ERREn	[24]	RW,0x0	收到 PIU1 的请求发生包格式错误报错使能。

PCIE0_CPL_ERREn	[20]	RW,0x0	收到 PIU0 的响应发生包格式错误报错使能。
PCIE0_REQ_ERREn	[16]	RW,0x0	收到 PIU0 的请求发生包格式错误报错使能。
MCU_CPL_ERREn	[12]	RW,0x0	收到 MCU 的响应发生包格式错误报错使能。
MCU_REQ_ERREn	[11:8]	RW,0x0	MCU_REQ_ERR[3]: IO 读越界报错使能; MCU_REQ_ERR[2]: IO 写越界报错使能; MCU_REQ_ERR[1]: 收到 MCU 的 IO 写请求发
			ECC 多错报错使能。 MCU_REQ_ERR[0]: 收到 MCU 的请求发生包格式错误报错使能。
CPM_CPL_ERREn	[4]	RW,0x0	CPM_CPL_ERR[0]: 收到 CPM 的响应发生包格式错误报错使能。
CPM_REQ_ERREn	[3:0]	RW,0x0	CPM_REQ_ERR[3]: IO 读越界报错使能; CPM_REQ_ERR[2]: IO 写越界报错使能; CPM_REQ_ERR[1]: 收到 CPM 的 IO 写请求发生数据 ECC 多错报错使能; CPM_REQ_ERR[0]: 收到 CPM 的请求发生包格式
—	其它	—	保留。

### 3.1.20 SYSERR\_NMI\_INT\_DST

SYSERR\_NMI\_INT\_DST 为系统故障和不可屏蔽中断的目标配置寄存器。

表 3-21: SYSERR\_NMI\_INT\_DST 寄存器域的描述

名称	范围	类型	描述
NMIINT_DSTEN	[5]	RW,0	复位为“0”；当该位为“0”时，中断目标在可接收中断的核心中轮转；当该位为“1”时，中断目标为 NMIINT_DST 所指定的核心，如果 NMIINT_DST 所指定的核心不可接收中断，则从其它可接收中断的核心中选择一个作为中断

SYSERRINT_DSTEN	[4]	RW,0	复位为“0”；当该位为“0”时，中断目标在可接收中断的核心中轮转；当该位为“1”时，中断目标为 SYSERRINT_DST 所指定的核心，如果 SYSERRINT_DST 所指定的核心不可接收中断，则从其它可接收中断的核心中选择一个作
NMIINT_DST	[3:2]	RW,0	编码表示不可屏蔽中断目标核心心号，复位值为 1'b0。
SYSERRINT_DST	[1:0]	RW,0	编码表示故障中断目标核心心号，复位值为 1'b0。
—	其它	—	保留。

### 3.1.21 IPU\_STAT\_ERREN

IPU\_STAT\_ERREN 为 IPU\_STAT 一一对应，是 IPU\_STAT 报错使能寄存器。表 3-22: IPU\_STAT\_ERREN 寄存器域的描述

名称	范围	类型	描述
NBOverFlowEn	[19:18]	RW,0	NB 计数器溢出报错使能：高位为 PIU1，低位为 PIU0。
OMCBetyRdErrEn	[15]	RW,0	OMCB 空时读错误报错使能。
OMRBetyRdErrEn	[12]	RW,0	OMRB 空时读错误报错使能。
IMCBFullWrErrEn	[9]	RW,0	IMCB 满时写入错误报错使能。
IMRBFullWrErrEn	[6]	RW,0	IMRB 满时写入错误报错使能。
—	其它	—	保留。

### 3.1.22 PCIE0\_INTx\_INDx

PCIE0\_INTx\_INDx 为 PCI-E0 的 INTx 中断映射寄存器。该寄存器用于把 INTa—INTd 中断映射至 64 位中断向量中的第几位。

表 3-23: PCIE0\_INTx\_INDx 寄存器域的描述

名称	范围	类型	描述
----	----	----	----

INTx_INDx	[7:0]	RW, 0x20, 0x21, 0x22, 0x23	PCI-E0 的 INTx 中断向量号, INTa-d 对应复位的初值为 0x20-0x23。
—	其它	—	保留。

### 3.1.23 PCIE1\_INTx\_INDx

PCIE1\_INTx\_INDx 为 PCI-E1 的 INTx 中断映射寄存器。该寄存器用于把 INTa—INTd 中断映射至 64 位中断向量中的第几位。

表 3-24: PCIE1\_INTx\_INDx 寄存器域的描述

名称	范围	类型	描述
INTx_INDx	[7:0]	RW, 0x1C, 0x1D, 0x1E, 0x1F	PCI-E1 的 INTx 中断向量号, INTa-d 对应复位的初值为 0x1C-0x1F。
—	其它	—	保留。

### 3.1.24 IINT\_MIS

IINT\_MIS 设置在 IPU 中, 用于记录目标核心的核间中断丢失情况。该寄存器软件和维护可读写, 硬件实现为写清 0。如果核间中断 0、核间中断 1 的目标核心断连, 或者睡眠中断的目标核心不能接收中断, 则由硬件隐式写 IINT\_MIS 对应位。

表 3-25: IINT\_MIS 寄存器域的描述

名称	范围	类型	描述
SlpMIS	[35:32]	RWC,0x0	比特 32~33 分别对应核心 0~3 的睡眠中断丢失情况, 为“1”表示对应的目标核心发生中断丢失
CoreInt1MIS	[19:16]	RWC,0x0	比特 16~19 分别对应核心 0~3 的核间中断 1 丢失情况, 为“1”表示对应的目标核心发生中断丢失

CoreInt0MIS	[3:0]	RWC,0x0	比特 0~3 分别对应核心 0~3 的核间中断 0 丢失情况，为“1”表示对应的目标核心发生中断丢失
—	其它	—	保留。

### 3.1.25 IINTMIS\_RPTEN

IINTMIS\_RPTEN 设置在 IPU 中，用于 IOR: IINT\_MIS 的报错使能。该寄存器软件和维护可读写。

表 3-26: IINTMIS\_RPTEN 寄存器域的描述

名称	范围	类型	描述
SlpMIS_RPTEN	[35:32]	RW,0x0	比特 32~35 分别对应核心 0~3 的睡眠中断丢失情况的报错使能，为“1”表示如果对应的目标核
CoreInt1MIS_RPTEN	[19:16]	RW,0x0	比特 16~19 分别对应核心 0~3 的核间中断 1 丢失情况的报错使能，为“1”表示如果对应的目标核心发生中断丢失，可以向维护报错。
CoreInt0MIS_RPTEN	[3:0]	RW,0x0	比特 0~3 分别对应核心 0~3 的核间中断 0 丢失情况的报错使能，为“1”表示如果对应的目标核
—	其它	—	保留。

### 3.1.26 DEVINT\_MIS\_RPTEN

DEVINT\_MIS\_RPTEN 为 IOR: DINT\_MIS 中 PCI-E0 中断、维护中断、PCI-E1 中断、故障中断、不可屏蔽中断丢失情况的报错使能。维护 and 核心可读写。

表 3-27: DEVINT\_MIS\_RPTEN 寄存器域的描述

名称	范围	类型	描述
NMI_MISS_RPTEN	[21]	RW,0	不可屏蔽中断丢失报错使能。
SYSERR_MISS_RPTEN	[20]	RW,0	故障中断丢失报错使能。
PCIE1_MISS_RPTEN	[16:12]	RW,0	PCI-E1 中断丢失报错使能。同 PCI-E0。
PCIE0_MISS_RPTEN	[8:4]	RW,0	PCI-E0 中断丢失报错使能，从高到低对应 MSI、INTD、INTC、INTB、INTA 中断。

MCU_MISS_RPTEN	[0]	RW,0	维护中断丢失报错使能。
—	其它	—	保留。

### 3.1.27 ECCSERR

ECCSERR 为记录 IPU 模块发生 ECC 单错的情况。维护和核心可读写，写清“0”。

表 3-28: ECCSERR 寄存器域的描述

名称	范围	类型	描述
CntPIU1SErr	[3]	RWC,0	IPU 收到 PIU1 包头和数据单错达到阈值。
CntPIU0SErr	[2]	RWC,0	IPU 收到 PIU0 包头和数据单错达到阈值。
CntMCUSErr	[1]	RWC,0	IPU 收到 MCU 包头和数据单错达到阈值。
CntFXSErr	[0]	RWC,0	IPU 收到 CPM 包头和数据单错达到阈值。
—	其它	—	保留。

### 3.1.28 ECCSERR\_RPTEN

ECCSERR\_RPTEN 为记录 IPU 模块发生 ECC 单错的报错使能寄存器。维护和核心可读写。

表 3-29: ECCSERR\_RPTEN 寄存器域的描述

名称	范围	类型	描述
CntPIU1SErrEn	[3]	RW,0	IPU 收到 PIU1 包头和数据单错达到阈值报错使能。
CntPIU0SErrEn	[2]	RW,0	IPU 收到 PIU0 包头和数据单错达到阈值报错使能。
CntMCUSErrEn	[1]	RW,0	IPU 收到 MCU 包头和数据单错达到阈值报错使能。
CntFXSErrEn	[0]	RW,0	IPU 收到 CPM 包头和数据单错达到阈值报错使能。
—	其它	—	保留。

### 3.1.29 ECCMERR

ECCMERR 为记录 IPU 模块发生 ECC 多错的情况。维护和核心可读写，写清“0”。

表 3-30: ECCMERR 寄存器域的描述

名称	范围	类型	描述
CntPIU1MErr	[3]	RWC,0	IPU 收到 PIU1 包头和数据发生多错。
CntPIU0MErr	[2]	RWC,0	IPU 收到 PIU0 包头和数据发生多错。
CntMCUMErr	[1]	RWC,0	IPU 收到 MCU 包头和数据发生多错。
CntFXMErr	[0]	RWC,0	IPU 收到 CPM 包头和数据发生多错。

—	其它	—	保留。
---	----	---	-----

### 3.1.30 ECCMERR\_RPTEN

ECCMERR\_RPTEN 为记录 IPU 模块发生 ECC 多错的报错使能寄存器。维护和核心可读写。 表 3-31: ECCMERR\_RPTEN 寄存器域的描述

名称	范围	类型	描述
CntPIU1MErrEn	[3]	RW,0	IPU 收到 PIU1 包头和数据发生多错报错使能。
CntPIU0MErrEn	[2]	RW,0	IPU 收到 PIU0 包头和数据发生多错报错使能。
CntMCUMErrEn	[1]	RW,0	IPU 收到 MCU 包头和数据发生多错报错使能。
CntFXMErrEn	[0]	RW,0	IPU 收到 CPM 包头和数据发生多错报错使能。
—	其它	—	保留。

### 3.1.31 DEVINT\_WKEN

DEVINT\_WKEN 为 PCI-E0/1 中断、维护中断的睡眠唤醒使能寄存器。维护和核心可读写。 表 3-32: DEVINT\_WKEN 寄存器域的描述

名称	范围	类型	描述
PCIE1_WKEN	[16:12]	RW,0	PCI-E1 中断唤醒使能。含义同 PCI-E0。
PCIE0_WKEN	[8:4]	RW,0	[8:4]: PCI-E0 中断唤醒使能, 从高到低对应 MSI、INTD、INTC、INTB、INTA 中断。当收到 PCI-E0 中断, 且目标核心睡眠且 INTEN 使能, 如果对应位为“1”, 则登记 IOR: DEVINT_MIS 的同时, 唤醒该
MCU_WKEN	[0]	RW,0	维护中断唤醒使能。当收到普通维护中断, 且对应目标核心睡眠且 INTEN 使能, 如果该位为“1”, 则登记 IOR: DEVINT_MIS 的同时, 唤醒该目标核
—	其它	—	保留。

### 3.1.32 MAIL\_BOX\_i

MAIL\_BOX\_i (i: 0~3) 异步消息队列, 该队列为 64b×16 的 FIFO 结构, 记录其它核心 (包括 核心 i) 发给核心 i 的异步中断短消息。

如果其它核心（包括核心 i）对 IOR: MAIL\_BOX\_i 进行 IO 写，会触发对核心 i 的异步消息中断，同时，IO 写携带的数据（中断短消息）缓存入异步消息队列中。核心 i 在适当时刻读该队列的中断短消息，从而达到核心间通信的目的。

MAIL\_BOX\_i 只有核心 i 可以读，所有核心可写；维护不可读写。复位时，所有队列为空。

### 3.1.33 MAIL\_STAT\_i

MAIL\_STAT\_i (i: 0~3) 为异步消息队列状态寄存器。维护和核心只读。具体定义如下表：

表 3-33: MAIL\_STAT\_i 寄存器域的描述

名称	范围	类型	描述
MAIL_NUM	[20:16]	RO,0	该目标核心的收到的邮件数量。每个核心最多可接收 16 个异步消息。
FULL_FLAG	[3:0]	RO,0	某位为“1”则表示某个源核心对该目标核心发送异步消息时，邮箱已满。
—	其它	—	保留。

当核心 n 想往核心 m 发异步消息时，可通过写 IOR: MAIL\_BOX\_m 发送一个 64 位的消息，系统接口收到后，如果 IOR: MAIL\_STAT\_m[MAIL\_NUM] 指示邮箱不满，则将消息保存，系统接口自动向核心 m 发一个核间异步消息中断请求。如果 IOR: MAIL\_STAT\_m[MAIL\_NUM] 指示邮箱满，则置 IOR: MAIL\_STAT\_m[FULL\_FLAG[n]] 为 1，指示本次消息丢失。软件必须保证源核心每发一次消息，都要检查 IOR: MAIL\_STAT\_m[FULL\_FLAG[n]] 是否为“1”，如果为“1”，则说明刚发出的消息不成功，需要重发。

### 3.1.34 PCIE0\_MSI[i]\_INDX

PCIE0\_MSI[i]\_INDX (i: 0~255) 为 PCI-E0 的 MSI 中断映射 REG，维护和核心可读写，该寄存器用于把 MSI[i] 中断映射至中断向量中的第几位。《申威 411 处理器管理员手册》附录 B 描述了除 PCI-E MSI 中断外其它中断向量位置，具体与 IBOX\_CSR: INT\_VEC，以及 IPU 的 IOR: SYSERR\_INDX、SHORT\_INDX、MT\_INDX、PCIE0\_INTx\_INDX 和 PCIE1\_INTx\_INDX 一一对应。附录 B 描述中的其它保留域，或者 INT\_STAT1 和 INT\_STAT2 可用于指定 PIC-E0 或 PCI-E1 的 MSI 中断位置。软件在配置 PCI-E 时需要配置该 IO 寄存器以指定中断位置，注意不要影响附录 B 中已经定义各域。

表 3-34: PCIE0\_MSI[i]\_INDX 寄存器域的描述

名称	范围	类型	描述
MSI[i]_INDX	[7:0]	RW,i	PCI-E0 的 MSI 中断向量号。



—	其它	—	保留。
---	----	---	-----

### 3.1.35 PCIE1\_MSI[i]\_INDX

PCIE1\_MSI[i]\_INDX (i: 0~255) 为 PCI-E1 的 MSI 中断映射 REG，维护和核心可读写，该寄存器用于把 MSI[i] 中断映射至中断向量中的第几位。《申威 411 处理器管理员手册》附录 B 描述了除 PCI-E MSI 中断外其它中断向量位置，具体与 IBOX CSR: INT\_VEC，以及 IPU 的 IOR : SYSERR\_INDXX、SHORT\_INDXX、MT\_INDXX、PCIE0\_INTx\_INDXX 和 PCIE1\_INTx\_INDXX 一一对应。附录 B 描述中的其它保留域，或者 INT\_STAT1 和 INT\_STAT2 可用于指定 PIC-E0 或 PCI-E1 的 MSI 中断位置。软件在配置 PCI-E 时需要配置该 IO 寄存器以指定中断位置，注意不要影响附录 B 中已经定义各域。

表 3-35: PCIE1\_MSI[i]\_INDX 寄存器域的描述

名称	范围	类型	描述
MSI[i]_INDX	[7:0]	RW,i	PCI-E1 的 MSI 中断向量号。
—	其它	—	保留。

## 4 维护接口的 IO 寄存器

### 4.1 维护 IOR 的复位说明

MCU 受上电复位控制，MCU 的复位状态即主状态机处于上电复位。

### 4.2 维护 IOR 的寄存器说明

#### 4.2.1 MC\_ONLINE

MC\_ONLINE 为存控在位使能寄存器，维护可读写，软件只读。该寄存器用于表示存储控制器 是否在位，复位时初值为“0x3”。

表 4-1: MC\_ONLINE 寄存器域的描述

名称	范围	类型	描述
MC_ONLINE	[1:0]	RW,0x3	MC_ONLINE[i]为“1”，表示该存控在位。
—	其它	—	保留。

注：MC\_ONLINE 对于存控时钟和复位的控制是直接控制，即对该 IOR 的修改直接影响存控时 钟的高低频选择和复位。

#### 4.2.2 CG\_FAULT\_STAT

CG\_FAULT\_STAT 为核组总错状态寄存器，维护和软件可读写。该寄存器用于存放各个核组 的 总错信息。复位时初值为全“0”。采上升沿登记；按位写“1”清除。

表 4-2: CG\_FAULT\_STAT 寄存器域的描述

名称	范围	类型	描述
----	----	----	----

MC1_FAULT	[31:28]	RW1C,0	[28]: MC1 主存读数据产生单错预警; [29]: MC1 主存读数据有多错; [30]: MC1 写主存数据产生单错预警; [31]: MC1 写主存数据产生多错, 包括请求通路的偶校验错。
SOFTERR	[27:24]	RW1C,0	分别表示对应 4 个核心的软件错误。根据相应的 IOR:
MC0_FAULT	[23:20]	RW1C,0	[20]: MC0 主存读数据产生单错预警; [21]: MC0 主存读数据有多错; [22]: MC0 写主存数据产生单错预警; [23]: MC0 写主存数据产生多错, 包括请求通路的偶校验错。
CPM_FAULT	[19:16]	RW1C,0	[16]: CPM 产生控制错误、一致性错等硬件故障; [17]: CPM 单错预警, 含 CTAG 和传输通路; [18]: CPM 多错, 含 CTAG 和传输通路; [19]: 保留。

Core_FAULT	[15:0]	RW1C,0	<p>分别记录每个核组内各个核心的错误信息，具体含义如下： [0]：核心 0 产生机器检查错故障（Rob 检查错、Dbox 检查错、Sbox 检查错）该位置 1；如果软件也设置了相应 CSR： HARD_ERR， 该位也为 1；</p> <p>[1]：核心 0 产生单错预警； [2]：核心 0 产生多错；</p> <p>[3]：核心 0 产生软件故障（写地址越界， 或者如果软件设置了相应 CSR： SOFT_ERR， 则该位也为 1）；</p> <p>[4]：核心 1 产生机器检查错故障（同上）； [5]：核心 1 产生单错预警；</p> <p>[6]：核心 1 产生多错； [7]：核心 1 产生软件故障（同上）。 [8]：核心 2 产生机器检查错故障（同上）； [9]：核心 2 产生单错预警； [10]：核心 2 产生多错； [11]：核心 2 产生软件故障（同上）。 [12]：核心 3 产生机器检查错故障（同上）； [13]：核心 3 产生单错预警； [14]：核心 3 产生多错； [15]：核心 3 产生软件故障（同上）。</p>
—	其它	—	保留。

### 4.2.3 CG\_FAULT\_EN

CG\_FAULT\_EN 为核组总错状态报错使能寄存器，维护和软件都可读写。该寄存器用于指示

CG<sub>x</sub>\_FAULT\_STAT 中那些错误标志位参与总错信号的生成。复位时初值为全“0”。

表 4-3: CG\_FAULT\_EN 寄存器域的描述

名称	范围	类型	描述
MC1_FAULT_EN	[31:28]	RW,0	[28]: MC1 主存读数据产生单错预警报错使能; [29]: MC1 主存读数据有多错报错使能; [30]: MC1 写主存数据产生单错预警报错使能;
SOFTERR_EN	[27:24]	RW,0	分别表示对应 4 个核心的软件错误报错使能。
MC0_FAULT_EN	[23:20]	RW,0	[20]: MC0 主存读数据产生单错预警报错使能; [21]: MC0 主存读数据有多错报错使能; [22]: MC0 写主存数据产生单错预警报错使能;
CPM_FAULT_EN	[19:16]	RW,0	[16]: CPM 产生控制错误、一致性错等硬件故障报错使能; [17]: CPM 单错预警, 含 CTAG 和传输通路报错使能; [18]: CPM 多错, 含 CTAG 和传输通路报错使能; [19]: 保留。

Core_FAULT_EN	[15:0]	RW,0	<p>分别记录每个核组内各个模块的错误信息，具体含义如下：</p> <p>[0]：核心 0 产生机器检查错故障报错使能； [1]：核心 0 产生单错预警报错使能； [2]：核心 0 产生多错报错使能； [3]：核心 0 产生软件故障报错使能； [4]：核心 1 产生机器检查错故障报错使能； [5]：核心 1 产生单错预警报错使能； [6]：核心 1 产生多错报错使能； [7]：核心 1 产生软件故障报错使能。 [8]：核心 2 产生机器检查错故障报错使能； [9]：核心 2 产生单错预警报错使能； [10]：核心 2 产生多错报错使能； [11]：核心 2 产生软件故障报错使能。</p>
---------------	--------	------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

			[12]: 核心 3 产生机器检查错故障报错使 能; [13]: 核心 3 产生单错预警报错使能; [14]: 核心 3 产生多错报错使能; [15]: 核心 3 产生软件故障报错使能。
—	其它	—	保留。

## 4.2.4 CG\_FAULT\_INT\_EN

CG\_FAULT\_INT\_EN 为核组故障中断使能寄存器，维护和软件都可读写。该寄存器用于指示

CG\_FAULT\_STAT 中那些错误标志位参与故障中断的生成。复位时初值为全“0”。

表 4-4: CG\_FAULT\_INT\_EN 寄存器域的描述

名称	范围	类型	描述
MC1_FAULT_INT_EN	[31:28]	RW,0	[28]: MC1 主存读数据产生单错预警中断使 能; [29]: MC1 主存读数据有多错中断使能; [30]: MC1 写主存数据产生单错预警中断使 能;
SOFTERR_INT_EN	[27:24]	RW,0	分别表示对应 4 个核心的软件错误中断使能。
MC0_FAULT_INT_EN	[23:20]	RW,0	[20]: MC0 主存读数据产生单错预警中断使 能; [21]: MC0 主存读数据有多错中断使能; [22]: MC0 写主存数据产生单错预警中断使 能;
CPM_FAULT_INT_EN	[19:16]	RW,0	[16]: CPM 产生控制错误、一致性错等硬件故障中断 使能; [17]: CPM 单错预警, 含 CTAG 和传输通路中 断使能; [18]: CPM 多错, 含 CTAG 和传输通路中 断使能; [19]: 保留。

Core_FAULT_INT_EN	[15:0]	RW,0	<p>分别记录每个核组内各个模块的错误信息，具体含义如下：</p> <p>[0]：核心 0 产生机器检查错故障中断使能； [1]：核心 0 产生单错预警中断使能；</p> <p>[2]：核心 0 产生多错中断使能； [3]：核心 0 产生软件故障中断使能； [4]：核心 1 产生机器检查错故障中断使能；</p> <p>[5]：核心 1 产生单错预警中断使能；</p>
-------------------	--------	------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------



			[6]: 核心 1 产生多错中断使能; [7]: 核心 1 产生软件故障中断使能。 [8]: 核心 2 产生机器检查错故障中断使能; [9]: 核心 2 产生单错预警中断使能; [10]: 核心 2 产生多错中断使能; [11]: 核心 2 产生软件故障中断使能。 [12]: 核心 3 产生机器检查错故障中断使能; [13]: 核心 3 产生单错预警中断使能; [14]: 核心 3 产生多错中断使能; [15]: 核心 3 产生软件故障中断使能。
—	其它	—	保留。

## 4.2.5 INIT\_CTL

INIT\_CTL 是初始化状态机控制寄存器，维护可读写，软件只读。该寄存器用来控制复位初始化流程，复位时初值为全“0”。

表 4-5: INIT\_CTL 寄存器域的描述

名称	范围	类型	描述
RSTENT_EN	[41]	RW,0x1	该位为“1”，表示需要在进入复位状态进行正常延时等待。
STRUN_EN	[40]	RW,0x1	该位为“1”，表示需要在启动运行状态进行运行。
ICLD_EN	[39]	RW,0x1	该位为“1”，表示需要在初始化加载状态进行加载。
CFG3_EN	[38]	RW,0x1	该位为“1”，表示需要在配置 3 状态进行配置。
BIST_EN	[37]	RW,0x1	该位为“1”，表示需要在存储器自测试状态进行测试。
CFG2_EN	[36]	RW,0x1	该位为“1”，表示需要在配置 2 状态进行配置。
RSTEND_EN	[35]	RW,0x1	该位为“1”，表示需要在结束复位状态进行正常延时等待。
CLKCHG_EN	[34]	RW,0x1	该位为“1”，表示需要在时钟切换 1/2 状态进行正常切换。
PLLUP_EN	[33]	RW,0x1	该位为“1”，表示需要在 PLL 升频状态等待升频结束。
CFG1_EN	[32]	RW,0x1	该位为“1”，表示需要在配置 1 状态进行配置。
MM_FREQ	[19:16]	RW,0xa	内存频率配置。在配置 1 状态下，若引脚 CFG_SEL_L 为“0”，则修改为引脚 CFG_XBX_H[2:0] 的值（注意：引脚对应寄存器的高三位，寄存器的最低位此时为零）

XBX_FREQ	[11:8]	RW,0xa	互连频率配置。在配置 1 状态下，若引脚 CFG_SEL_L 为“0”，则修改为引脚 CFG_XBX_H[2:0] 的值（注意：引脚对应寄存器的高三位，寄存器的最低位此时为零）
----------	--------	--------	------------------------------------------------------------------------------------------

CORE_FREQ	[4:0]	RW,0x1 a	核心频率配置。在配置 1 状态下，若引脚 CFG_SEL_L 为“0”， 则修改为引脚 CFG_XBX_H[2:0]的值（注意：引脚对应
—	其它	—	保留。

在引脚 CFG\_SEL\_L 为零时，硬件会根据 INIT\_MODE\_H 以及表 4-6 对 INIT\_CTL 进行隐式写，具体见表 4-7。

表 4-6: INIT\_MODE\_H 说明

INIT_MODE_H[1:0] 初始化状态机状态	2'b00	2'b01	2'b10	2'b11
配置 1 状态	无操作			可读写维护控制器的 I/O 寄存器
PLL 升频状态	无操作			延时等待
时钟切换状态	快速进行时钟切换			正常进行时钟切换
结束复位状态	无操作			延时等待
配置 2 状态	停在此状态	无操作	可读写维护控制器的 I/O 寄存器和配置存储器自测试相关的扫描链寄存器	
存储器自测试状态	—	无操作	进行存储器自修复测试	
配置 3 状态	—	无操作	可读写所有 I/O 寄存器	
初始化加载状态	—	无操作	进行初始化程序加载	
启动运行状态	—	停在此状态	延时等待	
应用场景	测试模拟	程序模拟	运行模拟	正常运行

表 4-7: INIT\_CTL 隐式写说明

INIT_MODE_H INIT_CTL	2'b00	2'b01	2'b10	2'b11
RSTENT_EN	0	0	0	1
STRUN_EN	0	0	1	1
ICLD_EN	0	0	1	1
CFG3_EN	0	0	1	1
BIST_EN	0	0	1	1
CFG2_EN	1	0	1	1
RSTEND_EN	0	0	0	1
CLKCHG_EN	0	0	0	1
PLLUP_EN	0	0	0	1



CFG1_EN	0	0	0	1
---------	---	---	---	---

## 4.2.6 BIST\_CTL

BIST\_CTL 是 BIST 控制寄存器，维护可读写，软件只读。该寄存器用来控制复位初始化流程，复位时初值为全“0”。

表 4-8: BIST\_CTL 寄存器域的描述

名称	范围	类型	描述
SAEN_SEL_TData	[27:26]	RW,0x0	对应三级 Cache 的数据部分和 Tag 部分的 SAEN_SEL 的
SAEN_SEL_CORE	[23:22]	RW,0x0	对应核心的 SAEN_SEL 的值。
SAEN_SEL_EN	[21]	RW,0x0	指定 SAEN 值的有效位。该功能只针对 BIST 测试的新增需求（在不扫入的情况指定 SAEN 进行 BIST 或 BISR 的测试）。即上电复位冷复位时，把该寄存器配置为“1”，并配置 SAEN_SEL_CORE/SAEN_SEL_CPM，且 BistMode 配置为 BIST 或 BISR 模式，则相应的阵列（可统一测试）进行指定 SAEN 的测试。 注 1: 各阵列的 SAEN 的值指定相同；注 2: 正常流程时该位“0”；注 3: 配置该寄存器其测试类型必须是 BIST 或 BISR，否则为非法配置。
SAEN_SEL_CTRL	[20]	RW,0x0	指示 SAEN 选择策略：为“0”则按照可靠性优先原则，否则为性能优先原则。按照可靠性选择原则时，灵敏放大器配置值从 3、2、1 到 0 的顺序选择可通过测试的配置；否则从 0、1、2 到 3 的顺序选择可通过测试的配置。
MEMIDValid	[19]	RW,0x0	当进行存储器自测试时有意义，表示对指定阵列进行测试。阵列编号由 CGID 和 MEMID 指定，BIST 类型由 BistMode 指定。主要用于对指定阵列的 debug 测试。
CGID	[18:17]	RW,0x0	CG 编号；当指定阵列测试时有意义。
MEMID	[16:8]	RW,0x0	指定阵列编号；当指定阵列测试时有意义。 [16:14]没有意义； [13]=1'b1，表示对三级 Cache 或三级 Tag 或 PCACHE0 或 PCACHE1 进行 Bist 测试；

			<p>[12:11]定义如下：</p> <p>2'b00: 表示 TCache 或 TTTAG;</p> <p>2'b10: 表示 PCache0;</p> <p>2'b11: 表示 PCache1;</p> <p>当[12:11]为 2'b00 时, [10:8]定义如下：</p> <p>3'b000: 表示 TTag;</p> <p>3'b001: 表示 TCache;</p> <p>其它值: 保留。</p> <p>当[12:11]为 2'b10 或 2'b11 时, [10:8]定义如下：</p> <p>3'b001: 表示 Pache;</p> <p>其它值: 保留。</p> <p>[13]=1'b0, 表示对核心进行 Bist 测试; 核心号由[12:11]指定;</p> <p>2'b00: 核心 0;</p> <p>2'b01: 核心 1;</p> <p>2'b10: 核心 2;</p> <p>2'b11: 核心 3;</p> <p>核心内的阵列号由[10:8]指定;</p> <p>3'b001: 表示 ICache;</p> <p>3'b011: 表示 DCache;</p> <p>3'b100: 表示 Stag;</p> <p>3'b101: 表示 SCache;</p> <p>其它值: 保留。</p>
<p>BistMode</p>	<p>[6:4]</p>	<p>RW,0x6</p>	<p>Bist 测试类型; 当指定阵列测试和全阵列测试都有意义。</p> <p>“3'b0xx” : Debug 测试 (特殊的 BIST 自测试, 当发现错误时, 暂停测试, 将错误信息锁存到一个可扫描的寄存器中, 此模式下只能对指定的存储器进行), [1:0]为敏感放大器设置值; “3'b100” : 使用之前保留的灵敏放大器配置进行 BIST 初始化 (该配置可以通过 Scan 扫入配置); “3'b101” : 使用之前保留的灵敏放大器配置进行 BISR 自修复测试 (该配置可以通过 Scan 扫入配置); “3'b11x” : 自动迭代的灵敏放大器配置进行 BISR 自修复</p>

			测试。注：Debug 测试只有在指定阵列测试时有意义；非指定阵列测试时，debug 模式强制成 BIST、BISR 或 ScanBISR（即直接把 BistMode[2]强制成“1”）；
HOTRST_CTL[1: 0]	[3:2]	RW,0x0	0x: 表示核心在唤醒以及热复位进行保留原 SAEN 的 BIST 测试； 10: 表示核心在唤醒以及热复位时进行保留原 SAEN 的 BISR 测试； 11: 表示核心在唤醒以及热复位时进行 SAEN 自动迭代的 BISR 测试。注：睡眠唤醒流程根据该域指定测试类型，默认是 BIST 测试。
REBIST	[1]	RW,0x0	重新 BIST 测试指示，在配置 3 状态判断该标志，如果该位有效，则重新转入 BIST 测试，同时对该位进行自清
—	其它	—	保留。

#### 4.2.7 BIST\_OVERTIME\_TH

存储器测试超时阈值寄存器 BIST\_OVERTIME\_TH，维护可读写，软件只读。该寄存器用于存储器自测试时的超时阈值设置。

表 4-9: BIST\_OVERTIME\_TH 寄存器域的描述

名称	范围	类型	描述
BIST_OVERTIME_EN	[32]	RW,0x1	存储器自测试超时机制使能。
BIST_OVERTIME_TH	[31:0]	RW,0xffff,ffff	存储器自测试时的超时阈值。
—	其它	—	保留。

#### 4.2.8 MT\_STATE

MT\_STATE 是状态寄存器，维护和软件只读，复位为“0x0”；

表 4-10: MT\_STATE 寄存器域的描述

名称	范围	类型	描述
OBS	[37:33]	RO,0x0	观测输出（仅选择稳定输出信号）。

			<p>[37]: 核心 PLL 时钟锁定。</p> <p>[36]: 互连 PLL 时钟锁定。</p> <p>[35]: 存控 PLL 时钟锁定。</p> <p>[34]: PCIe-0 接收方链路连接建立。</p> <p>[33]: PCIe-1 接收方链路连接建立。</p>
FLAG	[32:28]	RO,0x0	<p>[32]: 保留;</p> <p>[31]: 对应 IOR: FlagReg[3], 该 IOR 可被核心 3 读写, 指示 3 核心状态, <b>高电平有效</b>; [30]: 对应 IOR: FlagReg[2], 该 IOR 可被核心 2 读写, 指示 2 核心状态, <b>高电平有效</b>; [29]: 对应 IOR: FlagReg[1], 该 IOR 可被核心 1 读写, 指示 1 核心状态, <b>高电平有效</b>; [28]: 对应 IOR: FlagReg[0], 该 IOR 可被核心 0 读写, 指示 0 核心状态, <b>高电平有效</b>。</p>
BIST_INF	[27:24]	RO,0x0	<p>[27]: BIST 测试完成, <b>高电平有效</b>。</p> <p>[26:25]: 测试结果。</p> <p>2'b00, 表示测试无错;</p> <p>2'b01, 表示有错可修复;</p> <p>2'b10, 表示有错不可修复;</p> <p>2'b11, 表示测试时间超时。 [24]: 指示在存储器 Debug 测试方式下, 发现错误而暂停存储器 Debug 测试, 此时可通过状态扫描获得具体的错误信息, <b>高电平有效</b>。</p>
SYS_ERR	[22]	RO,0x0	系统错标志, <b>高电平有效</b> 。
INT_DONE	[21]	RO,0x0	中断完成标志, <b>高电平有效</b> 。
MT_ERR	[20]	RO,0x0	自定义维护接口错, <b>高电平有效</b> 。
SFSM3_STATE	[19:17]	RO,0x0	核心 3 的子状态机状态 (同核心 0)。
SFSM2_STATE	[16:14]	RO,0x0	核心 2 的子状态机状态 (同核心 0)。
SFSM1_STATE	[13:11]	RO,0x0	核心 1 的子状态机状态 (同核心 0)。
SFSM0_STATE	[10:8]	RO,0x0	<p>核心 0 的子状态机状态。</p> <p>0: 空闲状态;</p> <p>1: 运行状态;</p> <p>2: 降频状态;</p>



			3: 睡眠状态; 4: 升频状态; 5: 存储器自测试状态; 其他保留。
MFSM_STATE	[7:4]	RO,0x0	主状态机状态。 0: 上电复位状态; 1: 冷复位结束状态; 2: 配置 1 状态; 3: PLL 升频状态; 4: 时钟切换 1 状态; 5: 结束复位状态; 6: 配置 2 状态; 7: BIST 测试状态; 8: 配置 3 状态; 9: 初始化加载状态; 10: 预备运行状态; 11: 运行状态; 12/13: 进入复位状态; 14/15: 时钟切换 2 状态。
RTPU_STATE	[3:0]	RO,0x0	串行接口状态机状态。 0: 空闲状态; 1: 请求接收状态; 2: 等待维护响应状态; 4: 发送维护响应状态; 8: 处理维护读命令状态; 9: 处理维护写命令状态; 10: 处理维护 SCAN 读命令状态; 11: 处理维护 SCAN 写命令状态; 12: 处理维护 SROM 加载命令状态; 其他保留。
—	其它	—	保留。

## 4.2.9 CORE\_ONLINE

CORE\_ONLINE 为核心在位使能寄存器，维护可读写，软件只读。该寄存器用于控制核心的在位使能，复位时初值为“0xf”。

表 4-11: CORE\_ONLINE 寄存器域的描述

名称	范围	类型	描述
CORE_ONLINE	[3:0]	RW,0xf	CORE_ONLINE[i]为“1”，表示该核心在位，该位只控制核心。
—	其它	—	保留。

## 4.2.10 MT\_INT

MT\_INT 是维护中断寄存器，维护只写。通过维护请求写该寄存器时，将对目标核心产生一次维护中断。该寄存器硬件实现时不真实存在，直接通过请求包向 IPU 发送（唤醒中断直接在 MCU 根据核心状态处理）。

表 4-12: MT\_INT 寄存器域的描述

名称	范围	类型	描述
MT_INT_Type	[9:8]	WO,0	指示中断类型，0x: 正常维护中断；10: 维护睡眠中断；11: 维护唤醒中断。
MT_INT_Tar	[1:0]	WO,0	指示中断目标编码指示。
—	其它	—	保留。

## 4.2.11 MT\_INT\_END

MT\_INT\_END 是维护中断结束寄存器，维护和软件可读写，用于标记核心已经处理完维护中断。复位为 0x1；

收到维护中断（包括睡眠中断、唤醒中断和一般维护中断）都清除标志位。对于一般的维护中断由中断处理程序重新置该位；对于睡眠中断，维护收到对应核心写 SleepDone 的请求时重新置该位；

对于唤醒中断，唤醒请求能够被处理或者被丢弃（已经唤醒或正在唤醒）或者被后面的睡眠请求覆盖，重新置该位；

该标志作为中断完成的结果输出给系统；

表 4-13: MT\_INT\_END 寄存器域的描述

名称	范围	类型	描述
MT_INT_END	[0]	RW,0x1	“1”时表明已经处理完一次维护中断。
—	其它	—	保留。

## 4.2.12 CPUID

CPUID 是 CPU 号寄存器，维护和软件都可读写，用于设置 CPU 编号信息。

表 4-14: CPUID 寄存器域的描述

名称	范围	类型	描述
CPUID	[55:0]	RW,0	CPU 编号。
—	其它	—	保留。

## 4.2.13 SOFT\_INF0~15

SOFT\_INF0~15 是软件信息寄存器（共 16 个寄存器，每个核心对应一个寄存器），维护和软件 都可读写，软件可使用该寄存器与维护接口交换信息。其中[63]为软件错误标志，写该位为 1 时，维护接口将置 IOR: GG\_FAULT\_STAT[SOFTERR]位为 1。

表 4-15: SOFT\_INF0~15 寄存器域的描述

名称	范围	类型	描述
ERROR	[63]	RW,0	软件错误标志。
INF	[62:0]	RW,0	软件信息位。

## 4.2.14 LONG\_TIME

LONG\_TIME 为长时钟寄存器，维护和软件都可读写，复位时初值为“0”。该寄存器用于长时钟计数。软件可以通过 IO 写设置该寄存器的初值，计数时钟为维护时钟。

表 4-16: LONG\_TIME 寄存器域的描述

名称	范围	类型	描述
LONG_TIME	[63:0]	RW,0	长时钟计数器值。

## 4.2.15 LONG\_PRESET

LONG\_PRESET 为长时钟预置寄存器，维护和软件都可读写，上电或冷复位预置为

“0x61A8”，即对 25MHz 的维护时钟计数 1ms 的次数。每个维护时钟减“1”，减至“0”后循环至预置值。减至“0”时，将长时钟寄存器 LONG\_TIME 加“1”。表 4-17: LONG\_PRESET 寄存器域的描述

名称	范围	类型	描述
LONG_PRESET	[31:0]	RW,0x61A8	长时钟预置值。
—	其它	—	保留。

## 4.2.16 TESTSEL

控制 TESTOUT 选择输出的选择端；维护和软件都可读，维护只写。

表 4-18: TESTSEL 寄存器域的描述

名称	范围	类型	描述
TESTSEL	[5:0]	RW,0x0	TESTOUT 输出选择信号。
—	其它	—	保留。

## 4.2.17 IO\_START

IO\_START 是外围接口使能寄存器，维护和软件都可读写。该寄存器用于故障隔离和容错，IO\_START[1:0]分别对应 PCI-E1 和 PCI-E0。此二接口根据对应 IO\_START 来完成对接收到的请求返回响应，如果对应位为“1”则按照正常的流程处理；否则直接返回读/写错误响应。复位时初值为“0x3”。

表 4-19: IO\_START 寄存器域的描述

名称	范围	类型	描述
IO_START	[1:0]	RW,0x3	外围接口使能。
—	其它	—	保留。

## 4.2.18 PERST\_N\_PCIE0~1

PERST\_N\_PCIE0~1 用于对 PCI-E 接口（PCI-E0/PCI-E1）进行复位，可以实现由软件发起 PCIE 接口部分非粘连逻辑复位。维护和软件可读写，软件自己控制复位信号的宽度，写“0”表示复位，写“1”表示复位结束。

表 4-20: PERST\_N\_PCIE0~1 寄存器域的描述

名称	范围	类型	描述
----	----	----	----

PERST_N	[0]	RW,0x1	PCI-E 接口 PERST 复位。
—	其它	—	保留。

## 4.2.19 BUTTON\_RST\_N\_PCIE0~1

BUTTON\_RST\_N\_PCIE0~1 是对 PCI-E 接口 (PCI-E0/PCI-E1) 的所有逻辑进行复位。维护和软件可读写，软件自己控制复位信号的宽度，写“0”表示复位，写“1”表示复位结束。

表 4-21: BUTTON\_RST\_N\_PCIE0~1 寄存器域的描述

名称	范围	类型	描述
BUTTON_RST_N	[0]	RW,0x1	PCI-E 接口 BUTTON_RST 复位。
—	其它	—	保留。

## 4.2.20 MTRSPMISS

MTRSPMISS[15:0]用于对维护接口接收到的响应与目前的请求 SMAF 编号是否相等进行记录，维护和软件只读；

表 4-22: MTRSPMISS 寄存器域的描述

名称	范围	类型	描述
MTRSPMISS	[15:0]	RO,0x0	维护接口接收到的响应与目前的请求编号不匹配则该计数器加 1。
—	其它	—	保留。

## 4.2.21 SLEEP\_DONE0~3

写 SLEEP\_DONE0~3 则指示了睡眠核心已经可以睡眠，此时 MCU 将启动对应核心的睡眠流程。SLEEP\_DONE<sub>x</sub> 只能软件可写（睡眠中断处理程序），维护不可写，并且该寄存器硬件实现时不真实存在，如果来自 IPU 的写操作则需要判断写对应寄存器的请求源核心号是否与寄存器的编号[0~3]对应，如果不对应则返回错误响应。

表 4-23: SLEEP\_DONE0~3 寄存器域的描述

名称	范围	类型	描述
SLEEP_DONE	[0]	WO,0x0	写该寄存器则 MCU 对对应核心进入睡眠流
—	其它	—	保留。

## 4.2.22 PLL\_CHG\_CNT

时钟切换时间间隔寄存器 PLL\_CHG\_CNT，维护可读写，软件只读。该寄存器用于在复位结束和复位开始时，根据该值决定不同核心、存控、和系统接口的时钟在低频和高频间切换的时间间隔。

表 4-24: PLL\_CHG\_CNT 寄存器域的描述

名称	范围	类型	描述
PLL_CHG_CNT	[31:0]	RW,65536	核心、存控、和系统接口的时钟在低频和高频间切换的时间间隔。
—	其它	—	保留。

## 4.2.23 CG\_BIST\_STAT

CG\_BIST\_STAT 为 BIST 状态寄存器，维护和软件只读，该寄存器用于存放 CG 内各核心内阵列的 BIST 结果 (Fail/Repair)，以及 CTAG 的 BIST 结果。复位时初值为全“0”。

该寄存器 MCU 进行隐式写，其中各核心的 BIST 结果在 MCU 和 MTBOX 的接口上采用串行移位的方式；

表 4-25: CG\_BIST\_STAT 寄存器域的描述

名称	范围	类型	描述
CPMbistinfo	[51:48]	RO,0	CPM BIST 信息，从高到低对应 TCache，TTag（每个阵列 2 位，高位表示 Fail，低位表示 Repair）。 [Fail、Repair]具体如下： [1:0]=2'b00，表示测试无错； [1:0]=2'b01，表示有错可修复； [1:0]=2'b10，表示有错不可修复； [1:0]=2'b11，表示测试时间超时。
Core3bistinfo	[47:36]	RO,0	Core3 BIST 信息（同 Core0）。
Core2bistinfo	[35:24]	RO,0	Core2 BIST 信息（同 Core0）。
Core1bistinfo	[23:12]	RO,0	Core1 BIST 信息（同 Core0）。

Core0bistinfo	[11:0]	RO,0	Core0 BIST 信息，从高到低对应 Scache，Stag，Dcache，Dtag，Icache，Itag（每个阵列 2 位，高位表示 Fail，低位表示 Repair）。 [Fail、Repair]具体如下：
---------------	--------	------	-----------------------------------------------------------------------------------------------------------------



			[1:0]=2'b00, 表示测试无错; [1:0]=2'b01, 表示有错可修复; [1:0]=2'b10, 表示有错不可修 复;
—	其它	—	保留。

## 4.2.24 BIST\_STAT

BIST\_STAT 为 BIST 状态寄存器，维护和软件只读，该寄存器用于存放各核心和 CTAG 的 BIST

结果。复位时初值为全“0”。

表 4-26: BIST\_STAT 寄存器域的描述

名称	范围	类型	描述
PC1_BIST_Info	[56:54]	RO,0	指示 Page Cache1 的 BIST 结果（同
PC0_BIST_Info	[53:51]	RO,0	指示 Page Cache0 的 BIST 结果（同
CPM_BIST_Info	[50:48]	RO,0	指示 CPM 的 BIST 结果（同 Core0）；
Core3_BIST_Info	[11:9]	RO,0	指示核心 3 的 BIST 结果（同 Core3）；
Core2_BIST_Info	[8:6]	RO,0	指示核心 2 的 BIST 结果（同 Core2）；
Core1_BIST_Info	[5:3]	RO,0	指示核心 1 的 BIST 结果（同 Core1）；
Core0_BIST_Info	[2:0]	RO,0	指示核心 0 的 BIST 结果；该结果对应核心内 所有阵列或指定阵列测试： [2]: Done; [1]: Fail; [0]: Repair。 [Fail、Repair]具体如下： [1:0]=2'b00, 表示测试无错； [1:0]=2'b01, 表示有错可修复； [1:0]=2'b10, 表示有错不可修 复。
—	其它	—	保留。

## 4.2.25 TAPSEL

TAPSEL 为 TAP 控制器选择寄存器，维护和软件只读。用于记录芯片引脚的值。

表 4-27: TAPSEL 寄存器域的描述

名称	范围	类型	描述
TAPSEL	[1:0]	RO	用于控制 TAP 控制器的输出： 2'b00：表示选择 PCI-E 0 的 TAP 控制器； 2'b01：表示选择 PCI-E 1 的 TAP 控制器。 2'b1x：表示选择 Tbox 的 TAP 控制器；
—	其它	—	保留。

#### 4.2.26 ST\_NEXT\_FLAG

状态机控制寄存器 ST\_NEXT\_FLAG。该寄存器不真实存在，维护只写，不关心写的内容。写该寄存器作为状态切换的条件，具体包括：配置状态 1 向等待 PLL 稳定状态切换、配置状态 2 向 SRAM 状态切换、SCAN 配置状态向存储器自测试状态切换。在写该寄存器之前，维护已经完成对其它寄存器的配置。

表 4-28： ST\_NEXT\_FLAG 寄存器域的描述

名称	范围	类型	描述
ST_NEXT_FLAG	[0]	WO,0x0	写该寄存器作为主状态机从配置状态 2 向 SRAM 状态切换的条件。
—	其它	—	保留。

#### 4.2.27 SERR\_CNTTH

MCU 单错预警阈值，维护和主核可读写，复位时初值是 0xFFFF\_FFFF；

表 4-29： SERR\_CNTTH 寄存器域的描述

名称	范围	类型	描述
SERR_CNTTH	[31:0]	RW, 0xFFFFFFFF F	MCU 单错预警阈值。如果单错统计计数器的计数值等于该阈值且对应的单错报错使能位打开，则在 SI_FAULT_STAT 记录单错预警。 单错统计计数器共有 3 个： IPUSERR_CNT， PIU0SERR_CNT， PIU1SERR_CNT。
—	其它	—	保留。

## 4.2.28 IPUSERR\_CNT

该寄存器统计：MCU 接收到 IPU 方向请求和响应发生 ECC 单错的次数。维护和主核可读写。该寄存器写清零。

表 4-30：IPUSERR\_CNT 寄存器域的描述

名称	范围	类型	描述
IPUSERR_CNT	[31:0]	RW,0	来自 IPU 的请求或响应报发生单错，则计数器加 1。
—	其它	—	保留。

## 4.2.29 PIUXSERR\_CNT

该寄存器统计：MCU 接收到 PIU 方向响应包发生 ECC 单错的次数。维护和主核可读写。该寄存器写清零。

表 4-31：PIUXSERR\_CNT 寄存器域的描述

名称	范围	类型	描述
PIUXSERR_CNT	[31:0]	RW,0	来自 PIUx 的响应报发生单错，则计数器加 1。
—	其它	—	保留。

## 4.2.30 ERRRPT\_EN

该寄存器控制 MCU 各种报错使能。维护和主核可读写，复位初始为全“0”。具体定义如下

表：表 4-32：ERRRPT\_EN 寄存器域的描述

名称	范围	类型	描述
TBX_TI_TransErr	[19]	RW,0	维护接口错使能（JTAG）。
MCU_SI_TransErr	[18]	RW,0	维护接口错使能（自定义串口）。
FromIPUResMErrEn	[17]	RW,0	IPU 发向 MCU 的响应包的包头多错报错使能。
FromIPUResCtrlErrEn	[16]	RW,0	IPU 发向 MCU 的响应包控制错使能。
FromIPUReqMErrEn	[15]	RW,0	IPU 发向 MCU 的请求包多错报错使能。
FromIPUReqCtrlErrEn	[14:13]	RW,0	IPU 发向 MCU 的请求的包头和包尾信息不一致使能。
ToIPUReqCtrlErr	[12]	RW,0	维护发向 IPU 的请求的控制错使能。
ToMCUCtrlErrEn	[11]	RW,0	维护发向 MCU 的请求的控制错使能。
ToPIU1CtrlErrEn	[10:9]	RW,0	维护发向 PIU1 的请求的控制错使能。

ToPIU0CtrlErrEn	[8:7]	RW,0	维护发向 PIU0 的请求的控制错使能。
CntPIU1MErrEn	[6]	RW,0	MCU 收到 PIU1 ECC 多错报错使能。
CntPIU0MErrEn	[5]	RW,0	MCU 收到 PIU0 ECC 多错报错使能。
CntIPUMErrEn	[4]	RW,0	MCU 收到 IPU ECC 多错报错使能。
CntPIU1SErrEn	[3]	RW,0	MCU 收到 PIU1 ECC 单错报错使能。
CntPIU0SErrEn	[2]	RW,0	MCU 收到 PIU0 ECC 单错报错使能。
CntIPUSErrEn	[1]	RW,0	MCU 收到 IPU ECC 单错报错使能。
CtrlErrEn	[0]	RW,0	MCU 内部控制错报错使能。
—	其它	—	保留。

### 4.2.31 ERR\_INF

该寄存器记录 MCU 各种报错。维护和主核可读写，写清零。具体定义如下表：

表 4-33: ERR\_INF 寄存器域的描述

名称	范围	类型	描述
CtrlErrCode	[35:32]	RW,0	表示出现控制错时，各源的有效位情况； [3]: 表示 PIU1； [2]: 表示 IPU； [1]: 表示 MCU； [0]: 表示 PIU0。
TBX_TL_TransErr	[19]	RW,0	维护接口错（JTAG）。
MCU_SI_TransErr	[18]	RW,0	维护接口错（自定义串口）。
FromIPUResMErr	[17]	RW,0	IPU 发向 MCU 的响应包的包头有多错；
FromIPUResCtrlErr	[16]	RW,0	多包的 OMCB 响应，在传输发数据包时维护接口被复位，此时 MCU 要把数据收齐，该报文被丢弃。
FromIPUReqMErr	[15]	RW,0	IPU 发向 MCU 的请求包有多错。
FromIPUReqCtrlErr	[14:13]	RW,0	IPU 发向 MCU 的请求的地址[39:36]与 MCU 的编址不匹配。 IPU 发向 MCU 的请求的包头和包尾信息不一致（必须是单包）。
ToIPUReqCtrlErr	[12]	RW,0	多包的 IMRB 请求，在传输发数据包时维护接口被复位，此时 MCU 要把数据传完，但不保证数据的正确性。

ToMCUCtrlErr	[11]	RW,0	维护发向 MCU 的请求的控制错，即包头和包尾信息不一致（必须是单包）。
ToPIU1CtrlErr	[10:9]	RW,0	维护发向 PIU1 的请求的控制错，即上一个请求没有串行输出完成又发出一个请求，该错误也

			可能是维护串行接口复位造成。 维护发向 PIU1 的请求的控制错，即包头和包尾信息不一致（必须是单包）。
ToPIU0CtrlErr	[8:7]	RW,0	维护发向 PIU0 的请求的控制错，即上一个请求没有串行输出完成又发出一个请求，该错误也可能是维护串行接口复位造成。 维护发向 PIU0 的请求的控制错，即包头和包尾信息不一致（必须是单包）。
CntPIU1MErr	[6]	RW,0	MCU 收到 PIU1 的响应有 ECC 多错报错。
CntPIU0MErr	[5]	RW,0	MCU 收到 PIU0 的响应有 ECC 多错报错。
CntIPUMErr	[4]	RW,0	MCU 收到 IPU 的响应数据包有 ECC 多错报错。
CntPIU1SErr	[3]	RW,0	MCU 收到 PIU1 的响应有 ECC 单错预警报错。
CntPIU0SErr	[2]	RW,0	MCU 收到 PIU0 的响应有 ECC 单错预警报错。
CntIPUSErr	[1]	RW,0	MCU 收到 IPU 的请求或响应（包括数据包）有 ECC 单错预警报错。
CtrlErr	[0]	RW,0	MCU 内部控制错报错，同时收到多个响应。 即同时收到来自 MCU、PIU1、PIU0 和 IPU 的响应。
—	其它	—	保留。

### 4.2.32 SCAN\_WAIT\_CFG

SCAN\_WAIT\_CFG 是 SCAN 间隔控制寄存器，维护可读写，软件只读，用于控制两个 SCAN

操作（对字节而言，最小间隔是 5 拍处理一次）之间的间隔，复位为“0x0”。

表 4-35: SCAN\_WAIT\_CFG 寄存器域的描述

名称	范围	类型	描述
----	----	----	----

SCAN_WATL_CFG	[2:0]	RW,0x0	3'h0: 0 拍; 即 5 拍处理 1 个; 3'h1: 1 拍; 即 6 拍处理 1 个; 3'h2: 2 拍; 即 7 拍处理 1 个; 3'h3: 3 拍; 即 8 拍处理 1 个; 3'h4: 4 拍; 即 9 拍处理 1 个; 3'h5: 5 拍; 即 10 拍处理 1 个; 3'h6: 6 拍; 即 11 拍处理 1 个; 3'h7: 7 拍; 即 12 拍处理 1 个。
---------------	-------	--------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------



—	其它	—	保留。
---	----	---	-----

### 4.2.33 CLU\_TEST

CLU\_TEST 是 CLU 测试控制寄存器，维护和软件都可读写，复位为“0x0”。

表 4-36: CLU\_TEST 寄存器域的描述

名称	范围	类型	描述
PLL_TEST_CORE	[2]	RW,0x0	核心 PLL 测试模式指示，高有效。
PLL_TEST_MM	[1]	RW,0x0	存控 PLL 测试模式指示，高有效。
PLL_TEST_XBX	[0]	RW,0x0	互连 PLL 测试模式指示，高有效。
—	其它	—	保留。

### 4.2.34 PC\_BIST\_STAT

PC\_BIST\_STAT 为 BIST 状态寄存器，维护和软件只读，该寄存器用于存放 Page Cache 阵列的

BIST 结果 (Fail/Repair)。复位时初值为全“0”。

该寄存器 MCU 进行隐式写，其中 Page Cache 的 BIST 结果在 MCU 和 MTBOX 的接口上采用串行移位的方式；

表 4-25: PC\_BIST\_STAT 寄存器域的描述

名称	范围	类型	描述
PC1bistinfo	[11:6]	RO,0	Page Cache 1 BIST 信息 (同 Page Cache
PC0bistinfo	[5:0]	RO,0	Page Cache 0 BIST 信息，从高到低对应 PCDATA 每个阵列 2 位，高位表示 Fail，低位表示 Repair)。 [5]: BISTDONE 信号 [Fail、Repair]具体如下： [4:3]=2'b00，表示测试无错； [4:3]=2'b01，表示有错可修复； [4:3]=2'b10，表示有错不可修复； [4:3]=2'b11，表示测试时间超时。
—	其它	—	保留。

### 4.2.35 Corex\_ERRRST\_N

核心错误信息清除寄存器，用于清除核心记录的错误信息，维护、软件均可读写,复位时初值为

0x1。

表 4-37: Corex\_ERRRST\_N 寄存器域的描述

名称	范围	类型	描述
ERRRST_N	[0]	RW,0x1	为“0”表示要清除核心记录的错误信息。
—	其它	—	保留。

### 4.2.36 SI\_FAULT\_STAT

SI\_FAULT\_STAT 为系统接口总错状态寄存器，维护和软件可读写。该寄存器用于系统接口总错信息。复位时初值为全“0”。采上升沿登记；按位写“1”清除。

表 4-38: SI\_FAULT\_STAT 寄存器域的描述

名称	范围	类型	描述
PIU1_LinkRstFault	[22]	RW1C,0	指示 PCIe 链路因软件配置或链路断掉而发生了复位故障，对应 PIU1。
PIU0_LinkRstFault	[21]	RW1C,0	指示 PCIe 链路因软件配置或链路断掉而发生了复位故障，对应 PIU0。
IPU_FaultIntLost	[20]	RW1C,0	表示维护故障中断丢失。
IPU_NMIIntLost	[19]	RW1C,0	表示不可屏蔽故障中断丢失。
IPUERR	[15:12]	RW1C,0	由高到低分别表示 IPU 产生的已纠正错、不可纠正错、控制错和软件错。
PIU1ERR	[11:8]	RW1C,0	由高到低分别表示 PIU1 产生的已纠正错、不可纠正错、控制错和软件错。
PIU0ERR	[7:4]	RW1C,0	由高到低分别表示 PIU0 产生的已纠正错、不可纠正错、控制错和软件错。
MCUERR	[3:0]	RW1C,0	由高到低分别表示 MCU 产生的已纠正错、不可纠正错、控制错和软件错。
—	其它	—	保留。

### 4.2.37 SI\_FAULT\_EN

SI\_FAULT\_EN为系统接口总错使能寄存器，维护和软件都可读写。该寄存器用于控制系统接口总错信息。复位时初值为全“0”。

表 4-39: SI\_FAULT\_EN 寄存器域的描述

名称	范围	类型	描述
PIU1_LinkRstFault_EN	[22]	RW,0	指示 PCIe 链路因软件配置或链路断掉而发生了复位故障，该位表示该故障的报错使能。对应
PIU0_LinkRstFault_EN	[21]	RW,0	指示 PCIe 链路因软件配置或链路断掉而发生了复位故障，该位表示该故障的报错使能。对应 PIU0。
IPU_FaultIntLost_EN	[20]	RW,0	表示故障中断丢失报错使能。
IPU_NMIIntLost_EN	[19]	RW,0	表示不可屏蔽故障故障中断丢失报错使能。
IPUERR_EN	[15:12]	RW,0	由高到低分别表示 IPU 产生的已纠正错、不可纠正错、控制错和软件错报错使能。
PIU1ERR_EN	[11:8]	RW,0	由高到低分别表示 PIU1 产生的已纠正错、不可纠正错、控制错和软件错报错使能。
PIU0ERR_EN	[7:4]	RW,0	由高到低分别表示 PIU0 产生的已纠正错、不可纠正错、控制错和软件错报错使能。
MCUERR_EN	[3:0]	RW,0	由高到低分别表示 MCU 产生的已纠正错、不可纠正错、控制错和软件错报错使能。
—	其它	—	保留。

### 4.2.38 SI\_FAULT\_INT\_EN

SI\_FAULT\_INT\_EN 为系统接口故障中断使能寄存器，维护和软件都可读写。该寄存器用于控制系统接口产生故障中断。复位时初值为全“0”。

表 4-40: SI\_FAULT\_INT\_EN 寄存器域的描述

名称	范围	类型	描述
PIU1_LinkRstFault_INT_EN	[22]	RW,0	指示 PCIe 链路因软件配置或链路断掉而发生了复位故障，该位表示该故障的报中断使能。对应
PIU0_LinkRstFault_INT_EN	[21]	RW,0	指示 PCIe 链路因软件配置或链路断掉而发生了复位故障，该位表示该故障的报中断使能。对应

IPUERR_INT_EN	[15:12]	RW,0	由高到低分别表示 IPU 产生的已纠正错、不可纠正错、控制错和软件错故障中断使能。
PIU1ERR_INT_EN	[11:8]	RW,0	由高到低分别表示 PIU1 产生的已纠正错、不可纠正错、控制错和软件错故障中断使能。
PIU0ERR_INT_EN	[7:4]	RW,0	由高到低分别表示 PIU0 产生的已纠正错、不可纠正错、控制错和软件错故障中断使能。
MCUERR_INT_EN	[3:0]	RW,0	由高到低分别表示 MCU 产生的已纠正错、不可纠正错、控制错和软件错故障中断使能。
—	其它	—	保留。

### 4.2.39 CLK\_SEL\_PCIE0~1

PCI-E 时钟选择寄存器，维护和软件可读写；在上电复位/冷复位有效（MCU 给 PIU 的上电复位/冷复位）或 BUTTON\_RST\_N 有效时该寄存器处于复位状态(零)，即此时不可配置；

注：PERST\_N 不影响该 IOR，且 BUTTON\_RST\_N 和 PERST\_N 的复位值都是无效（高）；

表 4-41: CLK\_SEL\_PCIE0~1 寄存器域的描述

名称	范围	类型	描述
CLK_SEL_PCIE	[0]	RW,0x1	该位为“1”PIU 选择 PCLK，否则选择 MCLK。
—	其它	—	保留。

### 4.2.40 FIFO\_SYNSEL

FIFO\_SYNSEL 是 SBOX、CPM 和 IPU 模块中异步 FIFO 读/写地址同步的级数配置寄存器，维护和软件可读写。

表 4-42: FIFO\_SYNSEL 寄存器域的描述

名称	范围	类型	描述
SYNSEL1	[7:4]	RW,0x1	[7:6]: 保留； [5:4]: CPM 或 IPU 内异步 FIFO 读/写地址同步的级数： 2'b00: 2 级； 2'b01: 3 级； 2'b10: 4 级； 2'b11: 5 级。 缺省为 2'b01。

SYNSELO	[3:0]	RW,0x2	[3:2]: 保留; [1:0]: SBox 内异步 FIFO 读/写地址同步的级数: 2'b00: 2级; 2'b01: 3级;
			2'b10: 4级; 2'b11: 5级。 缺省为
—	其它	—	保留。

#### 4.2.41 CPU\_INFO

CPU\_INFO 寄存器用于记录申威 411 处理器的产品信息。维护可读写，软件只读。

表 4-43: CPU\_INFO 寄存器域的描述

名称	范围	类型	描述
MANUFACTURE_ID	[63:0]	RO,0x0	该字段用于记录生产批次号等信息。

#### 4.2.42 FlagReg

维护和软件可读写。可由任意核心进行软件读写。软件写和维护写直接按位写（4 位一起）。维护读和软件读按位返回（4 位一起）。

表 4-44: FlagReg 寄存器域的描述

名称	范围	类型	描述
Flag3	[3]	RW,0x0	
Flag2	[2]	RW,0x0	
Flag1	[1]	RW,0x0	
Flag0	[0]	RW,0x0	
—	其它	—	保留。

#### 4.2.43 OP\_MASK

维护操作屏蔽控制寄存器寄存器 OP\_MASK。该寄存器维护可读写，软件只读。

表 4-45: OP\_MASK 寄存器域的描述

名称	范围	类型	描述
MemWrMask	[3]	WO,0x0	维护写存储器被屏蔽。

MemRdMask	[2]	WO,0x0	维护读存储器被屏蔽。
IOWrMask	[1]	WO,0x0	维护写 IO 寄存器被屏蔽。
IORdMask	[0]	WO,0x0	维护读 IO 寄存器被屏蔽。
—	其它	—	保留。

#### 4.2.44 BIST\_GOON

Debug 测试控制寄存器 BIST\_GOON。该寄存器不真实存在，维护只写，不关心写的内容。  
 Debug 测试扫出出错信息后，写该寄存器继续进行测试。

表 4-46: BIST\_GOON 寄存器域的描述

名称	范围	类型	描述
BIST_GOON	[0]	WO,0x0	Debug 测试扫出出错信息后，写该寄存器继续进行测试。
—	其它	—	保留。

#### 4.2.45 WAKEUP\_CTL

外部唤醒中断控制寄存器 WAKEUP\_CTL。该寄存器维护可读写，软件只读。

表 4-47: WAKEUP\_CTL 寄存器域的描述

名称	范围	类型	描述
WAKEUP_L	[4]	RO	用于观测芯片引脚 WAKEUP_L。
TYPE_SEL	[1]	RW,0x0	有效类型选择。 0: 表示低电平有效。 1: 表示高电平有效。
MASK_EN	[0]	RW,0x1	屏蔽使能。该位为“1”不响应中断。
—	其它	—	保留。

#### 4.2.46 NMI\_CTL

外部不可屏蔽中断控制寄存器 NMI\_CTL。该寄存器维护可读写，软件只读。

表 4-48: NMI\_CTL 寄存器域的描述

名称	范围	类型	描述
NMI_L	[4]	RO	用于观测芯片引脚 NMI_L。

NMI_CTL	[3:2]	RW,0x3	有效类型选择。
			2`b00: 表示上升沿有效; 2`b01: 表示下降沿有效; 2`b10: 表示高电平有效; 2`b11: 表示低电平有效。
MASK_EN	[0]	RW,0x1	屏蔽使能。该位为“1”不响应中断。
—	其它	—	保留。

#### 4.2.47 PIUPLL\_CNT

PIU PLL 稳定等待阈值寄存器 PIUPLL\_CNT。该寄存器维护可读写，软件只读。

表 4-50: PIUPLL\_CNT 寄存器域的描述

名称	范围	类型	描述
PLL_CHG_CNT	[31:0]	RW,655 36	PIU PLL 稳定的等待时间阈值。状态机在 PLL 升频状态等待计数器满足（计数阈值由该寄存器设置），且等待 PIU 的 Clock Stable 信号（该信号为电平信号）。
—	其它	—	保留。

## 5 PCI-E 接口 IO 寄存器

PCI-E 接口的 IO 寄存器的属性与 1.1 节的描述不同，将遵循 5-1 中的约定。

表 5-1: RC 配置寄存器各域的属性列表

属性	描述
HwInit	硬件初始化 (Hardware Initialized)。该位由硬件控制，软件只读。
RO	只读 (Read-Only)。软件只读。
RW	可读写 (Read-Write)。
RW1C	只读且写“1”清除 (Read-Only/Write-1-to-Clear)。软件对该位写“1”会清除该位，写“0”则没有任何效果。
ROS	粘性只读 (Sticky Read-Only)。软件只读，且在辅助电源的情况下即便是复位也不会清除该域。
RWS	粘性读写 (Sticky Read-Write)。软件可读写，但是在有辅助电源的情况下复位不会清除该域。
RW1CS	粘性只读且写“1”清除 (Sticky Read-Only/Write-1-to-Clear)。软件对该位写“1”会清除该位，写“0”则没有任何效果；且在有辅助电源的情况下即便是复位也不会清除该域。

### 5.1 PCI-E 接口控制部件 IO 寄存器

PCI-E 接口控制部件 (PIU) 内部的 IO 寄存器按照 128B 对界编址，每个寄存器的有效位宽为 64 位。核心对这些寄存器的访问必须是 64 位粒度的。并且，PIU 内部寄存器划分为 IOR0、IOR1 两部分，其基地址描述见 1.2.5 节。

#### 5.1.1 PIU 内部 IOR0 部分寄存器

##### 5.1.1.1 PIUConfig0

PIU 配置寄存器 0，配置 PCI-E 接口中 DMA 通路的相关功能参数，可读写。

表 5-2: PIUConfig0 寄存器域的描述

名称	范围	类型	描述
----	----	----	----



MaxPendDMARCnt	[14:10]	RWS,5'd16	<p>DMA 请求通路上悬挂的 DMAR 请求的最大数量，有效取值为 0~16。</p> <p>使用建议：根据 PCI-E 接口的性能模拟结果，建议软件对该域做如下配置——如果应用中的 DMAR 请求大多是 256B 粒度的数据包，则将该域配置为 8；否则 如果请求大多是 512B 粒度的数据包，则将该域配置</p>
IOMMU_Mode	[9:8]	RWS,0	<p>DMA 请求地址代换模式选择：</p> <p>00：普通地址代换；</p> <p>10：段式 IOMMU 地址代换；</p> <p>11：页式 IOMMU 地址代换； 其他：保留</p>
DMAWNBConfig	[7:6]	RWS,10	<p>指示提交给系统接口的 DMAW 请求数据包中的 NB 位（请求包中 NB 位为“1”表示需要等前面的 DMA 请求全部收到响应后才可以提交当前请求）</p> <p>如何设置：</p> <p>0b00：NB 位固定为“1”；</p> <p>0b01：NB 位固定为“0”；</p> <p>0b1x：PIU 在接收到 DMAW 请求时，会按照一定的对界规则将其拆分成多个子包；该配置时，拆分出的第一个子包的 NB 位为“1”，其余子包的 NB 位为“0”；</p>
DMARNBConfig	[5:4]	RWS,01	<p>指示提交给系统接口的 DMAR 请求数据包中的 NB 位 如何设置：</p> <p>0b00：NB 位固定为“1”；</p> <p>0b01：NB 位固定为“0”；</p> <p>0b10：PIU 在接收到 DMAR 请求时，会按照一定的对界原则将其拆分成多个子包；该配置时，拆分出的第一个子包的 NB 位为“1”，其余子包的 NB 位为“0”；</p> <p>0b11：如果有连续的 DMAR 请求，则第一个 DMAR 请求拆分出的第一个子包的 NB 位为“1”，其余子包</p>

CCConfig	[3:2]	RWS,01	指示提交给系统接口的 DMAR/W 请求数据包中的 CC 位（决定是否做 Cache 一致性处理）如何设置： 00：则 CC 位固定给 0； 01：则 CC 位固定给 1； 1x：根据 PCI-E 接口上 TLP 报文头中的 ATTR[0]位（No Snoop）设置，即 No Snoop 位取反。
CplRelaxOrder	[1]	RWS,1	对于接收到的请求和响应数据包，该位为 0 时表示只有本地发起的 NP 类请求的响应包收到以后才可以接收 DMA 请求数据包（暂停请求包的接收），否则按顺
DMAStepByStep	[0]	RWS,0	配置 DMAR、DMAW 是否单步执行。
—	其它	—	保留。

### 5.1.1.2 EPDMABAR

EP 设备 DMA 请求基址寄存器，指示在普通地址代换模式下（PIUConfig0 中的 IOMMU\_Mode 域为 2'b00），EP 发起的 DMA 请求的地址应该在该基址以上的范围内，可读写，复位时初值为 0x4\_0000\_0000。

表 5-3：EPDMABAR 寄存器域的描述

名称	范围	类型	描述
EPDMABAR	[63:0]	RW, 0x400000000	EPDMA 请求基址，在 PIU 配置寄存器的 IOMMU_EN 关 闭时用于 DMA 请求的地址代换。代换方式是： DMA 请求地址减去该基址。  注：1) 该寄存器[63:40]位段无意义；2) 软件在配置

### 5.1.1.3 IOMMUItem0~63

IOMMU 段式代换段表寄存器 0~63，共计 64 个，在段式代换模式时（PIUConfig0 中的

IOMMU\_Mode 域为 2'b10）用作段表快表，可读写。

表 5-4：IOMMUItem0~63 寄存器域的描述

名称	范围	类型	描述
Valid	[63]	RW,0	为“1”时，指示该寄存器有效。

BaseAddr	[62:23]	RW,0	该 tag 对应 EP 设备 DMA 空间的 40 位物理基址。 注：该基址应至少保证[12:0]为 0，即 8KB 页面对
Mask	[22:16]	RW,0	该 tag 对应设备 DMA 请求允许的空间大小（以 256M 为单 位，DMA 允许访问地址范围为：
TAG	[15:0]	RW,0	索引，由{8 位总线号, 5 位设备号, 3 位功能号}组成。

#### 5.1.1.4 DTBaseAddr

页式代换设备表基址寄存器，在页式地址代换模式时（PIUConfig0 中的 IOMMU\_Mode 域为

2'b11）指定一级设备表在主存中的基址，可读写。

表 5-5: DTBaseAddr 寄存器域的描述

名称	范围	类型	描述
DTBaseAddr[39:13]	[39:13]	RW,0	一级设备表基址。注：一级设备表基址必 须按 8KB 自然对界。
—	其它	—	保留。

#### 5.1.1.5 IOMMUExcpt\_Ctrl

IOMMU 地址代换异常控制寄存器，配置 IOMMU 地址代换（段式或页式时）发生异常时的行 为，可读写。

表 5-6: IOMMUExcpt\_Ctrl 寄存器域的描述

名称	范围	类型	描述
INT_EN	[63]	RW,1	指示 IOMMU 地址代换异常是否被允许发送中断。
INT_TYPE	[12:4]	RW,0	发生 IOMMU 地址代换异常时需要发送的中断类 型：
INT_DST	[3:0]	RW,0	该中断的中断目标核心编号。
—	其它	—	保留。

#### 5.1.1.6 IOMMUExcpt\_Status

IOMMU 地址代换异常状态寄存器，指示异常的类型、设备号等信息，可读  
写。表 5-7: IOMMUExcpt\_Status 寄存器域的描述

名称	范围	类型	描述
----	----	----	----

Valid	[63]	W1C,0	为“1”时，指示该寄存器有效，即发生了 IOMMU 地址代换异常。 软件处理该类异常后，需要将该位写 1 清 0。
Retrans_EN	[62]	RW,1	软件处理 IOMMU 异常过程中，需要同时配置该位，以指示：在异常处理结束后，发生异常的请求是否需要重新代换。 该位在 Valid 位由 1 跳变为 0（即软件清 0）的同时被硬件采样，硬件根据该值决定丢弃发生异常的请求（为‘0’时）还是重新代换（为‘1’时）。
ExcptType	[61:59]	RO,0	表示当前 IOMMU 地址代换异常的类型： 0：主存中一级设备表表项无效； 1：主存中二级设备表表项无效； 2：主存中一级页表表项无效； 3：主存中二级页表表项无效； 4：页式代换越权访问； 5：访问主存中设备表/页表，返回非法响应； 6：IOMMU 段式代换缺失； 7：IOMMU 段式代换越界访问。
SegExcptIndex	[58:53]	RO,0	发生段式代换越界异常时，记录发生越界的段表索引
DeviceID	[52:37]	RO,0	记录发生异常的 DMA 请求的设备 ID（{8 位总线号, 5 位设备号, 3 位功能号}）。
DVA	[31:13]	RO,0	记录发生异常的 DMA 请求的虚页号，用于页式代换的异常处理。
—	其它	—	保留。

### 5.1.1.7 IOMMUPage\_PAddr1

IOMMU 页式地址代换地址寄存器 1，记录页式代换过程的中间结果，即二级设备表在主存中的基址，只读。

表 5-8: IOMMUPage\_PAddr1 寄存器域的描述

名称	范围	类型	描述
Addr	[39:13]	RO,0	页式地址代换过程中，访问一级设备表而得到的二级设备表基址。 该寄存器主要用于芯片调试。
—	其它	—	保留。

### 5.1.1.8 IOMMUPage\_PAddr2

IOMMU 页式地址代换地址寄存器 2，记录页式代换过程的中间结果，即一级页表在主存中的基址，只读。

表 5-9: IOMMUPage\_PAddr2 寄存器域的描述

名称	范围	类型	描述
Addr	[39:13]	RO,0	页式地址代换过程中，访问主存中的二级设备表或硬件的设备表 TLB 而得到的一级页表基址。该寄存器主要用于芯片调试。
—	其它	—	保留。

### 5.1.1.9 IOMMUPage\_PAddr3

IOMMU 页式地址代换地址寄存器 3，记录页式代换过程的中间结果，即二级页表在主存中的基址，只读。

表 5-10: IOMMUPage\_PAddr3 寄存器域的描述

名称	范围	类型	描述
Addr	[39:13]	RO,0	页式地址代换过程中，访问主存或页表 Cache 中的一级页表项而得到的二级页表基址。
—	其它	—	保留。

### 5.1.1.10 DTLB\_FlushAll

设备表 TLB 全刷新寄存器，写该寄存器即对设备表 TLB 的所有条目进行刷新，可读写，且读返回 0。

### 5.1.1.11 DTLB\_FlushDev

指定设备号设备 TLB 表刷新寄存器，写该寄存器即按照指定设备号对设备表 TLB 所有匹配条目进行刷新，可读写。

表 5-11: DTLB\_FlushDev 寄存器域的描述

名称	范围	类型	描述
DevNum	[15:0]	RW,0	指定需要刷新的设备表 TLB 表项的设备编号： [15:8] 总线号； [7:3] 设备号； [2:0] 功能号。

—	其它	—	保留。
---	----	---	-----

### 5.1.1.12 PTLB\_FlushAll

页表 TLB 全刷新寄存器，写该寄存器即对页表 TLB 的所有条目进行刷新，可读写，且读返回 0。

### 5.1.1.13 PTLB\_FlushDev

指定设备号页表 TLB 刷新寄存器，写该寄存器即按照指定设备号对页表 TLB 所有匹配的条目进行刷新，可读写。

表 5-12: PTLB\_FlushDev 寄存器域的描述

名称	范围	类型	描述
DevNum	[15:0]	RW,0	指定需要刷新的页表 TLB 表项的设备编号： [15:8] 总线号； [7:3] 设备号； [2:0] 功能号。
—	其它	—	保留。

### 5.1.1.14 PTLB\_FlushVAddr

指定设备和虚地址页表 TLB 刷新寄存器，写该寄存器即按照指定设备号、指定虚页号对页表

TLB 中匹配的条目进行刷新，可读写。

表 5-13: PTLB\_FlushVAddr 寄存器域的描述

名称	范围	类型	描述
VirAddr	[34:16]	RW,0	指定需要刷新的页表 TLB 项的虚页号（即 Addr[31:13]）；
DevNum	[15:0]	RW,0	指定需要刷新的页表 TLB 项的设备编号： [15:8] 总线号； [7:3] 设备号； [2:0] 功能号。
—	其它	—	保留。

### 5.1.1.15 PCache\_FlushAll

页表 Cache 全刷新寄存器，写该寄存器即对页表 Cache 的所有条目进行刷新，可读写，且读返回 0。

### 5.1.1.16 PCache\_FlushDev

指定设备和索引页表 Cache 刷新寄存器，写该寄存器即按照指定索引、指定设备号对页表 Cache

所有匹配的条目进行刷新，可读写。

表 5-14: PCache\_FlushDev 寄存器域的描述

名称	范围	类型	描述
Index	[21:16]	RW,0	指定需要刷新的页表 Cache 条目所处的索引位置；该域的宽度与页表 Cache 容量相关。
DevNum	[15:0]	RW,0	指定需要刷新的页表 Cache 条目的设备编号： [15:8] 总线号； [7:3] 设备号； [2:0] 功能号。
—	其它	—	保留。

### 5.1.1.17 PCache\_FlushPAddr

指定物理地址页表 Cache 刷新寄存器，写该寄存器即按照指定物理地址对页表 Cache 匹配的条目进行刷新，可读写。

表 5-15: PCache\_FlushPAddr 寄存器域的描述

名称	范围	类型	描述
PhyAddr	[39:7]	RW,0	指定需要刷新的页表 Cache 条目的物理地址高段 (Addr[39:7])；
—	其它	—	保留。

### 5.1.1.18 Timeout\_Config

超时配置寄存器，可读写。

表 5-16: Timeout\_Config 寄存器域的描述

名称	范围	类型	描述
PTCplTO	[3:0]	RW,0x7	页式 IOMMU 地址代换的主存响应超时阈值。单位是 128 个周期，并且 0x0 对应 1 个超时单位；默认值是 8 个超时单位（即 1024 个周期）。
—	其它	—	保留。

### 5.1.1.19 MSIAddr

MSI 地址寄存器，配置 MSI 中断请求的地址。

表 5-17: MSIAddr 寄存器域的描述

名称	范围	类型	描述
MSIAddr	[63:0]	RW,0	MSI 中断地址。对该地址的 DMA 写请求即为 MSI 请求。该地址不经代换（普通的、或 IOMMU_EN 打开的）直接使用。

### 5.1.1.20 MSISConfig0~255

MSI 中断配置寄存器，共计 256 个，配置相应中断的控制、状态信息，可读写。表 5-18: MSISConfig0~255 寄存器域的描述

名称	范围	类型	描述
Valid	[63]	W1C,0	为“1”时，指示该类型中断有效，即有该类型中断提交到对应核心；软件在处理完该中断后需要将该位写 1 清除。
EN	[62]	RW,0	指示该中断矢量是否被使能。
MSI_Data	[27:4]	RW,0	保存接收到的 MSI 中断的中断数据的[31:8]位段。
INTDST	[3:0]	RW,0	该中断的中断目标核心编号。
—	其它	—	保留。

### 5.1.1.21 INTxConfig

INTx 中断配置寄存器，共 INTA/B/C/D 四种类型，配置相应中断的控制、状态信息，可读写。表 5-19: INTA/B/C/Dconfig 寄存器域的描述

名称	范围	类型	描述
Valid	[63]	RO,0	指示该类型中断是否有效，硬件自动更新，软件只读。
EN	[62]	RW,0	中断提交使能位。只有该位为 1 时，才可以向核心提交该 INTx 中断。该位相当于屏蔽位取反。软件收到该类型 INTx 中断时，在进入中断处理程序后，需要先将该域写 0；等相关中断事件处理完后，再将该位写 1。
INTTYPE	[7:4]	RW,0	对于 PCI-E 接口上收到的该类 INTx 中断请求，该位配置应该将其重新映射成何种类型的中断再提交给核心： 0b1000: INTD; 0b0100: INTC; 0b0010: INTB; 0b0001: INTA; 其他：无效。



INTDST	[3:0]	RW,0	该中断的中断目标核心编号。
—	其它	—	保留。

### 5.1.1.22 aerErrIntConfig

高级错误报告错 INT 中断配置寄存器，可读写。

表 5-20: aerErrIntConfig 寄存器域的描述

名称	范围	类型	描述
Valid	[63]	W1C,0	为“1”时，指示该类型中断有效，即有该事件引起的中断提交到对应核心；软件在处理完该中断后需要将该位写 1 清 0。
EN	[62]	RW,0	指示该中断事件源是否被允许触发中断。
INTTYPE	[7:4]	RW,0	配置该事件触发的中断类型： 0b1000: INTD; 0b0100: INTC; 0b0010: INTB; 0b0001: INTA; 其他：无效。
INTDST	[3:0]	RW,0	该中断的中断目标核心编号。
—	其它	—	保留。

### 5.1.1.23 aerErrMsiConfig

高级错误报告错 MSI 中断配置寄存器，可读写。

表 5-21: aerErrMsiConfig 寄存器域的描述

名称	范围	类型	描述
Valid	[63]	W1C,0	为“1”时，指示该类型中断有效，即有该事件引起的中断提交到对应核心；软件在处理完该中断后需要将该位写 1 清 0。
EN	[62]	RW,0	指示该中断事件源是否被允许触发中断。
INTTYPE	[8:4]	RO,0	RC 指定的 MSI 中断向量号，即该事件触发的 MSI 中断向量号 该中断向量号实际是由 RC 中的“根错误状态寄存器”中的 AERIntMsgNum 域指定（见 5.2.4.1.10）。
INTDST	[3:0]	RW,0	该中断的中断目标核心编号。
—	其它	—	保留。

### 5.1.1.24 pmeIntConfig

PME (Power Management Event, 电源管理事件) INT 中断配置寄存器, 可读写。

表 5-22: pmeIntConfig 寄存器域的描述

名称	范围	类型	描述
Valid	[63]	W1C,0	为“1”时, 指示该类型中断有效, 即有该事件引起的中断提交到对应核心; 软件在处理完该中断后需要将该位写 1 清 0。
EN	[62]	RW,0	指示该中断事件源是否被允许触发中断。
INTTYPE	[7:4]	RW,0	配置该事件触发的中断类型: 0b1000: INTD; 0b0100: INTC; 0b0010: INTB; 0b0001: INTA; 其他: 无效。
INTDST	[3:0]	RW,0	该中断的中断目标核心编号。
—	其它	—	保留。

### 5.1.1.25 pmeMsiConfig

PME MSI 中断配置寄存器, 可读写。

表 5-23: pmeMsiConfig 寄存器域的描述

名称	范围	类型	描述
Valid	[63]	W1C,0	为“1”时, 指示该类型中断有效, 即有该事件引起的中断提交到对应核心; 软件在处理完该中断后需要将该位写 1 清 0。
EN	[62]	RW,0	指示该中断事件源是否被允许触发中断。
INTTYPE	[8:4]	RO,0	RC 指定的 MSI 中断向量号, 即该事件触发的 MSI 中断向量号 该中断向量号实际是由 RC 中的“PCI-E 能力列表寄存器”
INTDST	[3:0]	RW,0	该中断的中断目标核心编号。
—	其它	—	保留。

### 5.1.1.26 hpIntConfig

热插拔 INT 中断配置寄存器，可读写。

表 5-24: hpIntConfig 寄存器域的描述

名称	范围	类型	描述
Valid	[63]	W1C,0	为“1”时，指示该类型中断有效，即有该事件引起的中断提交到对应核心；软件在处理完该中断后需要将该位写 1 清 0。
EN	[62]	RW,0	指示该中断事件源是否被允许触发中断。
INTTYPE	[7:4]	RW,0	配置该事件触发的中断类型： 0b1000: INTD; 0b0100: INTC; 0b0010: INTB; 0b0001: INTA; 其他：无效。
INTDST	[3:0]	RW,0	该中断的中断目标核心编号。
—	其它	—	保留。

### 5.1.1.27 hpMsiconfig

热插拔 MSI 中断配置寄存器，可读写。

表 5-25: hpMsiConfig 寄存器域的描述

名称	范围	类型	描述
Valid	[63]	W1C,0	为“1”时，指示该类型中断有效，即有该事件引起的中断提交到对应核心；软件在处理完该中断后需要将该位写 1 清 0。
EN	[62]	RW,0	指示该中断事件源是否被允许触发中断。
INTTYPE	[8:4]	RO,0	RC 指定的 MSI 中断向量号，即该事件触发的 MSI 中断向量号 该中断向量号实际是由 RC 中的“PCI-E 能力列表寄存器”
INTDST	[3:0]	RW,0	该中断的中断目标核心编号。
—	其它	—	保留。

### 5.1.1.28 linkAutoBwIntConfig

链路自动带宽调节 INT 中断配置寄存器，可读写。

表 5-26: linkAutoBwIntConfig 寄存器域的描述

名称	范围	类型	描述
----	----	----	----

Valid	[63]	W1C,0	为“1”时，指示该类型中断有效，即有该事件引起的中断提交到对应核心；软件在处理完该中断后需要将该位写1清0。
EN	[62]	RW,0	指示该中断事件源是否被允许触发中断。
INTTYPE	[7:4]	RW,0	配置该事件触发的中断类型： 0b1000: INTD; 0b0100: INTC; 0b0010: INTB; 0b0001: INTA; 其他：无效。
INTDST	[3:0]	RW,0	该中断的中断目标核心编号。
—	其它	—	保留。

### 5.1.1.29 bwMgtIntConfig

链路带宽管理 INT 中断配置寄存器，可读写。

表 5-27: bwMgtIntConfig 寄存器域的描述

名称	范围	类型	描述
Valid	[63]	W1C,0	为“1”时，指示该类型中断有效，即有该事件引起的中断提交到对应核心；软件在处理完该中断后需要将该位写1清0。
EN	[62]	RW,0	指示该中断事件源是否被允许触发中断。
INTTYPE	[7:4]	RW,0	配置该事件触发的中断类型： 0b1000: INTD; 0b0100: INTC; 0b0010: INTB; 0b0001: INTA; 其他：无效。
INTDST	[3:0]	RW,0	该中断的中断目标核心编号。
—	其它	—	保留。

### 5.1.1.30 DMAWReqCplCnt

DMAW 请求/响应计数寄存器，可读写。

表 5-28: DMAWReqCplCnt 寄存器域的描述

名称	范围	类型	描述
DMAW_Cpl_Cnt	[63:32]	RW,0	从系统接口接收到的 DMA 写响应数目。
DMAW_Req_Cnt	[31:0]	RW,0	提交给系统接口的 DMA 写请求数目。

### 5.1.1.31 DMARReqCplCnt

DMAR 请求/响应计数寄存器，可读写。

表 5-29: DMARReqCplCnt 寄存器域的描述

名称	范围	类型	描述
DMAR_Cpl_Cnt	[63:32]	RW,0	从系统接口接收到的 DMA 读响应数目。
DMAR_Req_Cnt	[31:0]	RW,0	提交给系统接口的 DMA 读请求数目。

### 5.1.1.32 MSIRReqClrCnt

MSI 中断请求/处理计数寄存器，可读写。

表 5-30: MSIRReqClrCnt 寄存器域的描述

名称	范围	类型	描述
MSI_Process_Cnt	[63:32]	RW,0	软件处理 MSI 中断时，写 1 清中断有效位次数的计数。
MSI_Req_Cnt	[31:0]	RW,0	提交给系统接口的 MSI 中断请求数目。

### 5.1.1.33 INTXReqClrCnt

INTX 中断请求/处理计数寄存器，可读写。

表 5-31: INTXReqClrCnt 寄存器域的描述

名称	范围	类型	描述
INTX_Process_Cnt	[63:32]	RW,0	软件处理 INTX 中断时，写 1 清中断有效位次数的计
INTX_Req_Cnt	[31:0]	RW,0	提交给系统接口的 INTX 中断请求数目。

### 5.1.1.34 PageMemReqCplCnt

页式代换访存请求/响应计数器，可读写。

表 5-32: INTXReqClrCnt 寄存器域的描述

名称	范围	类型	描述
PTransMem_Cpl_Cnt	[63:32]	RW,0	页式 IOMMU 地址代换时，访问主存中设备表/页表项的响应数目。
PTransMem_Req_Cnt	[31:0]	RW,0	页式 IOMMU 地址代换时，访问主存中设备表/页表项的请求数目。

## 5.1.2 PIU 内部 IOR1 部分寄存器

### 5.1.2.1 PIUConfig1

PIU 配置寄存器 1，配置 PCI-E 接口中 IO 通路的相关功能参数，可读写。

表 5-33: PIUConfig1 寄存器域的描述

名称	范围	类型	描述
EPMemWrWait	[5]	RWS,0	对于 EP 存储器空间的写请求数据包，指示 PIU 是否需要等待收齐后再发向 RC。1'b0 表示不需要等待收齐，1'b1 表示需要等待收齐。该位的作用是：在不同的频率配置下，保证 PIU 都能正确处理。注意：1) 如果核心的工作频率低于 PIU 的工作频率，则需要提前配置该位为 1'b1；2) 而通常情况下，核心的工作频率高于 PIU 的工作频率（250MHz），此时建议保持该位为 1'b0，以免影响性能。
AllOneRspEn	[1]	RWS,0	在 NP 类配置读访问 EP 的响应为 UR/CA 类型时，该域为“1”时指示需要将响应转化为数据全 1 的正常响应后返回给核心，且不再登记 PIO 响应 UR/CA 错误；为“0”时则将读错误响应直接返回给核心，并登记 PIO 响应 UR/CA 错。
PIOStepByStep	[0]	RWS,0	配置 PIO 请求是否单步执行。
—	其它	—	保留。

### 5.1.2.2 PIOReqCplCnt

PIO 请求/响应计数器，对来自 IPU 的 PIO 请求、及返回给 IPU 的 PIO 响应数据包的数目进行计数，可读写。

表 5-34: PIOReqCpl 寄存器域的描述

名称	范围	类型	描述
PIO_Cpl_Cnt	[63:32]	RW,0	提交给系统接口的 PIO 请求（Processor IO Request）的响应计数。
PIO_Req_Cnt	[31:0]	RW,0	从系统接口接收到的的 PIO 请求计数。

### 5.1.2.3 RCReqCplCnt

RC 请求/响应计数器，对来自 IPU 的对 RC 的访问请求、及 RC 返回的响应数据包进行计数，

可 读写。

表 5-35: RCReqCptCnt 寄存器域的描述

名称	范围	类型	描述
RC_Cpl_Cnt	[63:32]	RW,0	PIU 收到的 RC 返回的响应计数。
RC_Req_Cnt	[31:0]	RW,0	PIU 发送给 RC 的请求计数。

### 5.1.2.4 EPNPReqCplCnt

对 EP 的 NP 类请求/响应计数器，对来自 IPU 的对 EP 的 NP 类访问请求及 EP 返回的对应响应数据包进行计数，可读写。

表 5-36: NPNReqCplCnt 寄存器域的描述

名称	范围	类型	描述
NP_Cpl_Cnt	[63:32]	RW,0	PIU 收到的 EP 返回的 NP 类请求所对应响应的计数。
Client0_Req_Cnt	[31:0]	RW,0	PIU 发送给 EP 的 NP 类请求计数。

### 5.1.2.5 IOR0ReqCplCnt

IOR0 请求/响应计数器，对来自 IPU 的对 IOR0 部分寄存器的访问请求及 IOR0 部分寄存器返回的响应数据包进行计数，可读写。

表 5-37: IOR0ReqCplCnt 寄存器域的描述

名称	范围	类型	描述
IOR0_Cpl_Cnt	[63:32]	RW,0	PIU 内部 IOR0 寄存器部分返回的响应计数。
IOR0_Req_Cnt	[31:0]	RW,0	PIU 内部 IOR0 部分寄存器收到的访问请求计数。

### 5.1.2.6 IOR1ReqCplCnt

IOR1 请求/响应计数器，对来自 IPU 的对 IOR1 部分寄存器的访问请求及 IOR1 部分寄存器返回的响应数据包进行计数，可读写。

表 5-38: IOR1ReqCplCnt 寄存器域的描述

名称	范围	类型	描述
IOR1_Cpl_Cnt	[63:32]	RW,0	PIU 内部 IOR1 寄存器部分返回的响应计数。
IOR1_Req_Cnt	[31:0]	RW,0	PIU 内部 IOR1 部分寄存器收到的访问请求计数。

### 5.1.2.7 NewLTSSMState0~4

NewLTSSMState0~4 寄存器记录了 PCI-E 链路的 LTSSM 的最近的 60 个状态（以移位寄存器的方式），其中NewLTSSMState0[4:0]记录的是最新的状态。

表 5-39: NewLTSSMState0~4 寄存器域的描述



名称	范围	类型	描述
LTSSMState	[59:0]	RO,0	每 5 位表示一个 LTSSM 状态，低段是较新的状态。状态编码含义见附录 B。
—	其它	—	保留。

### 5.1.2.8 OldLTSSMState0~4

OldLTSSMState0~4 寄存器记录了 PCI-E 链路初始化链路训练时 LTSSM 的 60 个状态，其中

OldLTSSMState0[4:0]记录的是前 60 个状态中最新的状态。

如果工作期间，PCI-E 链路的的状态迁移少于 60 个，则 NewLTSSMState0~4 和 OldLTSSMState0~4 记录的状态完全一样；状态达到 60 个后，OldLTSSMState0~4 即停止记录，保持最初的状态，而 NewLTSSMState0~4 则继续记录最新的状态。

表 5-40: OldLTSSMState0~4 寄存器域的描述

名称	范围	类型	描述
LTSSMState	[59:0]	RO,0	每 5 位表示一个 LTSSM 状态，低段是较新的状态。状态编码含义见附录 B。
—	其它	—	保留。

### 5.1.2.9 ErrReport

错误登记寄存器，登记 PIU 内部检测到的各种类型错误，可读写。

表 5-41: ErrReport 寄存器域的描述

名称	范围	类型	描述
PCacheTagPErr_OverLimit	[30]	RW,0	IOMMU 页式地址代换时，页表 Cache 的 TAG 阵列发生偶校验错的次数超过阈值。
PCacheDataECCSErr_OverLimit	[29]	RW,0	IOMMU 页式地址代换时，页表 Cache 的数据阵列发生 ECC 单错的次数超过阈值。
PCacheDataECCMErr	[28]	RW,0	IOMMU 页式地址代换时，页表 Cache 的数据阵列发生 ECC 多错。
DMARCplTimeout	[27]	RW,0	PCI-E 链路提示 DMAR 请求的响应发生超时。
PIOLengthErr	[26]	RW,0	收到对 RC 配置空间、EPI/O 以及配置空间的 I/O 请求长度超过 32 位。
DMAWLengthErr	[24]	RW,0	收到的 DMAW 请求长度超过 512B。



DMARHardWall	[22]	RW,0	收到对不可访问的 CG (核组) 的 DMAR 操作, 登记错误。
DMAWHardWall	[21]	RW,0	收到对不可访问的 CG (核组) 的 DMAW 操作, 登记错误。
DMAAddrErr	[20]	RW,0	非 IOMMU 地址代换时, DMA 请求地址错误——请求地址的 [36:0] 位段低于 EPIOSpace 寄存器 [36:0] 位段的值。
MSIErr	[19]	RW,0	MSI 请求错误, 即携带数据长度不是 32 位。
PIOCpl_CRS	[18]	RW,0	PIO 请求返回的响应带有 CRS 标志 (Configuration Request Retry Status)。
PIOCplTimeout	[17]	RW,0	NP 类 PIO 请求的响应超时。
PIOCpl_UR	[16]	RW,0	PIO 请求返回的响应带有 UR 标志 (Unsupported Request)。
PIOCpl_CA	[15]	RW,0	PIO 请求返回的响应带有 CA 标志 (Completion Abort)。
PIOReqECCSErr_OverLimit	[14]	RW,0	来自 IPU 的 PIO 请求数据包发生 ECC 单错的次数超过了阈值 (见 5.2.14 中 ECCSErrLimit 寄存器)。
PIOReqECCMErr	[13]	RW,0	来自 IPU 的 PIO 请求数据包发生 ECC 多错。
MCUECCSErr_OverLimit	[12]	RW,0	来自 MCU 的请求数据包发生 ECC 单错的次数超过了阈值 (见 5.2.14 中 ECCSErrLimit 寄存器)。
MCUECCMErr	[11]	RW,0	来自 MCU 的请求发生 ECC 多错。
PIOReqTypeErr	[10]	RW,0	来自 IPU 的 PIO 请求数据包类型出错。
PIOReqAddrErr	[9]	RW,0	来自 IPU 的 PIO 请求的地址不在 PCI-E 空间内。
PIOWrReqDVLDErr	[8]	RW,0	来自 IPU 的 PIO 写请求携带的数据有效位 (DVLD, Data Valid) 为全 0。
DMACplECCSErr_OverLimit	[7]	RW,0	来自 IPU 的 DMAR 响应发生 ECC 单错的次数超过了阈值 (见 5.2.14 中 ECCSErrLimit 寄存器)。
DMACplECCMErr	[6]	RW,0	来自 IPU 的 DMAR 响应发生 ECC 多错。
IPUDMAWrErrAck	[5]	RW,0	收到来自 IPU 的 DMAW 错误响应。
IPUDMARdErrAck	[4]	RW,0	收到来自 IPU 的 DMAR 错误响应。
DMARReqFifo_OverflowErr	[3]	RW,0	DMA 请求队列发生上溢。
DMAWrDataFifo_OverflowErr	[2]	RW,0	DMAW 数据缓冲发生上溢。
DMAWrDataFifo_UnferflowErr	[1]	RW,0	DMAW 数据缓冲发生下溢。

DMAWrData_Poison	[0]	RW,0	收到了带有数据中毒标志的 DMAW 请求。
—	其它	—	保留。

### 5.1.2.10 ErrEnable

错误使能寄存器，控制是否使能 PIU 内部相应的错误检测，可读写。

表 5-42: ErrEnable 寄存器域的描述

名称	范围	类型	描述
PCacheTagPErrOverLimit_En	[30]	RW,1	是否使能 PCacheTagPErr_OverLimit 错误登记。
PCacheDataECCSErrOverLimit_En	[29]	RW,1	是否使能 PCacheDataECCSErr_OverLimit 错误登记。
PCacheDataECCMErr_En	[28]	RW,1	是否使能 PCacheDataECCMErr 错误登记。
DMARCplTimeout_En	[27]	RW,1	是否使能 DMARCplTimeout 错误登记。
PIOLengthErr_En	[26]	RW,1	是否使能 PIOLengthErr 错误登记。
DMAWLengthErr_En	[24]	RW,1	是否使能 DMAWLengthErr 错误登记。
DMARHardWall_En	[22]	RW,1	是否使能 DMARHardWall_NotAllOneRsp 错误登记。
DMAWHardWall_En	[21]	RW,1	是否使能 DMAWHardWall 错误登记。
DMAAddrErr_En	[20]	RW,1	是否使能 DMAAddrErr 错误登记。
MSIErr_En	[19]	RW,1	是否使能 MSIErr 错误登记。
PIOCpl_CRS_En	[18]	RW,1	是否使能 PIOCpl_CRS 错误登记。
PIOCplTimeout_En	[17]	RW,1	是否使能 PIOCplTimeout 错误登记。
PIOCpl_UR_En	[16]	RW,1	是否使能 PIOCpl_UR 错误登记。
PIOCpl_CA_En	[15]	RW,1	是否使能 PIOCpl_CA 错误登记。
IPUReqECCSErrOverLimit_En	[14]	RW,1	是否使能 IPUReqECCSErr_OverLimit 错误登记。
IPUReqECCMErr_En	[12]	RW,1	是否使能 IPUReqECCMErr 错误登记。
MCUECCSErrOverLimit_En	[12]	RW,1	是否使能 MCUECCSErr_OverLimit 错误登记。
MCUECCMErr_En	[11]	RW,1	是否使能 MCUECCMErr 错误登记。
PIOReqPTypeErr_En	[10]	RW,1	是否使能 PIOReqPTypeErr 错误登记。
PIOReqAddrErr_En	[9]	RW,1	是否使能 PIOReqAddrErr 错误登记。
PIOWrReqDVLDErr_En	[8]	RW,1	是否使能 PIOWrReqDVLDErr 错误登记。
IPUCplECCSErrOverLimit_En	[7]	RW,1	是否使能 IPUCplECCSErr_OverLimit 错误登记。
IPUCplECCMErr_En	[6]	RW,1	是否使能 IPUCplECCMErr 错误登记。

IPUDMAWrErrAck_En	[5]	RW,1	是否使能 IPUDMAWrErrAck 错误登记。
IPUDMARdErrAck_En	[4]	RW,1	是否使能 IPUDMARdErrAck 错误登记。
DMAReqFifoOverflowErr_En	[3]	RW,1	是否使能 DMAReqFifo_OverflowErr 错误登记。
DMAWrDataFifoOverflowErr_En	[2]	RW,1	是否使能 DMAWrDataFifo_OverflowErr 错误登记。
DMAWrDataFifoUnferflowErr_En	[1]	RW,1	是否使能 DMAWrDataFifo_UnferflowErr 错误登记。
DMAWrData_Poison_En	[0]	RW,1	是否使能 DMAWrData_Poison 错误登记。
—	其它	—	保留。

### 5.1.2.11 DMACpl\_SErrCnt

DMA 响应 ECC 单错计数器，可读写，且写清 0。

表 5-43: DMACpl\_SErrCnt 寄存器域的描述

名称	范围	类型	描述
DMACplECCSErrCnt	[31:0]	RWC,0	来自 IPU 的 DMAR 响应数据包发生 ECC 单错计数器。该计数器设定为饱和计数，全 1 时停止计数并
—	其它	—	保留。

### 5.1.2.12 PIOReq\_SErrCnt

PIO 请求 ECC 单错计数器，可读写，且写清 0。

表 5-44: PIOReq\_SErrCnt 寄存器域的描述

名称	范围	类型	描述
PIOReqECCSErrCnt	[31:0]	RWC,0	来自 IPU 的 PIO 请求数据包发生 ECC 单错计数器。该计数器设定为饱和计数，全 1 时停止计数并保持。
—	其它	—	保留。

### 5.1.2.13 MCUREq\_SErrCnt

MCU 请求 ECC 单错计数器，可读写，且写清 0。

表 5-45: MCUREq\_SErrCnt 寄存器域的描述

名称	范围	类型	描述
MCUREqECCSErrCnt	[31:0]	RWC,0	来自 MCU 的请求数据包发生 ECC 单错计数器。该计数器设定为饱和计数，全 1 时停止计数并保持。
—	其它	—	保留。

### 5.1.2.14 PCData\_SErrCnt

页表 Cache 数据单错计数寄存器，对页式代换过程中发现的页表 Cache 数据阵列 ECC 单错进行计数，可读写，且写清 0。

表 5-46: PCData\_SErrCnt 寄存器域的描述

名称	范围	类型	描述
PCDATA_SErrCnt	[31:0]	RWC,0	页表 Cache 的数据阵列的 ECC 校验单错计数。该计数器为饱和计数，全 1 时停止计数并保持。该计数器的计数达到 ECCSErrLimit 指定的阈值后，会报单错预警给 MCU。
—	其它	—	保留。

### 5.1.2.15 PCTag\_PErrCnt

页表 Cache TAG 偶校验错计数寄存器，对页式代换过程中发现的页表 Cache Tag 阵列偶校验错进行计数，可读写，且写清 0。

表 5-47: PCTag\_PErrCnt 寄存器域的描述

名称	范围	类型	描述
PCTAG_PErrCnt	[31:0]	RWC,0	页表 Cache 的 TAG 阵列的偶校验错计数。该计数器设定为饱和计数，全 1 时停止计数并保持。该计数器的计数达到 ECCSErrLimit 指定的阈值后，会报单错预警给 MCU。
—	其它	—	保留。

### 5.1.2.16 ECCSErrLimit

ECC 单错计数阈值寄存器，可读写。

表 5-48: ECCSErrLimit 寄存器域的描述

名称	范围	类型	描述
ECCSErrLimit	[31:0]	RW,0xffffffff	ECC 单错阈值寄存器。如果来自 IPU 的 PIO 请求数据包、或 DMAR 响应数据包、或来自 MCU 的请求数据包的 ECC 单错计数超过了该阈值，则在 ErrReport 寄存

—	其它	—	保留。
---	----	---	-----

### 5.1.2.17 RCCControl

RC 控制寄存器，配置 RC 的行为参数，可读写。

表 5-49: RCCControl 寄存器域的描述

名称	范围	类型	描述
cfg_l1_clk_removal_en	[3]	RW,0	配置该位为“1”，则链路在 L1 状态下也可以关闭时钟。
tx_lane_flip_en	[2]	RW,0	控制传输通道手动进行通道翻转 (lane reversal)，可用于硬件调试。
rx_lane_flip_en	[1]	RW,0	控制接收通道手动进行通道翻转 (lane reversal)，可用于硬件调试。
app_ltssm_enable	[0]	RW,1	复位后，软件可设置该位为 0，控制链路训练状态机停留在 Detect 状态，以便有足够时间配置 RC 内部寄存器；等配置结束后，再将该位设置为 1。链路正常工作后，对该位先配为 0 再配为 1，可以触发链路热复位。
—	其它	—	保留。

### 5.1.2.18 RCPowerManage

RC 电源管理寄存器，配置 RC 电源管理行为参数，可读写。

表 5-50: RCPowerManage 寄存器域的描述

名称	范围	类型	描述
sys_aux_pwr_det	[3]	RW,0	指示当前系统是否配置有辅助电源。
app_req_entr_l1	[2]	RW,0	应用 (application, 指 PIU) 要求链路进入 L1 状态，需要 ASPM 支持 L1 状态，当核心正在处理传输事务时会忽略该请求。
app_req_exit_l1	[1]	RW,0	应用请求退出 L1 状态，仅在 ASPM 支持 L1 状态时有用。
app_ready_entr_l23	[0]	RW,1	应用准备好进入 L2、L3 状态，当设置该信号为“0”时，核心推迟发 PM_Enter_L23 消息 (对应 PM_Turn_Off) 直到变为“1”。
—	其它	—	保留。

### 5.1.2.19 RCDebugInf0

RC 调试信息寄存器 0，记录 RC 链路传输方面的调试信息，只读，对该寄存器的写请求被忽略 但不报错。

表 5-51: RCDebugInf0 寄存器域的描述

名称	范围	类型	描述
cxpl_debug_info	[63:0]	RO,0	<p>RC 调试信息： [63]: 链路禁止数据缠绕；</p> <p>[62]: 链路训练状态处于禁用 (Disable) 状态，链路不可用； [61]: 链路正在训练中； [60]: 链路正在测试极性颠倒 (polarity reversal)； [59]: 链路需要复位； [54]: PIPE 接口的接收器检测、或环回测试请求； [53]: PIPE 发出电气空闲请求； [52]: PIPE 发出兼容性测试 (compliance) 请求； [51]: 应用申请复位链路； [47:40]: 链路接收方通告/确认的通道数； [37]: 发送了 SKP 有序集；</p> <p>[36]: PMA 报告链路成功 (link up)； [35]: 接收器报告收到 SKP 符号； [34]: 接收器报告收到 TS1 序列 (脉冲)； [33]: 接收器报告收到 TS2 序列 (脉冲)； [32]: 接收器检测到通道反向 (lane reversal)； [31:28]: 链路另一端通告的链路控制位； [27]: 协商通道数目时在相应的域为 PAD 空符号； [26]: 协商链路数目时在相应的域为 PAD 空符号； [25]: 接收方收到逻辑空闲； [24]: 第 2 个符号依然空闲 (只在 Gen2 模式下 PIPE 接口宽度为 16 时有意义)；</p> <p>[23:8]: PIPE 接口传输的数据； [7:6]: PIPE 接口传输的数据是否为控制字符的指示； [5:0]: LTSSM 当前的状态；</p> <p>其它: 保留。</p>



### 5.1.2.20 RCDebugInf1

RC 调试信息寄存器 1，只读。

表 5-52: RCDebugInf1 寄存器域的描述

名称	范围	类型	描述
aux_pm_en	[14]	RO,0	辅助电源使能。
pm_pme_en	[13]	RO,0	电源管理事件使能。
pm_status	[12]	RO,0	电源管理事件状态。
pm_dstate	[11:9]	RO,0	设备的电源管理状态 (DSTATE)。
xmlh_ltssm_state	[8:4]	RO,0	链路训练状态。
pm_curnt_state	[3:1]	RO,0	指示当前电源管理状态。
rdlh_link_up	[0]	RO,0	接收方链路建立。
—	其它	—	保留。

### 5.1.2.21 RCElectroControl

RC 电气特性控制寄存器，可读写。

表 5-53: RCElectroControl 寄存器域的描述

名称	范围	类型	描述
sys_atten_button_pressed	[7]	RW,0	指示按了系统警示按钮 (attention button)。
sys_pre_det_state	[6]	RW,0	指示插槽中有无插卡。
sys_mrl_sensor_state	[5]	RW,0	指示 MRL (Manually-operated Retention Latch) 传感器的状态。
sys_pwr_fault_det	[4]	RW,0	指示发现插槽供电错误。
sys_mrl_sensor_chged	[3]	RW,0	指示 MRL 传感器的状态有变化。
sys_pre_det_chged	[2]	RW,0	指示插槽插卡在位检测状态有变化。
sys_cmd_cpled_int	[1]	RW,0	指示热插拔控制器完成了一条命令。
sys_eml_interlock_engaged	[0]	RW,0	指示系统电气互锁设备有无被占用。
—	其它	—	保留。

### 5.1.2.22 RCElectroStatus

RC 电气特性状态寄存器，只读。

表 5-54: RCElectroStatus 寄存器域的描述

名称	范围	类型	描述
cfg_eml_control	[0]	RO,0	电气互锁设备控制状态。
—	其它	—	保留。

### 5.1.2.23 RCHotRst

RC 复位寄存器，控制 RC 生成热复位，可读写。

表 5-55: RCHotRst 寄存器域的描述

名称	范围	类型	描述
app_init_rst	[0]	RW,0	应用申请复位，以对链路重新进行训练。链路正常工作后，对该位先配为 1 再配为 0，可以触发链路热复位。
—	其它	—	保留。

### 5.1.2.24 MSGControl0

MSG 控制寄存器 0。写该地址即产生一个脉冲信号控制 app\_unlock\_msg；读返回全 0。

### 5.1.2.25 MSGControl1

MSG 控制寄存器 1。写该地址即产生一个脉冲信号控制 apps\_pm\_xmt\_turnoff；读返回全 0。

### 5.1.2.26 LinkDownDelay

链路复位延迟控制寄存器，可读写。

表 5-56: LinkDownDelay 寄存器域的描述

名称	范围	类型	描述
LinkdownCnt	[7:0]	RW,0x20	PCI-E 链路热复位时，为保证内部逻辑的正确性，需要延迟该热复位操作；该参数即配置了内部生成热复位到
—	其它	—	保留。

### 5.1.2.27 PMAControl

PMA 控制寄存器，可读写。

表 5-57: PMAControl 寄存器域的描述

名称	范围	类型	描述
pcs_tx_swing_low	[57:52]	RWS,0x1c	PMA 发送器振幅调节控制, 低摆幅。
pcs_tx_swing_full	[51:46]	RWS,0x39	PMA 发送器振幅调节控制, 全摆幅。
pcs_tx_deemph_gen1	[45:41]	RWS,0xc	Gen1 模式时 PMA 发送器 boost 调节控制。
pcs_tx_deemph_gen2_6db	[40:36]	RWS,0x18	Gen2 模式时 PMA 发送器 boost 调节控制, low boost。
pcs_tx_deemph_gen2_3p5db	[35:31]	RWS,0xe	Gen2 模式时 PMA 发送器 boost 调节控制, high boost。
phy_ref_is_fast	[30]	RWS,0	参考时钟频率大于 200MHz 时, 该位必须为“1”, 表示是快钟。 默认情况下, PCI-E 链路的参考时钟为
phy_pipe0_en	[29]	RWS,1	PIPE 接口使能。
phy_pipe_en_gen2	[28]	RWS,1	PCI-E Gen2 模式使能。
phy_vp_is_1p2	[27]	RWS,0	低电源设置。指示芯片 ASIC 的低电压水平。如果芯片的数据电源是 1.2V, 则该信号必须为 1'b1; 但如果芯片的数字电压低于 1.2V, 则必须为
phy_test_powerdown	[26]	RWS,0	所有电路关闭电源的控制。 关闭 PHY 中的所有电阻以进行 IDDQ 测试。该模式下 PHY 不再工作; 该信号撤销后需要重新复位。
phy_test_bypass	[25]	RWS,0	ASIC 接口旁路。旁路掉 ASIC 接口所有的输入到输出, 以进行 ATPG 测试。
phy_test_burnin	[24]	RWS,0	激活所有电路。 激活 PHY 的所有电路以进行 burn-in 测试。该信号在 test_powerdown 有效时不能有
phy_rtune_req	[23]	RWS,0	调节阻抗请求。
phy_res_req_in	[22]	RWS,0	多个 PHY 共享参考电阻的仲裁信号线 (arbitration lines), 与参考电阻调节有关。
phy_mpll_multiplier	[19:13]	RWS,0x19	MPLL (Multiplying PLL, 倍频锁相环) 倍频参数。

phy_los_level	[12:8]	RWS,0x9	LOS(Loss Of Singal, 信号丢失敏感度)检测器灵敏度水平控制。
phy_acjt_level	[7:3]	RWS,0x12	1149.6 接收器灵敏度水平控制。
phy_common_clocks	[1]	RWS,0	在接收通路上选择公共时钟模式 (common clock mode)。控制信号, 说明链路两端连接了相同的参考 时钟, 即 RX 弹性缓冲两侧的时钟偏斜较小, 可以缩短弹性缓冲的穿透延时。
pipe_tx2rx_loopbk	[0]	RWS,0	从 PIU 到 PCS (Physical Coding Sub-layer, 物理编码子层, 在 RC 和 PMA 之间) 的环回测试, 进行芯片测试时使用。
—	其它	—	保留。

## 5.2 PCI-E 根部件内部的 IO 寄存器

PCI-E 根部件 (RC) 内部的 IO 寄存器遵从 PCI-E 2.0 规范, 按照 4B 对界编址。申威 411 处理器对这些寄存器的访问必须是 32 位粒度的。如果写操作访问只读属性的寄存器 (或寄存器域), 则 对该寄存器 (或寄存器域) 不执行写操作, 且返回正常响应。

### 5.2.1 RC 寄存器空间布局

申威 411 处理器的 PCI-E RC 实现了一个功能 (function), 其包含 4096 字节的 PCI-E 配置空间。该配置空间可划分为:

- 1) 64 字节的 PCI 3.0 兼容配置空间头;
- 2) PCI 标准能力结构, 起始偏移量 0x40; PCI 兼容配置空间头和 PCI 标准能力结构属于 PCI 配置空间;
- 3) PCI-E 扩展能力结构, 起始偏移量 0x100;
- 4) 端口逻辑寄存器 (申威 411 处理器自定义部分), 起始偏移量 0x700。端口逻辑寄存器主要是用于测试。PCI-E 扩展能力结构和端口逻辑寄存器属于 PCI-E 扩展配置空间。

RC 配置空间的布局如图 6-1 所示:

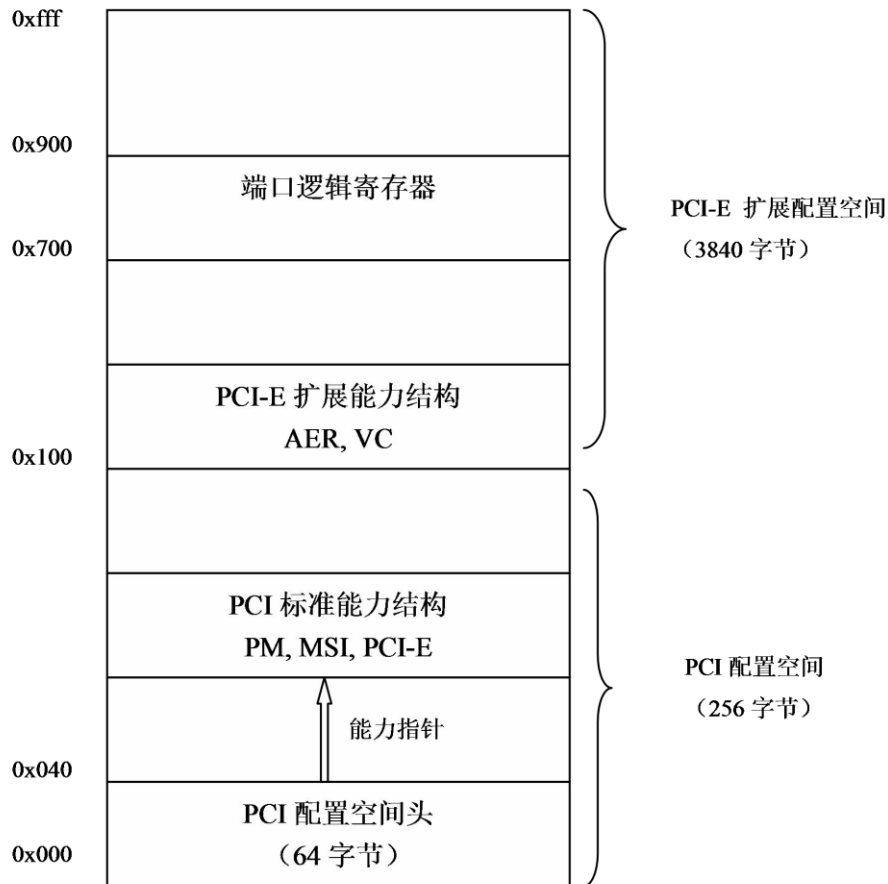


图 5-1 RC 配置空间布局

### 5.2.1.1 寄存器映射

#### 5.2.1.1.1 PCI 配置空间头

PCI 配置空间头的寄存器域定义如表 5-44 所示。该配置空间在 PCI-E 规范中属于 TYPE1 类型的配置空间（区别于 EP 的 TYPE0 类型配置空间）。大多数 PCI 兼容的寄存器域在 PCI 3.0 和 PCI-E 中有相同的软件含义。

表 5-58: PCI 配置空间头

字节偏移	字节 3	字节 2	字节 1	字节 0
0x00	设备 ID		厂商 ID	
0x04	PCI 状态寄存器		PCI 命令寄存器	
0x08	类代码			修订版 ID
0x0c	BIST	头类型	主延迟计时器	Cache 行大小
0x10	保留			
0x14	保留			
0x18	二级延迟计时器	从属总线号	二级总线号	主总线号

0x1c	二级状态	I/O 界限	I/O 基址
0x20	存储器界限	存储器基址	
0x24	可预取存储器界限	可预取存储器基址	
0x28	可预取存储器基址高 32 位		
0x2c	可预取存储器界限高 32 位		
0x30	I/O 界限高 16 位	I/O 基址高 16 位	
0x34	保留		能力指针
0x38	扩展 ROM 基址		
0x3c	桥控制	中断引脚	中断线

### 5.2.1.1.2 PCI 能力结构

申威 411 处理器支持三种 PCI 标准能力结构。每个结构都有自己的能力 ID (Capability ID)，且起始地址都由指针指定，这些指针就形成了一个链表，如表 5-45 所示。其中第一个能力结构为 PCI 电源管理能力，其起始地址在 PCI 兼容配置空间头偏移为 0x34 的寄存器（即能力指针）中指定，并有专门的寄存器指定下一个能力结构的起始地址；这样依次组成链表。

三种能力结构，及其起始地址和链表组织情况如表 5-45 所示，各种能力结构的寄存器列表如表

5-46~5-48 所示：

表 5-59：PCI 能力结构偏移和指针

起始偏移地址	能力名称	下一个能力指针
0x00	——	0x40
0x40	PCI 电源管理能力	0x50
0x50	MSI 中断能力	0x70
0x70	PCI-E 能力	——

表 5-60：电源管理能力结构

字节偏移	字节 3	字节 2	字节 1	字节 0
0x40	电源管理能力寄存器		下一能力指针 (0x50)	能力 ID (0x01)
+0x04	电源管理控制状态寄存器			

表 5-61：MSI 能力结构

字节偏移	字节 3	字节 2	字节 1	字节 0
------	------	------	------	------

0x50	MSI 控制寄存器	下一能力指针 (0x70)	能力 ID (0x05)
0x54	MSI 低 32 位地址寄存器		
0x58	MSI 高 32 位地址寄存器		
0x5C	保留	MSI 数据寄存器	

表 5-62: PCI-E 能力结构

字节偏移	字节 3	字节 2	字节 1	字节 0
0x70	PCI-E 能力寄存器		下一能力指针 (0xB0)	能力 ID (0x10)
+0x4	设备能力寄存器			
+0x8	设备状态寄存器		设备控制寄存器	
+0xC	链路能力寄存器			
+0x10	链路状态寄存器		链路控制寄存器	
+0x14	插槽能力寄存器			
+0x18	插槽状态寄存器		插槽控制寄存器	
+0x1C	根能力寄存器			
+0x20	根状态寄存器		根控制寄存器	
+0x24	设备能力 2			
+0x28	设备状态 2 寄存器		设备控制 2 寄存器	
+0x2C	链路能力 2 寄存器			
+0x30	链路状态 2 寄存器		链路控制 2 寄存器	
+0x34	插槽能力 2 寄存器			
+0x38	插槽状态 2 寄存器		插槽控制 2 寄存器	

### 5.2.1.1.3 PCI-E 扩展能力结构

申威 411 处理器支持两种 PCI-E 扩展能力结构。每个结构都有自己的能力 ID (Capability ID)，且组织方式类似于 PCI 标准能力结构。第一个能力结构为高级错误报告能力结构，其在 RC 配置空间中的偏移量是 0x100，并有指针指向下一个 PCI-E 扩展能力结构。

两种能力结构，及其起始地址和链表组织情况如表 5-49 所示，各种能力结构的寄存器列表如表

5-50~5-51 所示：

表 5-63: PCI-E 扩展能力结构偏移和指针

起始偏移地址	能力名称	下一个能力指针
0x100	高级错误报告能力	0x140
0x140	虚通道能力	—

表 5-64: 高级错误报告能力结构

字节偏移	字节 3	字节 2	字节 1	字节 0
0x100	PCI-E 扩展能力头			
+0x4	不可纠正错状态寄存器			
+0x8	不可纠正错屏蔽寄存器			
+0xC	不可纠正错严重性寄存器			
+0x10	可纠正错状态寄存器			
+0x14	可纠正错屏蔽寄存器			
+0x18	高级错能力和控制寄存器			
+0x1C ~ +0x28	头记录寄存器			
+0x2C	根错误命令寄存器			
+0x30	根错误状态寄存器			
+0x34	错误源标志寄存器			

表 5-65: 虚通道能力结构

字节偏移	字节 3	字节 2	字节 1	字节 0
0x140	PCI-E 扩展能力头			

注：申威 411 处理器的 RC 中只实现了 1 个虚通道，即虚通道 0。

#### 5.2.1.1.4 端口逻辑寄存器

端口逻辑寄存器是申威 411 处理器自定义的寄存器，其映射关系如表 5-52 所示。

表 5-66: 端口逻辑寄存器映射

序号	名称	索引地址
1	ACK 延迟计时和重发计时寄存器	0x700
2	其它消息寄存器	+0x4
3	端口强制链路寄存器	+0x8
4	ACK 频率寄存器	+0xC
5	端口链路控制寄存器	+0x10



6	通道偏斜寄存器	+0x14
---	---------	-------

7	符号数目寄存器	+0x18
8	符号计时寄存器和过滤屏蔽寄存器 1	+0x1C
9	过滤屏蔽寄存器 2	+0x20
10	调试寄存器 0	+0x28
11	调试寄存器 1	+0x2C
12	传输 P 类事务流量控制信用状态寄存器	+0x30
13	传输 NP 类事务流量控制信用状态寄存器	+0x34
14	传输完成事务流量控制信用状态寄存器	+0x38
15	队列状态寄存器	+0x3C
16	Gen2 模式	+0x10C
其它	保留	—

### 5.2.1.2 RC 配置空间寄存器的访问

在申威 411 处理器中，RC 配置空间的访问基址为 0xC8,0000,0000。本章描述 RC 配置空间内部的寄存器偏移都是 4B 对界的，以与 PCI-E 规范保持一致。在实际访问中，需要转化为申威 411 处理器所规定的地址格式，RC 配置寄存器偏移与软件访问地址偏移量的关系为：寄存器偏移值左移 5 位即得到访问地址中的偏移量。如偏移为 0x38 的寄存器，其软件访问地址的偏移量为 0x700，访问地址为 0xC8,0000,0700。

尽管 RC 配置空间中有些寄存器域是只读类型（HwInit、RO、ROS 等），但是软件对这些寄存器的写操作依然会成功，且返回正常响应。

## 5.2.2 PCI 兼容 配置头寄存器

### 5.2.2.1 设备 ID 和厂商 ID 寄存器

偏移：  
0x00。 表5-67：设备ID和厂商ID寄存器描述

名称	范围	类型	描述
DeviceID	[31:16]	RO,0xABCD	设备 ID。
VendorID	[15:0]	RO,0x16C3	厂商 ID。

### 5.2.2.2 PCI 命令寄存器

偏移：0x04。

字节：0-1。

表5-68：命令寄存器描述

名称	范围	类型	描述
INTxDisable	[10]	RW,0	禁用 INTx 中断类型。
FastB2BEn	[9]	RO,0	快速背靠背使能。不适用于 PCI-E，必须固定为“0”。
SERREn	[8]	RW,0	SERR#使能。为 1 时允许向 RC 报告致命错和非致命错，即允许“PCI 状态寄存器”报告“SignaledSysErr”。
IDSEL	[7]	RO,0	IDSEL Stepping/ Wait Cycle Control。不适用于 PCI-E，必须固定为“0”。
ParErrRspEn	[6]	RW,0	奇偶校验错响应。为 1 时允许“PCI 状态寄存器”报告“MasterDataParErr”错误。
VGASnoop	[5]	RO,0	VGA 调色监听。不适用于 PCI-E，必须固定为“0”。
MemWrInv	[4]	RO,0	存储器写和置无效。不适用于 PCI-E，必须固定为“0”。
SpecialCycleEn	[3]	RO,0	特殊周期使能。不适用于 PCI-E，必须固定为“0”。
BusMasterEn	[2]	RW,0	总线主设备使能。控制上行方向上是否允许转发存储器、I/O 空间请求，但不影响上行方向的完成、消息事务，也不影响下行方向事务。 0：端口收到的上行方向的存储器、I/O 空间请求为 UR 请求，并需要返回非法响应。1：允许上行方向的存储器、I/O 空间请求。 PCI-E 链路链路训练成功后，需要设置该位为 1，以使链路能正常工作。
MemSpaceEn	[1]	RW,0	存储器空间使能。 0：忽略端口一级总线上的存储器空间访问。 1：接收并响应端口一级总线上的存储器空间访问。 PCI-E 链路链路训练成功后，需要设置该位为 1，以使链路能正常工作。
IOSpaceEn	[0]	RW,0	I/O 空间使能。 0：忽略端口一级总线上的存储器空间访问。 1：接收并响应端口一级总线上的存储器空间访问。

			PCI-E 链路链路训练成功后，需要设置该位为1，以使链路能正常工作。
—	其它	—	保留。

### 5.2.2.3 PCI 状态寄存器

偏移：0x04。

字节：2-3。

表5-69：状态寄存器描述

名称	范围	类型	描述
ParityErr	[15]	RW1C,0	检测到奇偶校验错。 不管“PCI 命令寄存器”的“ParErrRspEn”是否使能，只要在其一级总线上的 TLP 发生了数据中毒，就设置该位。该位实际上等同于“不可修正错状态寄存器”的“PoisonedTLPStatus”位。
SignaledErr	[14]	RW1C,0	报告系统错。 如果“PCI 控制寄存器”的“SERREn”为1，则该端口在向上游发送（含转发）ERR_FATAL、ERR_NONFATAL 消息时，需要设置该位。 该位实际上等同于“设备状态寄存器”中的“FatalErrDtct”、“NonFatalErrDtct”位。
RcvMasterAbort	[13]	RW1C,0	接收到主设备异常中止。
RcvTrgtAbort	[12]	RW1C,0	接收到目标方异常中止。
SignaledTrgtAbort	[11]	RW1C,0	报告目标方异常。 如果端口收到的存储器请求其目标端口不存在，则设置该位。
DEVSELTiming	[10:9]	RO,0	DEVSEL Timing。不适用于PCI-E，保留。
MasterDataParErr	[8]	RW1C,0	主设备数据奇偶校验错。 如果“PCI 命令寄存器”的“ParErrRspEn”使能，当以下事件发生时，端口需设置该位：1) 从二级总线向一级总线方向转发数据中毒的写请求； 2) 从一级总线方向收到了数据中毒的完成事务。

			该位实际上等同于“不可修正错状态寄存器”的“PoisonedTLPStatus”位。
FastB2BTranscCap	[7]	RO,0	快速背靠背事务能力。不适用于 PCI-E，硬连线为“0”。
66MhzCap	[5]	RO,0	66 Mhz 是否可用。不适用于 PCI-E，硬连线为“0”。
CapList	[4]	RO,1	能力列表。表示存在扩展能力，硬连线为“1”。
INTxStat	[3]	RO,0	INTx 状态。 0: 没有待发送的 INTx 消息； 1: 有待发送的 INTx 消息或 PEX_INTA#有效。
—	其它	—	保留。

#### 5.2.2.4 修订版 ID 寄存器

偏移：0x08。

字节：0。

表5-70：修订版ID寄存器描述

名称	范围	类型	描述
RevisionID	[7:0]	RO,0x01	修订版本。

#### 5.2.2.5 类代码寄存器

偏移：0x08。

字节：1-3。

表5-71：类代码寄存器描述

名称	范围	类型	描述
BaseClassCode	[23:16]	RO, 0x00	基本类代码。
SubClassCode	[15:8]	RO, 0x00	子类代码。
ProgramInf	[7:0]	RO, 0x00	寄存器级的编程接口。

#### 5.2.2.6 Cache 行大小寄存器

偏移：0x0C。

字节：0。

表5-72：Cache 行大小寄存器描述

名称	范围	类型	描述
----	----	----	----

CacheLineSize	[7:0]	RW, 0x00	Cache 行大小。该寄存器是为了兼容 PCI 设备，并不适用于 PCI-E 设备；修改该域不影响功能。
---------------	-------	----------	------------------------------------------------------

### 5.2.2.7 主延迟计时器寄存器

偏移：0x0C。

字节：1。

表5-73：主延迟计时器寄存器描述

名称	范围	类型	描述
MasterLatencyTimer	[7:0]	RO,0	主延时计数器。不适用于 PCI-E 设备，硬连线为“0”。

### 5.2.2.8 头类型寄存器

偏移：0x0C。

字节：2。

表5-74：头类型寄存器描述

名称	范围	类型	描述
MultiFunc	[7]	RO,0	多功能设备标志。
HeaderFmt	[6:0]	RO,0x1	配置头格式。硬连线为 0x01。

### 5.2.2.9 BIST 寄存器

偏移：0x0C。

字节：3。

表5-75：BIST 寄存器描述

名称	范围	类型	描述
BIST	[7:0]	RO,0x0	申威 1610/410 不支持 RC 的 BIST 寄存器功能，读该寄存器将返回全“0”。

### 5.2.2.10 总线号寄存器

偏移：0x18。

表5-76：总线号寄存器描述

名称	范围	类型	描述
2ndLatencyTimer	[31:24]	RO, 0x00	二级延迟计时器。不适用于 PCI-E，硬连线为 0x00。

SubordinateBusNum	[23:16]	RW, 0x00	从属总线号。
2ndBusNum	[15:8]	RW, 0x00	二级总线号。
PrimaryBusNum	[7:0]	RW, 0x00	主总线号。

### 5.2.2.11 I/O 基址和界限寄存器

偏移：0x1C。

字节：0-1。

表5-77：I/O 基址和界限寄存器描述

名称	范围	类型	描述
IOLimit	[15:12]	RW, 0x00	I/O 空间界限[15:12]。 I/O 界限地址[11:0]默认为 0xFF。
32bitIO	[8]	RO, 0	32 位 I/O 空间。 “0”：16 位 I/O 编址寻址； “1”：32 位 I/O 编址寻址。
IOBase	[7:4]	RW, 0x00	I/O 空间基址[15:12]。 I/O 基址地址[11:0]默认为 0x000。
32bitIO	[0]	RO, 0	32 位 I/O 空间。 “0”：16 位 I/O 编址寻址； “1”：32 位 I/O 编址寻址。 该位被修改时，第[8]位也一同被修改。
—	其它	—	保留。

### 5.2.2.12 二级状态寄存器

偏移：0x1C。

字节：2-3。

表5-78：二级状态寄存器描述

名称	范围	类型	描述
ParityErrDtct	[15]	RW1C, 0	检测到奇偶校验错。 下游端口从上游端口收到数据中毒的 TLP 时，就会设置该位为 1，没有使能位控制。
RcvSysErr	[14]	RW1C, 0	接收到系统错。 下游端口接收到 ERR_FATAL 或 ERR_NONFATAL 消息





			时，就会设置该位为 1。
RcvMasterAbort	[13]	RW1C, 0	接收到主设备异常中止。
RcvTrgtAbort	[12]	RW1C, 0	接收到目标方异常中止。
SignalTrgtAbort	[11]	RW1C, 0	报告目标方异常中止。
DEVSELTiming	[10:9]	RO, 0	DEVSEL# Timing。不适用于 PCI-E，硬连线为“0”。如果“桥控制寄存器”的“ParErrRspEn”使能，则以下事件发生时，下游端口会设置该位有效：1) 收到数据中毒的完成事务；2) 转发数据中毒的写事务。 该位实际上等同于“不可修正错状态寄存器”的“PoisonedTLPStatus”位。
MasterDataParErr	[8]	RW1C, 0	主设备数据奇偶校验错。
FastB2BTranscCap	[7]	RO, 0	快速背靠背事务能力。不适用于 PCI-E，硬连线为“0”。
66MhzCap	[5]	RO, 0	66 Mhz 是否可用。不适用于 PCI-E，硬连线为“0”。
—	其它	—	保留。

### 5.2.2.13 存储器基址和界限寄存器

偏移：0x20。

表5-79：存储器基址和界限寄存器描述

名称	范围	类型	描述
MemLimit	[31:20]	RW, 0x00	不可预取存储器空间界限地址[31:20]。 界限地址[19:0]默认为 0xF_FFFF。
MemBase	[15:4]	RW, 0x00	不可预取存储器空间基地址[31:20]。 基地址[19:0]默认为 0x0_0000。
—	其它	—	保留。

### 5.2.2.14 可预取存储器基址和界限寄存器

偏移：0x24。

表5-80：可预取存储器基址和界限寄存器描述

名称	范围	类型	描述
PreMemLimit	[31:20]	RW, 0x000	可预取存储器空间界限地址[31:20]。 界限地址[19:0]默认为 0xF_FFFF。
64bitPreMem	[16]	RO, 0	存储器空间编址标志。

			“0”：32 位编址寻址； “1”：64 位编址寻址。
PreMemBase	[15:4]	RW,0x000	可预取存储器空间基地址[31:20]。 基地址[19:0]默认为 0x0_0000。
64bitPreMem	[0]	RO,0	存储器空间编址标志。 “0”：32 位编址寻址； “1”：64 位编址寻址。 该位被修改时，第[16]位也一同被修改。
—	其它	—	保留。

### 5.2.2.15 可预取存储器基址高 32 位寄存器

偏移：0x28。

表5-81：可预取基址高32 位寄存器描述

名称	范围	类型	描述
PreMemBaseUpper	[31:0]	RW,0x00000000	64 位可预取存储器空间基址的高 32 位。只有当可预取存储器空间编址为 64 位模式时有效。

### 5.2.2.16 可预取存储器界限高 32 位寄存器

偏移：0x2C。

表5-82：可预取存储器界限高32 位寄存器描述

名称	范围	类型	描述
PreMemLimitUpper	[31:0]	RW,0x00000000	64 位可预取存储器空间界限地址的高 32 位。只有当可预取存储器空间编址为 64 位模式时有效。

### 5.2.2.17 I/O 基址和界限高 16 位寄存器

偏移：0x30。

表5-83：I/O 基址和界限高16 位寄存器描述

名称	范围	类型	描述
IOLimitUpper16	[31:16]	RW,0x0000	32 位 I/O 空间界限的高 16 位（如果设备支持 32 位 I/O 地址译码）。

IOBaseUpper16	[15:0]	RW,0x0000	32 位 I/O 空间基址的高 16 位（如果设备支持 32 位 I/O 地址译码）。
---------------	--------	-----------	---------------------------------------------

### 5.2.2.18 能力指针寄存器

偏移：0x34。

字节：0。

表5-84：功能指针寄存器描述

名称	范围	类型	描述
FirstCapPtr	[7:0]	RO,0x40	第一个能力指针。默认指向电源管理能力结构。

### 5.2.2.19 扩展 ROM 基址寄存器

偏移：0x38。

表5-85：扩展ROM 基址寄存器寄存器描述

名称	范围	类型	描述
ExpROMBase	[31:11]	RW,0x00000	扩展 ROM 空间基地址。不支持。
ExpROMEn	[0]	RW,0	扩展 ROM 使能。不支持。
—	其它	—	保留。

### 5.2.2.20 中断线寄存器

偏移：0x3C。

字节：0。

表5-86：中断线寄存器描述

名称	范围	类型	描述
IntrLine	[7:0]	RW,0xFF	中断线。其值因系统结构而异，当初始化或配置系统的时候，软件会写入路由信息。

### 5.2.2.21 中断引脚寄存器

偏移：0x3C。

字节：1。

表5-87: 中断引脚寄存器描述

名称	范围	类型	描述
IntrPin	[7:0]	RO,0x01	中断引脚。指定设备（或者设备功能）使用的中断消息。 有效值如下： 0x00: 设备（或者功能）没有使用 INTx 中断； 0x01: 设备（或者功能）使用了 INTA 中断； 0x02: 设备（或者功能）使用了 INTB 中断； 0x03: 设备（或者功能）使用了 INTC 中断； 0x04: 设备（或者功能）使用了 INTD 中断。

### 5.2.2.22 桥控制寄存器

偏移: 0x3C。

字节: 2-3。

表5-88: 桥控制寄存器描述

名称	范围	类型	描述
DscdTimerSERREn	[11]	RO,0	丢弃计时器 SERR 使能状态。不适用于 PCI-E, 硬连线为“0”。
DscdTimerStatus	[10]	RO,0	丢弃计时器状态。不适用于 PCI-E, 硬连线为“0”。
2ndDscdTimer	[9]	RO,0	二级丢弃计时器。不适用于 PCI-E, 硬连线为“0”。
PrimaryDscdTimer	[8]	RO,0	主丢弃计时器。不适用于 PCI-E, 硬连线为“0”。
FastB2BEn	[7]	RO,0	快速背靠背传输使能。不适用于 PCI-E, 硬连线为“0”。
2ndBusRst	[6]	RW,0	二级总线复位。 为 1 时会热复位链路的下游设备。
MasterAbortMode	[5]	RO,0	主设备异常中止模式。不适用于 PCI-E, 硬连线为“0”。
VGA16bDecEn	[4]	RW,0	VGA 16 位译码使能。 “VGAEn” 位或 “PCI 命令寄存器” 的 “VGASnoop” 位有效时, 用来使能 VGA 16 位 I/O 译码和转发功能。 0: 对 VGA I/O 访问执行 10 位地址译码; 1: 对 VGA I/O 访问执行 16 位地址译码。
VGAEn	[3]	RW,0	VGA 使能
ISAEn	[2]	RW,0	ISA 使能。
SERREn	[1]	RW,0	SERR#使能。 控制是否从二级总线接口向主总线接口转发 ERR_COR、ERR_FATAL、ERR_NONFATAL 消息。

ParErrRspEn	[0]	RW,0	奇偶校验错响应使能。
—	其它	—	保留。

## 5.2.3 PCI 标准能力结构寄存器

申威 1610/410 处理器支持三种 PCI 标准能力结构：

- 1) PCI 电源管理能力；
- 2) MSI 中断能力；
- 3) PCI-E 能力。

这三种能力结构中，需要重点关注的是 PCI-E 能力结构的寄存器。

### 5.2.3.1 PCI 电源管理能力

PCI 配置头里的能力指针指向 PCI 电源管理能力结构。该能力结构包含以下寄存器：

#### 5.2.3.1.1 电源管理能力 ID 寄存器

偏移：

0x40。

字节：0。

表5-89：电源管理能力ID 寄存器描述

名称	范围	类型	描述
PMCapID	[7:0]	RO,0x01	电源管理能力 ID。

#### 5.2.3.1.2 电源管理下一能力指针

偏移：

0x40。

字节：1。

表5-90：电源管理下一能力指针寄存器描述

名称	范围	类型	描述
NextCapPtr	[7:0]	RO,0x50	下一能力指针。默认指向 MSI 能力结构。

#### 5.2.3.1.3 电源管理能力寄存器

偏移：

0x40。

字节：

2-3。

表5-91：电源管理能力寄存器描述

名称	范围	类型	描述
----	----	----	----

PME_Support	[15:11]	RO,0x1B	是否支持电源管理事件（PME，Power Management Event）。 指定哪些功耗状态下可以生成 PME 消息。只有为“1”时该位对应的设备电源管理状态才可以产生 PME 消息：
			[11]: D0 状态是否可产生 PME 消息； [12]: D1 状态是否可产生 PME 消息； [13]: D2 状态是否可产生 PME 消息； [14]: D3hot 状态是否可产生 PME 消息； [15]: D3cold 状态是否可产生 PME 消息。 第[11]、[14]、[15]位必须设置。
D2Support	[10]	RO,1	是否支持 D2 电源管理状态。
D1Support	[9]	RO,1	是否支持 D1 电源管理状态。
AuxCurrent	[8:6]	RO,0x7	AUX 电流。
DSI	[5]	RO,0	设备特定初始化（DSI，Device Specific Initialization）。
PMEClock	[3]	RO,0	PME 时钟。硬连线为“0”。
PMVersion	[2:0]	RO,0x3	电源管理规格版本（Power Management Specification Version）。
—	其它	—	保留。

#### 5.2.3.1.4 电源管理控制和状态寄存器

偏移：0x44。

表5-92：电源管理控制和状态寄存器描述

名称	范围	类型	描述
ClkCtrlEn	[23]	RO,0	总线电源/时钟控制使能。硬连线为“0”。
B23Support	[22]	RO,0	是否支持 B2/B3。硬连线为“0”。
PMEStatus	[15]	RW1CS,0	PME 状态。指示是否发生 PME 事件。
PMEEEn	[8]	RWS,0	PME 使能。设为“1”表示设备被使能可以产生 PME。
NoSoftRst	[3]	RO,0	无软件复位。 为 1 表示当从 D3hot 状态迁移回 D0 状态时，不会导致内部复位。
PowerState	[1:0]	RW,0x0	配置设备的电源管理状态： 0b00：D0 状态，正常工作状态； 0b01：D1 状态，可选，低功耗状态； 0b10：D2 状态，可选，低功耗状态； 0b11：D3 状态，最低功耗状态。当特定状态不支持时，写入的值被忽略。
—	其它	—	保留。

#### 5.2.3.2 MSI 能力

##### 5.2.3.2.1 MSI 能力 ID

偏移：0x50。

字节：0。

表 5- 93: MSI 能力 ID 寄存器描述

名称	范围	类型	描述
MSICapID	[7:0]	RO,0x05	MSI 能力 ID。

#### 5. 2. 3. 2. 2 MSI 下一能力指针

偏移：0x50。

字节：1。

表5-94: MSI 下一能力指针寄存器描述

名称	范围	类型	描述
NextCapPtr	[7:0]	RO,0x70	下一功能指针。默认指向 PCI-E 能力。

#### 5. 2. 3. 2. 3 MSI 控制寄存器

偏移：0x50。

字节：2-3。

表 5- 95: MSI 控制寄存器描

名称	范围	类型	描述
MSI64En	7	RO,1	64 位地址能力。
MultiMSIMsgEn	6:4	RW,0	多中断消息使能。 取值 N 时，表示该功能分配有 2^N 个 MSI 中断向量。 该域的设置应不超过 MultiMSIMsgCap 域的值。
MultiMSIMsgCap	3:1	RO,0	多中断消息能力。 取值 N 时，表示该功能需要 2^N 个 MSI 中断向量。
MSIEn	0	RW,0	MSI 中断使能。 当设置该域时，必须禁用 INTx 中断（通过设置“PCI 命令寄存器”中的 INTxDisable 为 1）。
—	其它	—	保留。

#### 5. 2. 3. 2. 4 MSI 低 32 位地址寄存器

偏移：0x54。

表5-96: MSI 低32 位地址寄存器

名称	范围	类型	描述
MSILower32Addr	[31:0]	RW,0x0000_0000	MSI 地址的低 32 位。 [1:0]必须是 0。

#### 5. 2. 3. 2. 5 MSI 高 32 位地址寄存器

偏移：0x58。

表5-97: MSI 高32 位地址寄存器

名称	范围	类型	描述
----	----	----	----

MSIUpper32Addr	[31:0]	RW0x0000_0000	MSI 地址的高 32 位（MSI64En 位使能时）。 [1:0]必须是 0。
----------------	--------	---------------	---------------------------------------------

### 5.2.3.2.6 MSI 数据寄存器

偏移：0x5C。

字节：0-1。

表 5-98: MSI 数据寄	范围	类型	描述
MSIData	[15:0]	RW,0x0000	MSI 数据。数据低位段（位数取决于分配的中断数量）替换为 MSI 中断的向量号，可以标识不同的 MSI 中断。

## 5.2.3.3 PCI-E 能力

### 5.2.3.3.1 PCI-E 能力列表寄存器

偏移：

0x70。

字节：

0-1。

表5-99：PCI-E 能力列表寄存器描述

名称	范围	类型	描述
NextCapPtr	[15:8]	RO,0xB0	下一能力指针。
PCIECapID	[7:0]	RO,0x10	PCI-E 能力 ID。

### 5.2.3.3.2 PCI-E 能力寄存器

偏移：0x70

字节：2-3。

表 5-100：PCI-E 能力寄存器描述

名称	范围	类型	描述
MSIMsgNum	[13:9]	RO,0x0	电源管理事件、或热插拔事件所触发的 MSI 中断所对应的中断向量号。
SlotImplemented	[8]	HwInit,0	是否实现了插槽。
DevicePortType	[7:4]	RO,0x4	设备/端口类型。申威 1610/410 的 PCI-E 端口用作 RC。
PCIECapVersion	[3:0]	RO,0x2	PCI-E 能力版本。
—	其它	—	保留。



### 5.2.3.3.3 设备能力寄存器

偏移：0x74。

表 5- 101：设备能力寄存器描述

名称	范围	类型	描述
RoleErrRpt	[15]	RO,1	基于角色的错误报告（Role-Base Error Reporting）。
EPAcptL1Latency	[11:9]	RO,0	EP 在链路 L1 状态下可接受的延迟。必须固定为“0”。
EPAcptL0sLatency	[8:6]	RO,0	EP 在链路 L0s 状态下可接受的延迟。必须固定为“0”。
ExtTagSupport	[5]	RO,0	是否支持扩展 Tag 域。必须固定为“0”。
PhantomFuncSupport	[4:3]	RO,0	是否支持仿真功能（Phantom Function）。必须固定为“0”。
MaxPayloadSize	[2:0]	RO,0x2	支持的最大有效负载。其编码含义为： 0b000：128B； 0b001：256B； 0b010：512B； 0b011：1024B； 0b100：2048B； 0b101：4096B； 其它：不支持。
—	其它	—	保留。

### 5.2.3.3.4 设备控制寄存器

偏移：0x78。

字节：0-1。

表 5- 102：设备控制寄存器描述

名称	范围	类型	描述
MaxReadReqSize	[14:12]	RW,0x2	最大读请求数据量。
NoSnoopEn	[11]	RW,1	不监听（No Snoop）使能。
AuxPwrPMEn	[10]	RW,0	辅助电源管理使能。
PhantomFuncEn	[9]	RW,0	仿真功能（Phantom Function）使能。
ExtTagEn	[8]	RW,0	扩展 tag 域使能。
MaxPayloadSize	[7:5]	RW,0x0	最大有效负载数。
RelaxedOrderingEn	[4]	RW,1	宽松的序（Relaxed Ordering）使能。
URRptEn	[3]	RW,0	不支持的请求（Unsupported Request）报告使能。
FatalErrRptEn	[2]	RW,0	致命错误报告使能。
NFatalErrRptEn	[1]	RW,0	非致命错误报告使能。
CErrRptEn	[0]	RW,0	可纠正错误报告使能。
—	其它	—	保留。

### 5.2.3.3.5 设备状态寄存器

偏移：0x78。

字节：2-3。

表 5- 103：设备状态寄存器描述

名称	范围	类型	描述
TransPend	[5]	RO,0	事务悬挂。硬连线为“0”。
AuxPwrDtct	[4]	RO,0	检测到 AUX 电源。
URDtct	[3]	RW1C,0	检测到不支持的请求。
FatalErrDtct	[2]	RW1C,0	检测到致命错误。检测到致命错误后直接登记到该域，而不管高级错误报告有没有使能（设备控制寄存器的 FatalErrRptEn 位）。
NfatalErrDtct	[1]	RW1C,0	检测到非致命错误。检测到非致命错误后直接登记到该域，而不管高级错误报告有没有使能（设备控制寄存器的 NFatalErrRptEn 位）。
CerrDtct	[0]	RW1C,0	检测到可修正错误。检测到可纠正错误后直接登记到该域，而不管高级错误报告有没有使能（设备控制寄存器的 CErrRptEn 位）。
—	其它	—	保留。

### 5.2.3.3.6 链路能力寄存器

偏移：0x7C。

表 5- 104：链路能力寄存器描述

名称	范围	类型	描述
PortNum	[31:24]	HwInit,0x00	端口数。
LinkBWNote	[21]	RO,1	链路带宽通告能力。硬连线为“1”。
DLLActiveRpt	[20]	RO,1	数据链接层主动报告能力。硬连线为“1”。
ClkPM	[18]	RO,1	时钟电源管理。在指示链路处于 L1、L2、L3 这些低功耗状态时是否可以移除参考时钟。

L1ExitLatency	[17:15]	RO,0x6	L1 状态退出延时。编码含义为： 0x0: 少于 1us; 0x1: 1us~2us; 0x2: 2us~4us; 0x3: 4us~8us; 0x4: 8us~16us; 0x5: 16us~32us; 0x6: 32us~64us; 0x7: 多于 64us。
---------------	---------	--------	--------------------------------------------------------------------------------------------------------------------------------------------------------------

L0sExitLatency	[14:12]	RO,0x3	L0s 状态退出延时。编码含义为： 0x0: 少于 64ns; 0x1: 64ns~128ns; 0x2: 128ns~256ns; 0x3: 256ns~512ns; 0x4: 512ns~1us; 0x5: 1us~2us; 0x6: 2us~4us; 0x7: 多于 4us。
ASPMSupport	[11:10]	RO,0x3	主动链路电源管理（ASPM, Active State Link Power Management）支持。编码含义为： 0x1: 支持 L0s 状态下的 ASPM; 0x3: 支持 L0s 和 L1 状态下的 ASPM; 其它: 保留。
MaxLinkWidth	[9:4]	RO,0x8	最大链接宽度。
MaxLinkSpeed	[3:0]	RO,0x2	最大链接速度。编码含义： 0b0001: 支持 2.5Gbps 速率; 0b0010: 支持 5.0Gbps 和 2.5Gbps 速率; 其它: 保留。
—	其它	—	保留。

### 5.2.3.3.7 链路控制寄存器

偏移: 0x80。

表 5- 105: 链路控制寄存器描述

名称	范围	类型	描述
LinkAutoBWIntEn	[11]	RO,0	链路自动调整带宽中断使能。设置时会使能该类型中断的生成, 来指示链接自动调整带宽 状态位已经被设置。

LinkBWMgtIntEn	[10]	RO,0	链路带宽管理中断使能。设置时会使能该类型中断的生成，来指示链路带宽管理状态位已经被设置。
ClkPMEEn	[8]	RW,0	时钟电源管理使能。 如果链路能力寄存器中 ClkPM 位为“0”，该位也为“0”。
ExtSsynch	[7]	RW,0	扩展同步。
ComClk	[6]	RW,0	公共时钟配置。
RetrainLink	[5]	RW,0	重新训练链路。
LinkDisable	[4]	RW,0	链路禁用。
RCB	[3]	RO,0	读完成边界（Read Completion Boundary）。编码含义为： 0: 64B;

			1: 128B。
ASPMCtrl	[1:0]	RW,0	主动链路电源管理控制。
—	其它	—	保留。

### 5.2.3.3.8 链路状态寄存器

偏移: 0x80。

字节: 2-3。

表 5-106: 链路状态寄存器描述

名称	范围	类型	描述
LinkAutoBWStatus	[15]	RW1C,0	链路自动调整带宽状态。该位表示硬件自动调节了链路速度或宽度。如果物理层报告速度或者宽度因下游设备发起而发生改变，那么该位一定要置为“1”。
LinkBWMgtStatus	[14]	RW1C,0	链路带宽管理状态。该位由硬件设置，表示以下情况之一发生： （1）将链路控制寄存器的 RetrainLink 位设为“1”之后，链路重新训练完成； （2）硬件改变链路速度或宽度来纠正不可靠的链路运行。如果物理层报告速度或者宽度因下游设备发起
DLLAactive	[13]	RO,0	数据链接层运作（active）。如果实现了链路能力寄存器的 DLLActiveRpt 位，则该位也必须实现。
SlotClkConfig	[12]	HwInit,1	插槽时钟配置。表示该器件使用的是连接器提供的相同物理参考时钟。
LinkTraining	[11]	RO,0	链路训练中。

NegLinkWidth	[9:4]	RO,0x1	协商的链接宽度。在链路初始化之后由硬件自动设置，链路起来（link up）之前该域无意义。
LinkSpeed	[3:0]	RO,0x1	链路速率。在链路初始化之后由硬件自动设置。链路起来（link up）之前该域无意义。编码含义为： 0b0001: Gen 1, 2.5Gbps; 0b0010: Gen 2, 5.0Gbps; 0b0011: Gen 3, 8.0Gbps; 其它：保留。
—	其它	—	保留。

### 5.2.3.3.9 插槽能力寄存器

偏移：0x84。

表5-107：插槽能力寄存器描述

名称	范围	类型	描述
PhySlotNum	[31:19]	HwInit,0x00	物理插槽数目。
NoCmdCplSupport	[18]	HwInit,0	支持无命令完成。该位为“1”表示热插拔控制器完成了一条命令后，插槽不会产生软件提示。
ElecInterlockPresent	[17]	HwInit,0	存在电气互锁。该位为“1”表示该插槽的底板上实现了电气互锁。
SlotPwrLmtScale	[16:15]	HwInit,0x0	插槽功耗限制比例（Slot Power Limit Scale）。该域的编码含义为： 0b00: 1.0x; 0b01: 0.1x; 0b10: 0.01x; 0b11: 0.001x。
SlotPwrLmtValue	[14:7]	HwInit,0x00	插槽功耗限制值，即物理插槽所支持的最大功耗。
HotPlugCap	[6]	HwInit,0	支持热插拔。为“1”表示插槽支持热插拔。
HotPlugSurprise	[5]	HwInit,0	突然热插拔。为“1”表示插槽允许设备在不提示软件的情况下移除。
PwrIndicatorPresent	[4]	HwInit,0	存在电源指示。为“1”表示插槽底板上实现了电源指示。
AttentionIndicatorPresent	[3]	HwInit,0	存在警告指示。为“1”表示插槽底板上实现了警告指示。

MRLSensorPresent	[2]	HwInit,0	存在手动保持门闩（MRL，Manually-operated Retention Latch）传感器指示。为“1”表示插槽底板上实现了 MRL 传感器。
PwrCtrlPresent	[1]	HwInit,0	存在电源控制器。为“1”表示插槽底板上实现了电源管理器。
AttentionButtonPresent	[0]	HwInit,0	存在警告按钮。为“1”表示插槽底板上实现了警告按钮。

### 5.2.3.3.10 插槽控制寄存器

偏移：0x88。

字节：0-1。

表5-108：插槽控制寄存器描述

名称	范围	类型	描述
DLLStateChangeEn	[12]	RW,0	数据链接层状态变化使能。
ElecInterlockCtrl	[11]	RW,0	电气互锁控制。
PwrCtrlrCtrl	[10]	RW,0	电源控制器控制。
PwrIndicatorCtrl	[9:8]	RW,0x3	电源指示器控制。
AttentionIndicatorCtrl	[7:6]	RW,0x3	警告指示器控制。
HPIntEn	[5]	RW,0	热插拔中断使能。
CmdCplIntEn	[4]	RW,0	命令完成中断使能。
PresenceDtctChangeEn	[3]	RW,0	在位检测变化使能。
MRLSensorChangeEn	[2]	RW,0	手动保持门闩传感器变化使能。
PwrFaultDtctEn	[1]	RW,0	电源故障检测使能。
AttentionButtonPressEn	[0]	RW,0	警告按钮按下使能。
—	其它	—	保留。

### 5.2.3.3.11 插槽状态寄存器

偏移：0x88。

字节：2-3。

表 5-109：插槽状态寄存器描述

名称	范围	类型	描述
DLLStateChange	[8]	RW1C,0	数据链接层状态变化。
ElecInterlockStatus	[7]	RO,0	电气互锁状态。
PresenceDtctState	[6]	RO,0	存在检测状态。

MRLSensorState	[5]	RO,0	手动保持门闩传感器状态。
CmdCpl	[4]	RW1C,0	命令完成。
PresenceDtctChange	[3]	RW1C,0	存在检测变化。
MRLSensorChange	[2]	RW1C,0	手动保持门闩传感器变化。
PwrFaultDtct	[1]	RW1C,0	检测到电源故障。
AttentionButtonPress	[0]	RW1C,0	按下警告按钮。
—	其它	—	保留。

### 5.2.3.3.12 根控制寄存器

偏移：0x8C。

字节：0-1。

表 5-110：根控制寄存器描述

名称	范围	类型	描述
CRSSoftVisibilityEn	[4]	RO,0	CRS（Configuration Request Retry Status，即 EP 的配置返回未成功需要重试的响应）软件可视性使能。不支持，硬连线为“0”。
PMEIntEn	[3]	RW,0	电源管理事件中断使能。
SerrFatalErrEn	[2]	RW,0	致命错误上的系统错误使能。
SerrNFatalErrEn	[1]	RW,0	非致命错误上的系统错误使能。
SerrCErrEn	[0]	RW,0	可纠正错误上的系统错误使能。
—	其它	—	保留。

### 5.2.3.3.13 根能力寄存器

偏移：0x8C。

字节：2-3。

表 5-111：根能力寄存器描述

名称	范围	类型	描述
CRSSoftVisibility	[0]	HwInit,0	对 EP 的配置请求未成功时，软件是否可见。不支持，硬连线为“0”。
—	其它	—	保留。

### 5.2.3.3.14 根状态寄存器

偏移：0x90。

表 5-112：根状态寄存器描述

名称	范围	类型	描述
PMEPend	[17]	RO,0	电源管理事件悬挂。
PMESstatus	[16]	RW1C,0	电源管理事件状态。

PMEReqID	[15:0]	RO,0x00	电源管理事件请求 ID。
—	其它	—	保留。

### 5.2.3.3.15 设备能力 2 寄存器

偏移：0x94。

表 5-113：设备能力 2 寄存器描述

名称	范围	类型	描述
CplTimeoutDisSpt	[4]	RO,1	支持禁用响应超时。
CplTimeoutRangeSpt	[3:0]	HWInit,0xF	支持的响应超时范围。编码含义为： 0bxxx1：支持范围 A； 0bxx1x：支持范围 B； 0bx1xx：支持范围 C； 0b1xxx：支持范围 D。
—	其它	—	保留。

### 5.2.3.3.16 设备控制 2 寄存器

偏移：0x98。

表 5-114：设备控制 2 寄存器描述

名称	范围	类型	描述
CplTimeoutDisable	[4]	RW,1	禁用完成超时。
CplTimeoutValue	[3:0]	RW,0x0	完成超时值。编码含义为： 0b0000：50us~50ms，默认范围； 0b0001：50us~100us； 0b0010：1ms~10ms； 0b0101：16ms~55ms； 0b0110：65ms~210ms； 0b1001：260ms~900ms； 0b1010：1s~3.5s； 0b1101：4s~13s； 0b1110： 17s~64s； 其它： 保留
—	其它	—	保留。

### 5.2.3.3.17 链路控制 2 寄存器 (Gen2)

偏移：0xA0。

字节：0-1。

表 5-115：链路控制 2 寄存器描述

名称	范围	类型	描述
----	----	----	----



DeemphLevel	[12]	RWS,0	链路训练状态机处于 Polling.Compliance 状态时的去加重等级 (De-emphasis Level)。编码含义为： 0: -6 dB; 1: -3.5 dB。 当链路工作在 2.5Gbps 时，该位无效。
ComplianceSOS	[11]	RWS,0	兼容性 SKP 有序集 (SOS, SKP Ordered Set)。为“1”表示链路训练状态机需要在兼容性模式中发送 SKP 有序集。 当链路工作在 2.5Gbps 时，该位无效。
EnterMdfyCompliance	[10]	RW,0	进入修改兼容性 (Modified Compliance)。为“1”表示当链路训练状态机进入 Polling.Compliance 状态时该 RC 需要发送兼容性序列。 当链路工作在 2.5GT/s 时，该位无效。
TransimitMargin	[9:7]	RW,0x0	传送差额。该域控制发送端口非去加重电压等级的值

			(value of the non-de-emphasized voltage level)，编码含义为： 0b000: 全摆幅 800~1200 mV，半摆幅 400~600mV; 0b001-0b010: 值必须与非 0 偏斜单调一致; 0b011: 全摆幅 200~400mV，半摆幅 100~200mV; 其它: 保留。 工作在 5.0Gbps 为全摆幅时，去加重率必须维持在规定操作值 (-3.5dB 或-6dB) 的 +/-1dB 之内。
SlctDeemph	[6]	RWS,0	可选择去加重。 链路工作在 5.0Gbps 时，选择去加重的等级： 0: -6 dB; 1: -3.5 dB。 当链路工作在 2.5GT/s 时，该位无效。
HwAutoSpeedDis	[5]	RO,0	禁止硬件自动调整速度。
EnterCompliance	[4]	RWS,0	进入兼容性模式。 软件可以强制链路按照 TrgtLinkSpeed 域指定的速率进入兼容性模式，并将会在链路上产生一个热复位。

TrgtLinkSpeed	[3:0]	RW,0x2	目标链路速率。 通过设置一个工作速率上限，来限制 RC 在链路训练序列（TS，Training Sequence）中的速率通告值。编码含义为： 0b0001：2.5Gbps 的目标链路速率； 0b0010：5Gbps 的目标链路速率； 其它：保留。任何其余的值写入，会导致未知结果。 该域指定了 RC 的最高链接速率值。
—	其它	—	保留。

### 5.2.3.3.18 链路状态 2 寄存器 (Gen2)

偏移：0xA0。

字节：2-3。

表 5-116：链路状态 2 寄存器描述

名称	范围	类型	描述
CurDeemphLevel	[0]	RO,0	当前去加重等级。 当链路工作在 5Gbps 时，该位表示去加重等级，编码含义为： 0：-6 dB；
			1：-3.5 dB。 当链路工作在 2.5Gbps 时，该位无效。
—	其它	—	保留。

## 5.2.4 PCI-E 扩展能力寄存器

申威 1610/410 处理器支持两种 PCI-E 扩展能力结构：

- 1) 高级错误报告能力；
- 2) 虚通道能力。

### 5.2.4.1 高级错误报告能力

#### 5.2.4.1.1 PCI-E 扩展能力头

偏移：0x100。

表5-117：PCI-E 扩展能力头寄存器描述

名称	范围	类型	描述
NextCapOffset	[31:20]	RO,0x140	下一能力偏移，指向虚通道能力结构。

CapVersion	[19:16]	RO,0x1	能力版本。
PCIEExtCapID	[15:0]	RO,0x1	PCI-E 扩展能力 ID。

#### 5.2.4.1.2 不可纠正错状态寄存器

偏移：0x104。

表5-118：不可纠正错寄存器描述

名称	范围	类型	描述
URrErrStatus	[20]	RW1CS,0	不支持请求错误状态。
ECRCrErrStatus	[19]	RW1CS,0	ECRC 错误状态。
MTLPStatus	[18]	RW1CS,0	畸形 TLP (Malformed TLP) 错误状态。
RcvOverflowStatus	[17]	RW1CS,0	接收溢出错误状态。
UnexpctCplStatus	[16]	RW1CS,0	非预期完成错误状态。
CAStatus	[15]	RW1CS,0	响应端异常中止错误状态。
CplTimeoutStatus	[14]	RW1CS,0	响应超时错误状态。
FCPrctlErrStatus	[13]	RW1CS,0	流控制协议错误状态。
PoisonedTLPStatus	[12]	RW1CS,0	中毒 TLP (posioned TLP) 错误状态。
DLPerrStatus	[4]	RW1CS,0	数据链路协议错误状态。
—	其它	—	保留。

#### 5.2.4.1.3 不可纠正错屏蔽寄存器

偏移：0x108。

表5-119：不可纠正错屏蔽寄存器描述

名称	范围	类型	描述
URrErrMask	[20]	RWS,0	不支持请求错误屏蔽。
ECRCrErrMask	[19]	RWS,0	ECRC 错误屏蔽。
MTLPMask	[18]	RWS,0	畸形 TLP 错误屏蔽。
RcvOverflowMask	[17]	RWS,0	接收溢出错误屏蔽。
UnexpctCplMask	[16]	RWS,0	非预期完成错误屏蔽。
CAMask	[15]	RWS,0	响应端异常中止错误屏蔽。
CplTimeoutMask	[14]	RWS,0	响应超时错误屏蔽。
FCPrctlErrMask	[13]	RWS,0	流控制协议错误屏蔽。
PoisonTLPMask	[12]	RWS,0	中毒 TLP 错误屏蔽。
DLPerrMask	[4]	RWS,0	数据链路协议错误屏蔽。
—	其它	—	保留。

#### 5.2.4.1.4 不可纠正错严重性寄存器

偏移：0x10C。

表5-120: 不可纠正错严重性寄存器描述

名称	范围	类型	描述
URrErrMask	[20]	RWS,0	不支持请求错误严重性。
ECRCrErrMask	[19]	RWS,0	ECRC 错误严重性。
MTLPMask	[18]	RWS,1	畸形 TLP 错误严重性。
RcvOverflowMask	[17]	RWS,1	接收溢出错误严重性。
UnexpctCplMask	[16]	RWS,0	非预期完成错误严重性。
CAMask	[15]	RWS,0	响应端异常中止错误严重性。
CplTimeoutMask	[14]	RWS,0	响应超时错误严重性。
FCPrctlErrMask	[13]	RWS,1	流控制协议错误严重性。
PoisonTLPMask	[12]	RWS,0	中毒 TLP 错误严重性。
DLPErrMask	[4]	RWS,1	数据链路协议错误严重性。
—	其它	—	保留。

#### 5.2.4.1.5 可纠正错状态寄存器

偏移: 0x110。

表5-121: 可纠正错状态寄存器描述

名称	范围	类型	描述
NFtatalErrStatus	[13]	RW1CS,0	报告非致命错误状态。
ReplyTimeoutStatus	[12]	RW1CS,0	重发计时器超时状态。
ReplyNumStatus	[8]	RW1CS,0	重发次数翻转状态。
BadDLLPStatus	[7]	RW1CS,0	错误 DLLP (Data Link Layer Packet, 数据链路层数据包) 状态。
BadTLPStatus	[6]	RW1CS,0	错误 TLP (Transaction Layer Packet) 状态。
RcvErrStatus	[0]	RW1CS,0	接收端错误状态。
—	其它	—	保留。

#### 5.2.4.1.6 可纠正错屏蔽寄存器

偏移: 0x114。

表5-122: 可纠正错屏蔽寄存器描述

名称	范围	类型	描述
NFtatalErrMask	[13]	RWS,1	报告非致命错误屏蔽。
ReplyTimeoutMask	[12]	RWS,0	重发计时器超时屏蔽。
ReplyNumMask	[8]	RWS,0	重发次数翻转屏蔽。
BadDLLPMask	[7]	RWS,0	错误 DLLP 屏蔽。
BadTLPMask	[6]	RWS,0	错误 TLP 屏蔽。

RcvErrMask	[0]	RWS,0	接收端错误屏蔽。
—	其它	—	保留。

#### 5.2.4.1.7 高级错误能力及控制寄存器

偏移：0x118。

表5-123：高级错误能力及控制寄存器描述

名称	范围	类型	描述
ECRCCheckEn	[8]	RWS,0	ECRC 检查使能。
ECRCCheckCap	[7]	RO,1	ECRC 检查能力。
ECRCGenEn	[6]	RWS,0	ECRC 生成使能。
ECRCGenCap	[5]	RO,1	ECRC 生成能力。
FirstErrPtr	[4:0]	ROS,0	第一个错误指针。
—	其它	—	保留。

#### 5.2.4.1.8 头记录寄存器

偏移：0x11C~0x128。

头记录寄存器记录检测到错误的 TLP 的包头，并遵从 PCI-E 2.0 标准。头记录寄存器都是 ROS 类型；默认值为 0x00000000。

表 5-124：头记录寄存器描述

偏移	字节 3	字节 2	字节 1	字节 0
0x11C	头记录寄存器(包头第一个 32 位)。			
0x120	头记录寄存器(包头第二个 32 位)。			
0x124	头记录寄存器(包头第三个 32 位)。			
0x128	头记录寄存器(包头第四个 32 位)。			

#### 5.2.4.1.9 根错误命令寄存器

偏移：0x12C。

表 5-125：根错误命令寄存器描述

名称	范围	类型	描述
FatalErrRptEn	[2]	RW,0	致命错误报告使能。
NfatalErrRptEn	[1]	RW,0	非致命错误报告使能。
CerrRptEn	[0]	RW,0	可纠正错报告使能。
—	其它	—	保留。

#### 5.2.4.1.10 根错误状态寄存器

偏移：0x130。

表 5-126：根错误状态寄存器描述

名称	范围	类型	描述
AERIntMsgNum	[31:27]	RO, 0	高级错误报告触发的 MSI 中断所对应的中断向量号。
FatalErrMsgRcv	[6]	RW1CS,0	接收到致命错误消息。
NFatalErrMsgRcv	[5]	RW1CS,0	接收到非致命错消息。
FirstUFatal	[4]	RW1CS,0	首个不可纠致命错。
MultiFatalNFatalErrRcv	[3]	RW1CS,0	接收到多个致命错 / 非致命错消息 (ERR_FATAL/NONFATAL)。
FatalNFatalErrRcv	[2]	RW1CS,0	接收到致命错/非致命错消息 (ERR_FATAL/NONTATAL)。
MultiCErrRcv	[1]	RW1CS,0	接收到多个错误纠正消息 (ERR_COR)。
CErrRcv	[0]	RW1CS,0	接收到错误纠正消息 (ERR_COR)。
—	其它	—	保留。

#### 5.2.4.1.11 错误源标志寄存器

偏移：0x134。

表 5- 127：错误源标志寄存器描述

名称	范围	类型	描述
FatalNFatalSrc	[31:16]	ROS, 0	致命/非致命错消息 (ERR_FATAL/NONTATAL) 源标志。
CErrSrc	[15:0]	ROS,0	可纠正错消息 (ERR_COR) 源标志。

## 5.2.4.2 虚通道能力

### 5.2.4.2.1 虚通道扩展能力包头

偏移：0x140。

表5-128：虚通道扩展能力包头寄存器域的描述

名称	范围	类型	描述
NextCapPtr	[31:20]	RO,0x000	下一能力偏移。
CapVersion	[19:16]	RO,0x1	能力版本。
PCIEExtCapID	[15:0]	RO,0x2	虚通道能力所对应的扩展能力号。

## 5.2.5 端口逻辑寄存器

### 5.2.5.1 应答延迟计时和重发计时寄存器

偏移：0x700。

表 5- 129: 应答延迟计时和重发计时寄存器描述

名称	范围	类型	描述
ReplyTimeLimit	[31:16]	RWS,0x1886	重发延时上限。当重发计时器达到该上限后，就会超时。该上限值的计算取决于工作频率（通常情况下申威 1610/410 的 PCI-E 模块工作在 250MHz）、链路宽度和最大负载（MaxPayloadSize）。
RTLLimit	[15:0]	RWS,0x817	往返延时时间上限（Round Trip Latency Time Limit）。当 Ack/Nak 延迟计时器达到该上限后，就会超时。该上限值的计算取决于工作频率（通常情况下申威 1610/410 的 PCI-E 模块工作在 250MHz）、链路宽度和最大负载（MaxPadloadSize）。

### 5.2.5.2 其它消息寄存器

偏移：0x704。

表5-130：其它消息寄存器描述

名称	范围	类型	描述
OtherMsg	[31:0]	RW,0xFFFF,FFFF	其它消息寄存器。该寄存器的作用是：要发送一个特定的 PCI-E 消息，可向该寄存器写入消息所携带的数据，然后设置端口链路控制寄存器（偏移为 0x710）的 OtherMsgReq 位（即第 0 位），则可发出该消息。

### 5.2.5.3 端口强制链路寄存器

偏移：0x708。

表 5-131：端口强制链路寄存器描述

名称	范围	类型	描述
LowPwrEnterCnt	[31:24]	RWS,0x7	低功耗模式进入等待周期数。 不适用于 RC。
ForceLinkState	[21:16]	RWS,0x00	强制进入的链路状态。编码含义见附录 B。
ForceLinkCmd	[15]	RW,0	强制链路进入指定状态。 配置该域为 1 会生成脉冲信息，而导致链路训练状态机重新进行链路训练。读该域总是返回 0。
LaneNum	[7:0]	RWS,0x08	通道数目。
—	其它	—	保留。



### 5.2.5.4 应答频率寄存器

偏移：0x70C。

表 5- 132：应答频率寄存器描述

名称	范围	类型	描述
EnterL1WithoutL0s	[30]	RWS,0	允许在链路另一端没有进入 L0s 的情况下，即可发起进入 L1 低功耗状态。
L1EntryLatency	[29:27]	RWS,0x3	L1 进入延时。编码含义为： 0b000: 1 μs; 0b001: 2 μs;
			0b010: 4 μs; 0b011: 8 μs; 0b100: 16 μs; 0b101: 32 μs; 其它: 64 μs。
L0sEntryLatency	[26:24]	RWS,0x3	L0s 进入延时。编码含义为： 0b000: 1 μs; 0b001: 2 μs; 0b010: 3 μs; 0b011: 4 μs; 0b100: 5 μs; 0b101: 6 μs; 其它: 7 μs。
ComFTSNum	[23:16]	RWS,0x10	使用共模时钟时，从 L0s 迁移到 L0 状态时需要发送的 FTS 有序集的数目。
FTSNum	[15:8]	RWS,0x10	从 L0s 迁移到 L0 状态时需要发送的 FTS 有序集的数目。
AckFreq	[7:0]	RWS,0	指定 RC 累积的 Ack 消息数量，达到该数量后 RC 会发送一个统一的 Ack 消息。
—	其它	—	保留。

### 5.2.5.5 端口链路控制寄存器

偏移：0x710。

表5- 133：端口链路控制寄存器描述

名称	范围	类型	描述
LinkModeEn	[21:16]	RWS,x0F	链路模式使能。 0x1: x1; 0x3: x2; 0x7: x4; 0xF: x8; 0x1F: x16; 0x3F: x32 (不支持)。
FastLinkMode	[7]	RWS,0	快速链路模式。仅在硬件模拟时使用。
DLLLinkEn	[5]	RWS,1	数据链路层链路使能。设置该位, 则进行流量控制信用初始化、并最终建立链路。
RstAssert	[3]	RWS,0	复位断言。配置该位为“1”则触发一个恢复并强制 LTSSM 进入热复位状态。
LoopbackEn	[2]	RWS,0	打开环回测试。
ScrambleDis	[1]	RWS,0	缠绕禁用。关闭数据缠绕。
OtherMsgReq	[0]	RWS,0	请求发送消息。 配置该位为“1”可以在 PCI-E 链路上发送消息, 消息的内容存放在其它消息寄存器 (偏移: 0x704) 中。

### 5.2.5.6 通道偏斜寄存器

偏移: 0x714。

表 5- 134: 通道偏斜寄存器描述

名称	范围	类型	描述
DisLane2LaneDeskew	[31]	RWS,0	禁用内部通道之间的去偏斜 (lane-to-lane deskew) 逻辑。
AckNakDis	[25]	RWS,0	禁止发送 Ack (Acknowledge, 确认) 和 Nak (Non Acknowledgement) DLLP (Data Link Layer Packet, 数据链路程数据包)。
FCDis	[24]	RWS,0	禁止发送流量控制 DLLP。
InstLaneSkew	[23:0]	RWS,0x000000	在传输时插入通道偏斜。测试时使用。每条通道依次对应 3 位, 且每个通道最多有 5 个符号时间的偏斜。
—	其它	—	保留。

### 5.2.5.7 符号数目寄存器

偏移：0x718。

表 5- 135：符号数目寄存器描述

名称	范围	类型	描述
FuncNum	[31:29]	RWS,0x1	功能数目。功能编号超过该值的配置请求将会被看做 UR。申威 1610/410 只实现了一个功能（function）。
FCTimerMdfy	[28:24]	RWS,0x0	增大流量控制监视计时器（Flow Control Watchdog Timer）数值。该域的单位是 16 个周期。
AckNackTimerMdfy	[23:19]	RWS,0x0	增大 Ack/Nak 延迟计时器的数值，即应答延迟计时和重发计时器（偏移为 0x700）的 RTLLimit 域。该域的单位是 64 个周期。
ReplyTimerMdfy	[18:14]	RWS,0x1	增大重发计时器的数值，即应答延迟计时和重发计时器（偏移为 0x700）的 ReplyTimeLimit 域。该域的单位是 64 个周期。
SKPNum	[10:8]	RWS,0x3	SKP 符号的数目。
NTSS	[3:0]	RWS,0xA	TS 符号的数目，即 TS1/TS2 序列中传输的 TS 标识符号的数目。
—	其它	—	保留。

### 5.2.5.8 符号计时寄存器和过滤屏蔽寄存器 1

偏移：0x71C。

表 5- 136：符号计时寄存器和过滤屏蔽寄存器 1 描述

名称	范围	类型	描述
----	----	----	----

MaskFilter1	[31:16]	RWS,0x0000	<p>使用屏蔽位来选择事务过滤和错误处理的规则。为“0”时，标识使用相关的过滤规则；为“1”时屏蔽相关的过滤规则。各位所对应的规则为：</p> <p>[31]：屏蔽配置请求的过滤；</p> <p>[30]：屏蔽 I/O 请求的过滤；</p> <p>[29]：屏蔽除厂商自定义消息以外的消息 TLP 的过滤；</p> <p>[28]：屏蔽完成事务 ECRC 错误过滤；</p> <p>[27]：屏蔽 ECRC 错误过滤； [26]：屏蔽完成事务的长度不匹配错误； [25]：屏蔽完成事务的属性不匹配错误；</p> <p>[24]：屏蔽完成事务的业务等级（Traffic Class）不匹配错误；</p> <p>[23]：屏蔽完成事务的功能不匹配错误； [22]：屏蔽完成事务的请求者 ID 不匹配错误； [21]：屏蔽完成事务的标签（tag）错误检查规则；</p> <p>[20]：屏蔽锁请求过滤；</p>
			<p>[19]：屏蔽 Type1 类型配置请求过滤；</p> <p>[18]：屏蔽基址寄存器匹配过滤；</p> <p>[17]：屏蔽中毒 TLP 的过滤；</p> <p>[16]：屏蔽请求功能不匹配过滤。</p>
FCTimerDis	[15]	RWS,0	禁用流量控制的监视计时器。
SKPInterval	[10:0]	RWS,1280	SKP 有序集间的传输间隔。
—	其它	—	保留。

### 5.2.5.9 过滤屏蔽寄存器 2

偏移：0x720。

表 5-137：过滤屏蔽寄存器 2 描述

名称	范围	类型	描述
----	----	----	----

MaskFilter2	[31:0]	RWS,0x00000000	<p>使用屏蔽位来选择事务过滤和错误处理的规则。为“0”时，标识使用相关的过滤规则；为“1”时屏蔽相关的过滤规则。各位所对应的规则为： [3]：屏蔽刷新请求（flush request）过滤； [2]：对应不期望的完成 TLP，屏蔽使能 DLLP 异常中止；</p> <p>[1]：屏蔽丢弃 Type1 类型的厂商自定义消息的处理；</p> <p>[0]：屏蔽丢弃 Type0 类型的厂商自定义消息并将其当作 UR 的处理；</p> <p>其它：保留。</p>
-------------	--------	----------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### 5.2.5.10 调试寄存器 0

偏移：0x728。

表 5-138：调试寄存器 0 描述

名称	范围	类型	描述
DebugInfo0	[31:0]	RO,0x00000000	<p>调试信息： [31:28]：链路另一端通告的链路控制位； [27]：协商通道数目时在相应的域为 PAD 空符号； [26]：协商链路数目时在相应的域为 PAD 空符号；</p> <p>[25]：接收方收到逻辑空闲；</p>
			<p>[24]：第 2 个符号依然空闲（只在 Gen2 模式下 PIPE 接口宽度为 16 时有意义）； [23:8]：PIPE 接口传输的数据； [7:6]：PIPE 接口传输的数据是否为控制字符的指示； [5]：发送了 SKP 有序集；</p> <p>[4:0]：LTSSM 当前的状态。</p>

### 5.2.5.11 调试寄存器 1

偏移：0x72C。

表5-139：调试寄存器1描述

名称	范围	类型	描述
DebugInfo1	[31:0]	RO,0x00000000	调试信息： [31]：链路禁止数据缠绕； [30]：链路训练状态处于禁用（Disable）状态，链路不可用； [29]：链路正在训练中； [28]：链路正在测试极性颠倒（polarity reversal）； [27]：链路需要复位； [22]：PIPE 接口的接收器检测、或环回测试请求； [21]：PIPE 发出电气空闲请求； [20]：PIPE 发出兼容性测试（compliance）请求； [19]：应用申请复位链路； [15:8]：链路接收方通告/确认的通道数； [4]：PMA 报告链路成功（link up）； [3]：接收器报告收到 SKP 符号； [2]：接收器报告收到 TS1 序列（脉冲）； [1]：接收器报告收到 TS2 序列（脉冲）； [0]：接收器检测到通道反向（lane reversal）； 其它：保留。

### 5.2.5.12 传输 P 类事务流量控制信用状态寄存器

偏移：0x730。

表5-140：传输P类事务流量控制信用状态寄存器描述

名称	范围	类型	描述
PHdrFCCrdt	[19:12]	ROS,0x00	链路另一端接收器所通告的 P 类事务包头信用。初始值为 0，在流量控制初始化后根据接收缓冲的规模进行设置。
PDataFCCrdt	[11:0]	ROS,0x000	链路另一端接收器所通告的 P 类事务数据信用。初始值为 0，在流量控制初始化后根据接收缓冲的规模进行设置。
—	其它	—	保留。

### 5.2.5.13 传输 NP 类事务流量控制信用状态寄存器

偏移：0x734。

表5-141：传输NP类事务流量控制信用状态寄存器描述

名称	范围	类型	描述
NPHdrFCCrdt	[19:12]	ROS,0x00	链路另一端接收器所通告的 NP 类事务包头信用。初始值为 0，在流量控制初始化后根据接收缓冲的规模进行设置。
NPDataFCCrdt	[11:0]	ROS,0x000	链路另一端接收器所通告的 NP 类事务数据信用。初始值为 0，在流量控制初始化后根据接收缓冲的规模进行设置。
—	其它	—	保留。

### 5.2.5.14 传输完成事务流量控制信用状态寄存器

偏移：0x738。

表5-142：传输完成事务流量控制信用状态寄存器描述

名称	范围	类型	描述
CplHdrFCCrdt	[19:12]	ROS,0x00	链路另一端接收器所通告的完成类事务包头信用。初始值为 0，在流量控制初始化后根据接收缓冲的规模进行设置。
CplDataFCCrdt	[11:0]	ROS,0x000	链路另一端接收器所通告的完成类事务数据信用。初始值为 0，在流量控制初始化后根据接收缓冲的规模进行设置。
—	其它	—	保留。

### 5.2.5.15 队列状态寄存器

偏移：0x73C。

表5-143：队列状态寄存器描述

名称	范围	类型	描述
RcvQNotEmpty	[2]	ROS,0	接收队列非空。
TransRBufNotEmpty	[1]	ROS,0	传输重发缓冲非空。
RcvFCCrdtNotRtrn	[0]	ROS,0	事务层数据包信用尚未返回。
—	其它	—	保留。

### 5.2.5.16 Gen2 模式寄存器

偏移：0x80C。

表5-144：Gen2 模式寄存器描述

名称	范围	类型	描述
TxCompRcv	[19]	RW,0	配置发送器兼容性测试接收位。设置为“1”时，通知 LTSSM 发送 TS 序列。
TxSwing	[18]	RW,0	配置 PMA 发送电压摆幅。编码含义为： 0：低摆幅； 1：全摆幅。
DrctSpeedChange	[17]	RW,0	指导进行速率改变。指示 LTSSM 是否要改变速率。
LaneEn	[16:8]	RW,0x00	通道使能。指示由 Polling.Active 和 Polling.Compliance（LTSSM 的状态，详见附录 B）的电气空闲状态退出时要检查的通道数目。该域指定了通道的最大数目，可用来忽略损坏的通道。而如果有损坏的通道未被忽略，则不能退出
FTSNum	[7:0]	RW,0x10	设置 FTS（Fast Training Sequence，快速训练序列）的数目。在 Gen2 链路训练时，RC 需要通告其 FTS 的数目，其作用是把链路恢复同步的能力告知链路的另一方。
—	其它	—	保留。



## 附录 A PCI-E 术语表

表 A-1: PCI-E 标准术语的缩写、英文全称和中文含义对照表

缩写	英文	含义
PCI-E	Peripheral Component Interconnection Express	设备接口协议增强版。
DM	PCI Express Dual Mode core	双模式 (RC/EP) 核心。
RC	PCI Express Root Complex core	根设备核心。
EP	PCI Express End Point core	端设备核心。
PIPE	PHY Interface for the PCI Express Architecture	PCI-E 结构中的 PHY 接口。
NW	double words	32 位。
DW	Data width	数据宽度, 可为: 32、64、128 位。
NF	Number of functions	功能设备数。
TLP	Transaction Layer Packet	事务层数据包。
DLLP	Data Link Layer Packet	数据链路层数据包。
VC	Virtual Channel	虚通道。
BAR	Base Address Register	基址寄存器。
—	Upstream	上游设备, 在系统中指 CPU。
—	Downstream	下游设备, 在系统中指 EP 设备。
MSI	Message Signalled Interrupt	中断消息。
TC	Traffic Classes	业务级别。
—	Sticky	粘着属性, 即指仅有上电复位才能复位该寄存器。
—	Non_ Sticky	非粘着属性, 即指所有复位都可以复位该寄存器。
PME	Power Management Event	电源管理事件。
—	Payload	请求携带的数据。
DBI	External Data Bus Interface	数据总线接口, 即用来配置读写 RC 内部寄存器的接口。
LTSSM	Link Training and Status State Machine	数据链路状态机。
CRS	Completion Request Retry Status	对 EP 的配置请求未成功, 需要重发。
UR	Unsupported Request	请求是 EP 不支持的类型。

CA	Completion Abort	EP 此时故障无法处理该笔事务。
—	Non-Posted	非邮寄事务，即需要返回响应的事务。
—	Posted	邮寄事务，即不需要返回响应的事务。
ASPM	Active State Power Management	有效状态电源管理策略。
FC	Flow Control	流量控制。

## 附录 B PCI-E 链路状态编码

申威 411 处理器的 PCI-E 链路状态机遵从 PCI-E2.0 规范，并自定义了部分状态以方便内部处理。

LTSSM 的状态编码及含义具体如下：

表 B-1：PCI-E 标准术语的缩写、英文全称和中文含义对照表

编码	PCI-E 2.0 规范的对应状态	含义
0x00	Detect.Quiet	检测状态进入。
0x01	Detect.Active	检测状态工作期间。
0x02	Polling.Active	查询状态工作期间。
0x03	Polling.Compliance	查询状态兼容性测试。
0x04	Polling.Configuration	查询状态配置。
0x05	PreDetect.Quiet (申威 411 处理器自定义)	预检测状态。
0x06	Detect.Wait (申威 411 处理器自定义)	检测状态等待。
0x07	Configuration.Linkwidth.Start	配置状态开始进行链路宽度协商。
0x08	Configuration.Linkwidth.Accept	配置状态完成链路宽度协商。
0x09	Configuration.Lanenum.Wait	配置状态进行通道编号协商。
0x0A	Configuration.Lanenum.Accept	配置状态完成通道编号协商。
0x0B	Configuration.Complete	配置状态完成。
0x0C	Configuration.Idle	配置状态保持。
0x0D	Recovery.RcvrLock	恢复状态进行接收器同步 (符号锁定、比特锁定)。
0x0E	Recovery.Speed	恢复状态进行链路速率协商。
0x0F	Recovery.RcvrCfg	恢复状态进行接收器配置。
0x10	Recovery.Idle	恢复状态保持。
0x11	L0	链路训练成功，正常工作状态。
0x12	L0s	L0s 低功耗状态退出。
0x13	L123_Send_EIDLE (申威 411 处理器自定义)	L1/L2/L3 低功耗状态下发送电气空闲 (Electrical Idle)。
0x14	L1.Idle	L1 低功耗保持状态。
0x15	L2.Idle	L2 低功耗保持状态。

0x16	L2.TransmitWake	L2 低功耗唤醒状态。
0x17	Disabled.Entry	禁用状态进入。
0x18	Disabled.Idle	禁用保持状态。
0x19	Disabled	禁用状态。此时链路已关闭 DC 共模电压。
0x1A	Lookback.Entry	环回测试状态进入。
0x1B	Loopback.Active	环回测试状态工作。
0x1C	Loopback.Exit	环回测试状态退出。
	Loopback.ExitTimeout (申威 411 处	环回测试退出超时。
0x1E	HotReset.Entry	热复位状态进入。
0x1F	HotReset	热复位状态。