

**LOONGSON**

**龙芯 7A2000 桥片**

**用户手册**

**V1.0**

2022 年 7 月

龙芯中科技股份有限公司

自主决定命运, 创新成就未来

北京市海淀区温泉镇中关村环保科技示范园龙芯产业园2号楼 100095  
Loongson Industrial Park, building 2, Zhongguancun environmental protection park  
Haidian District, Beijing



[www.loongson.cn](http://www.loongson.cn)

## 版权声明

本档版权归龙芯中科技术股份有限公司所有，并保留一切权利。未经书面许可，任何公司和个人不得将此档中的任何部分公开、转载或以其他方式散发给第三方。否则，必将追究其法律责任。

## 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。

## 龙芯中科技术股份有限公司

Loongson Technology Corporation Limited

地址：北京市海淀区中关村环保科技示范园龙芯产业园 2 号楼

Building No.2, Loongson Industrial Park,

Zhongguancun Environmental Protection Park, Haidian District, Beijing

电话(Tel): 010-62546668

传真(Fax): 010-62600826

## 阅读指南

《龙芯 7A2000 桥片用户手册》主要介绍桥片架构与寄存器描述，对桥片系统架构、主要模块的功能与配置、寄存器列表及位域进行详细说明。

## 版本信息

版本信息	文档名	龙芯 7A2000 桥片用户手册
	版本号	V1.0
	创建人	芯片研发部
更新历史		
序号.	版本号	更新内容
1	V1.0	第一版



## 目 录

阅读指南 .....	III
版本信息 .....	V
1 引言 .....	1
1.1 文档说明.....	1
1.1.1 文档约定.....	1
1.2 桥片概述.....	1
1.3 桥片主要功能.....	2
1.3.1 HT 接口 .....	2
1.3.2 PCIE 接口.....	2
1.3.3 GPU.....	3
1.3.4 显示接口.....	3
1.3.5 显存接口.....	3
1.3.6 USB 接口.....	3
1.3.7 SATA 接口.....	4
1.3.8 网络接口.....	4
1.3.9 HDA 接口.....	4
1.3.10 I2S 接口 .....	4
1.3.11 SPI 接口.....	4
1.3.12 LPC 接口 .....	5
1.3.13 UART 接口.....	5
1.3.14 CAN.....	5
1.3.15 I2C 接口.....	5
1.3.16 PWM.....	6
1.3.17 HPET.....	6
1.3.18 RTC.....	6
1.3.19 ACPI 接口 .....	6
1.3.20 GPIO 接口 .....	6
1.3.21 JTAG 接口 .....	6
1.4 桥片工作模式.....	7
1.4.1 HT 主桥模式 .....	7
2 桥片时钟结构.....	8
2.1 桥片时钟.....	8
2.2 时钟功能描述.....	9
2.2.1 系统参考时钟.....	9
2.2.2 内部控制器时钟.....	9
2.2.3 RTC 时钟 .....	11

2.2.4	HT PHY 参考时钟 .....	11
2.2.5	PCIE PHY 参考时钟 .....	12
2.2.6	USB PHY 参考时钟 .....	12
2.2.7	SATA PHY 参考时钟 .....	12
2.2.8	GMAC PHY 参考时钟 .....	12
2.2.9	PHY 时钟之间关系 .....	13
3	地址空间 .....	14
3.1	处理器的访问 .....	14
3.1.1	PCI 设备和功能 .....	14
3.1.2	PCI 配置访问地址 .....	16
3.1.3	桥片设备地址空间分配示例 .....	16
3.2	DMA 的访问 .....	18
3.3	龙芯 3A+7A 系统地址空间分配示例 .....	19
4	桥片配置寄存器 .....	20
4.1	通用配置寄存器 0 .....	20
4.2	通用配置寄存器 1 .....	23
4.3	引脚复用配置寄存器 0 .....	25
4.4	引脚复用配置寄存器 1 .....	28
4.5	固定地址配置寄存器 .....	28
4.6	PLL0 配置寄存器 .....	29
4.7	PLL1 配置寄存器 .....	30
4.8	PLL2 配置寄存器 .....	30
4.9	PLL_PIX_0 配置寄存器 .....	31
4.10	PLL_PIX_1 配置寄存器 .....	31
4.11	SSCPLL 配置寄存器 0 .....	32
4.12	SSCPLL 配置寄存器 1 .....	33
4.13	FREQSCALE 配置寄存器 .....	33
4.14	PCIE 配置访问路由控制寄存器 .....	34
4.15	GMAC0 配置寄存器 .....	34
4.16	桥片标识寄存器 .....	35
5	中断控制器 .....	36
5.1	中断源分配 .....	37
5.2	中断相关寄存器描述 .....	38
5.3	设备中断类型 .....	49
5.4	中断处理过程详细说明 .....	50
6	温度传感器 .....	52
6.1	温度传感器配置寄存器 .....	52
6.2	温度传感器中断控制寄存器 .....	52

6.3	温度传感器中断状态/清除寄存器 .....	53
7	HT 控制器 .....	54
7.1	HT 使用说明 .....	54
7.2	HT 配置寄存器 .....	54
7.2.1	Bridge Control .....	58
7.2.2	Capability Registers .....	58
7.2.3	Error Retry 控制寄存器 .....	60
7.2.4	Retry Count 寄存器 .....	61
7.2.5	Revision ID 寄存器 .....	61
7.2.6	Interrupt Discovery & Configuration.....	61
7.2.7	中断向量寄存器.....	62
7.2.8	中断使能寄存器.....	65
7.2.9	Link Train 寄存器.....	67
7.2.10	接收地址窗口配置寄存器.....	68
7.2.11	配置空间转换寄存器.....	71
7.2.12	POST 地址窗口配置寄存器.....	72
7.2.13	可预取地址窗口配置寄存器.....	73
7.2.14	UNCACHE 地址窗口配置寄存器 .....	74
7.2.15	P2P 地址窗口配置寄存器 .....	77
7.2.16	控制器参数配置寄存器.....	78
7.2.17	接收诊断寄存器.....	81
7.2.18	PHY 状态寄存器.....	81
7.2.19	软件频率配置寄存器.....	81
7.2.20	PHY 阻抗匹配控制寄存器.....	82
7.2.21	PHY 配置寄存器.....	83
7.2.22	链路初始化调试寄存器.....	84
7.2.23	LDT 调试寄存器 .....	84
7.2.24	HT TX POST ID 窗口配置寄存器 .....	86
7.2.25	外部中断转换配置.....	87
7.2.26	扩展地址寄存器.....	87
7.2.27	Unitid 控制寄存器 .....	88
8	MISC 低速设备 (D2:F0) .....	90
8.1	MISC 低速设备配置寄存器 (MISC-D2:F0) .....	90
8.2	内部设备地址路由.....	91
9	UART 控制器 .....	92
9.1	访问地址.....	92
9.2	寄存器描述.....	92
10	CAN 控制器.....	97

10.1	访问地址及引脚复用.....	97
10.2	标准模式.....	97
10.2.1	控制寄存器 (CR) .....	98
10.2.2	命令寄存器 (CMR) .....	99
10.2.3	状态寄存器 (SR) .....	100
10.2.4	中断寄存器 (IR) .....	100
10.2.5	验收代码寄存器 (ACR) .....	100
10.2.6	验收屏蔽寄存器 (AMR) .....	101
10.2.7	发送缓冲区列表.....	101
10.2.8	接收缓冲区列表.....	102
10.3	扩展模式.....	102
10.3.1	模式寄存器 (MOD) .....	104
10.3.2	命令寄存器 (CMR) .....	105
10.3.3	状态寄存器 (SR) .....	105
10.3.4	中断寄存器 (IR) .....	106
10.3.5	中断使能寄存器 (IER) .....	106
10.3.6	仲裁丢失捕捉寄存器.....	107
10.3.7	错误警报限制寄存器 (EMLR) .....	108
10.3.8	RX 错误计数寄存器 (RXERR) .....	108
10.3.9	TX 错误计数寄存器 (TXERR) .....	108
10.3.10	验收滤波器.....	109
10.3.11	RX 信息计数寄存器 (RMCR) .....	109
10.4	公共寄存器.....	109
10.4.1	总线定时寄存器 0 (BTR0) .....	109
10.4.2	总线定时寄存器 1 (BTR1) .....	110
10.4.3	输出控制寄存器 (OCR) .....	110
11	I2C 控制器 .....	111
11.1	访问地址及引脚复用.....	111
11.2	I2C 控制器寄存器说明.....	111
12	PWM 控制器.....	114
12.1	访问地址及引脚复用.....	114
12.2	寄存器描述.....	114
12.3	功能说明.....	115
12.3.1	脉宽调制功能.....	115
12.3.2	脉冲测量功能.....	116
12.3.3	防死区功能.....	116
13	HPET 控制器 .....	117
13.1	访问地址.....	117

13.2	寄存器描述.....	117
14	电源管理模块（ACPI） .....	121
14.1	访问地址.....	121
14.2	电源级别.....	121
14.3	寄存器描述.....	121
15	RTC.....	129
15.1	访问地址.....	129
15.2	寄存器地址列表.....	130
15.3	寄存器描述.....	130
15.3.1	SYS_TOYWRITE0 .....	130
15.3.2	SYS_TOYWRITE1 .....	130
15.3.3	SYS_TOYREAD0 .....	130
15.3.4	SYS_TOYREAD1 .....	131
15.3.5	SYS_TOYMATCH0/1/2.....	131
15.3.6	SYS_RTCCTRL .....	131
15.3.7	SYS_RTCWRITE .....	132
15.3.8	SYS_RTCREAD.....	132
15.3.9	SYS_RTCMATCH0/1/2 .....	132
16	GPIO.....	133
16.1	访问地址.....	133
16.2	控制寄存器.....	134
17	GMAC 控制器（D3:F0） .....	138
17.1	GMAC 配置寄存器（D3:F0） .....	138
18	USB 控制器（D4-5:F0/1, D25:F0） .....	139
18.1	EHCI 控制器 .....	139
18.1.1	EHCI 配置寄存器（D4:F1, D5:F1） .....	139
18.1.2	Capability 寄存器 .....	140
18.1.3	Operational 寄存器.....	140
18.2	OHCI 控制器 .....	141
18.2.1	OHCI 配置寄存器（D4:F0, D5:F0） .....	141
18.2.2	Operational 寄存器.....	142
18.3	XHCI 控制器 .....	142
18.3.1	XHCI 配置寄存器（D25:F0） .....	142
19	图形处理器（D6:F0） .....	144
19.1	GPU 配置寄存器（D6:F0） .....	144
19.2	DDR4 显存接口 .....	144
19.3	DDR4 SDRAM 控制器功能概述.....	145
19.4	DDR4 SDRAM 参数配置格式.....	145

20	显示控制器 (D6:F1)	156
20.1	DC 配置寄存器 (D6:F1)	156
20.2	DC 控制寄存器	157
20.2.1	帧缓冲配置寄存器	157
20.2.2	帧缓冲地址寄存器 0	157
20.2.3	帧缓冲地址寄存器 1	158
20.2.4	Meta0 低地址寄存器	158
20.2.5	Meta0 高地址寄存器	158
20.2.6	Meta1 低地址寄存器	158
20.2.7	Meta1 高地址寄存器	159
20.2.8	帧缓冲跨度寄存器	159
20.2.9	帧缓冲初始字节寄存器	159
20.2.10	颜色抖动配置寄存器	159
20.2.11	颜色抖动查找表低位寄存器	159
20.2.12	颜色抖动查找表高位寄存器	160
20.2.13	液晶面板配置寄存器	160
20.2.14	水平显示宽度寄存器	160
20.2.15	行同步配置寄存器	161
20.2.16	垂直显示高度寄存器	161
20.2.17	场同步配置寄存器	161
20.2.18	行场同步偏移配置寄存器	162
20.2.19	当前显示位置寄存器	162
20.2.20	伽马校正目录寄存器	162
20.2.21	伽马校正值寄存器	162
20.2.22	光标 0 配置寄存器	163
20.2.23	光标 0 存储地址寄存器	163
20.2.24	光标 0 显示位置寄存器	163
20.2.25	单色光标 0 背景色寄存器	164
20.2.26	单色光标 0 前景色寄存器	164
20.2.27	光标 1 配置寄存器	164
20.2.28	光标 1 存储地址寄存器	164
20.2.29	光标 1 显示位置寄存器	165
20.2.30	单色光标 1 背景色寄存器	165
20.2.31	单色光标 1 前景色寄存器	165
20.2.32	HDMI 区域配置寄存器	166
20.2.33	HDMI 控制寄存器	166
20.2.34	Audio BUF 配置寄存器	167
20.2.35	Audio N 配置寄存器	167

20.2.36	Audio CTS 配置寄存器 .....	167
20.2.37	Audio CTS Cal 配置寄存器 .....	167
20.2.38	Audio InfoFrame 配置寄存器 .....	168
20.2.39	Audio Sample 配置寄存器 .....	168
20.2.40	HDMI PHY 控制寄存器 .....	168
20.2.41	HDMI PHY PLL 配置寄存器 .....	169
20.2.42	HDMI PHY PEC0 寄存器 .....	169
20.2.43	HDMI PHY PEC1 寄存器 .....	170
20.2.44	HDMI PHY PEC2 寄存器 .....	170
20.2.45	AVI InfoFrame 内容寄存器 0 .....	171
20.2.46	AVI InfoFrame 内容寄存器 1 .....	171
20.2.47	AVI InfoFrame 内容寄存器 2 .....	171
20.2.48	AVI InfoFrame 内容寄存器 3 .....	172
20.2.49	AVI InfoFrame 控制寄存器 .....	172
20.2.50	Vendor Specific InfoFrame 配置寄存器 .....	172
20.2.51	场同步计数寄存器 .....	173
20.2.52	PLL_PIX 配置寄存器 0 .....	173
20.2.53	PLL_PIX 配置寄存器 1 .....	173
20.2.54	VGA 热插拔控制寄存器 .....	174
20.2.55	HDMI 热插拔状态寄存器 .....	174
20.2.56	HDMI 左声道状态寄存器 .....	174
20.2.57	HDMI 左声道用户数据寄存器 .....	175
20.2.58	HDMI 右声道状态寄存器 .....	175
20.2.59	HDMI 右声道用户数据寄存器 .....	175
20.2.60	中断寄存器 .....	176
21	HDA 控制器 (D7:F0) .....	178
21.1	HDA 配置寄存器 (D7:F0) .....	178
21.2	HDA 控制寄存器描述 .....	179
22	I2S 控制器 (D7:F0) .....	180
22.1	I2S 配置寄存器 (D7:F0) .....	180
22.2	I2S 控制器寄存器 .....	181
22.3	配置操作 .....	182
22.4	DMA 控制器 .....	183
22.4.1	DMA 控制器结构描述 .....	183
22.4.2	DMA 描述符 .....	183
23	SATA 控制器 (D8:F0) .....	187
23.1	SATA 配置寄存器 (D8:F0) .....	187
23.2	SATA 控制寄存器描述 .....	188

24	PCIE 控制器 (D9-16,19-20:F0) .....	190
24.1	PCI 配置寄存器 .....	190
24.2	地址空间划分 .....	192
25	SPI 控制器 (D22:F0) .....	194
25.1	SPI 配置寄存器 (D22:F0) .....	194
25.2	SPI 控制寄存器 .....	195
25.3	SPI 软件编程指南 .....	199
26	LPC 控制器 (D23:F0) .....	202
26.1	LPC 配置寄存器 (D23:F0) .....	202
26.2	LPC 地址空间 .....	204
26.3	LPC 中断 .....	204
26.4	LPC 控制寄存器 .....	204



## 图目录

图 1-1 7A2000 顶层结构图.....	1
图 1-2 7A2000 单路应用示意图 (HT 主桥).....	7
图 2-1 7A2000 的时钟结构.....	8
图 2-2 PLL 结构图.....	10
图 3-1 配置访问地址示意图.....	16
图 3-2 配置访问地址示意图.....	16
图 5-1 3A+7A 的中断系统示意图.....	36
图 12-1 防死区功能.....	116

## 表目录

表 2-1 桥片时钟输入.....	8
表 2-2 桥片时钟输出.....	9
表 2-3 PLL 相关配置信号说明表.....	10
表 2-4 时钟使用规则.....	12
表 2-5 PHY 时钟之间关系.....	13
表 3-1 各个设备的配置头访问对应关系.....	14
表 3-2 桥片固定地址设备地址空间.....	17
表 3-3 桥片 PCI 设备地址空间描述.....	17
表 3-4 DC 访问地址窗口命中和转换规则.....	18
表 3-5 GPU 访问地址窗口命中和转换规则.....	18
表 3-6 龙芯 3 号处理器+龙芯 7A 桥片地址空间划分示例.....	19
表 4-1 桥片配置寄存器列表.....	20
表 4-2 通用配置寄存器 0.....	20
表 4-3 通用配置寄存器 1.....	23
表 4-4 引脚复用配置寄存器 0.....	26
表 4-5 引脚复用配置寄存器 1.....	28
表 4-6 固定地址配置寄存器.....	29
表 4-7 PLL0 配置寄存器.....	29
表 4-8 PLL1 配置寄存器.....	30
表 4-9 PLL2 配置寄存器.....	30
表 4-10 PLL_PIX_0 配置寄存器列表.....	31
表 4-11 PLL_PIX_1 配置寄存器.....	32
表 4-12 SSCPLL 配置寄存器 0.....	32
表 4-13 SSCPLL 配置寄存器 1.....	33
表 4-14 FREQSCALE 配置寄存器.....	34
表 4-15 配置位与 PCIE 端口对应关系.....	34
表 4-16 GMAC0 配置寄存器.....	35
表 4-17 桥片标识寄存器.....	35
表 5-1 桥片中断控制器中断源分配.....	37
表 5-2 中断相关寄存器.....	38
表 5-3 中断寄存器地址分布.....	39
表 6-1 温度传感器配置寄存器.....	52
表 6-2 温度传感器中断控制寄存器.....	52

表 6-3 温度传感器中断状态/清除寄存器.....	53
表 7-1 HT 配置寄存器.....	54
表 7-2 Bus Reset Control 寄存器定义.....	58
表 7-3 Command, Capabilities Pointer, Capability ID 寄存器定义.....	58
表 7-4 Link Config, Link Control 寄存器定义.....	59
表 7-5 Revision ID, Link Freq, Link Error, Link Freq Cap 寄存器定义.....	60
表 7-6 Feature Capability 寄存器定义.....	60
表 7-7 Error Retry 控制寄存器.....	60
表 7-8 Retry Count 寄存器.....	61
表 7-9 Revision ID 寄存器.....	61
表 7-10 Interrupt Capability 寄存器定义.....	61
表 7-11 Dataport 寄存器定义.....	62
表 7-12 IntrInfo 寄存器定义 (1) .....	62
表 7-13 IntrInfo 寄存器定义 (2) .....	62
表 7-14 中断向量映射方式.....	63
表 7-15 HT 总线中断向量寄存器 1.....	63
表 7-16 HT 总线中断向量寄存器 2.....	64
表 7-17 HT 总线中断向量寄存器 3.....	64
表 7-18 HT 总线中断向量寄存器 4.....	64
表 7-19 HT 总线中断向量寄存器 5.....	64
表 7-20 HT 总线中断向量寄存器 6.....	64
表 7-21 HT 总线中断向量寄存器 7.....	65
表 7-22 HT 总线中断向量寄存器 8.....	65
表 7-23 HT 总线中断使能寄存器 1.....	66
表 7-24 HT 总线中断使能寄存器 2.....	66
表 7-25 HT 总线中断使能寄存器 3.....	66
表 7-26 HT 总线中断使能寄存器 4.....	66
表 7-27 HT 总线中断使能寄存器 5.....	66
表 7-28 HT 总线中断使能寄存器 6.....	67
表 7-29 HT 总线中断使能寄存器 7.....	67
表 7-30 HT 总线中断使能寄存器 8.....	67
表 7-31 Link Train 寄存器.....	67
表 7-32 HT 总线接收地址窗口 0 使能 (外部访问) 寄存器定义.....	68
表 7-33 HT 总线接收地址窗口 0 基址 (外部访问) 寄存器定义.....	69

表 7-34 HT 总线接收地址窗口 1 使能（外部访问）寄存器定义.....	69
表 7-35 HT 总线接收地址窗口 1 基址（外部访问）寄存器定义.....	69
表 7-36 HT 总线接收地址窗口 2 使能（外部访问）寄存器定义.....	70
表 7-37 HT 总线接收地址窗口 2 基址（外部访问）寄存器定义.....	70
表 7-38 HT 总线接收地址窗口 3 使能（外部访问）寄存器定义.....	70
表 7-39 HT 总线接收地址窗口 3 基址（外部访问）寄存器定义.....	71
表 7-40 HT 总线接收地址窗口 4 使能（外部访问）寄存器定义.....	71
表 7-41 HT 总线接收地址窗口 4 基址（外部访问）寄存器定义.....	71
表 7-42 配置空间扩展地址转换寄存器定义.....	72
表 7-43 扩展地址转换寄存器定义.....	72
表 7-44 HT 总线 POST 地址窗口 0 使能（内部访问）.....	72
表 7-45 HT 总线 POST 地址窗口 0 基址（内部访问）.....	73
表 7-46 HT 总线 POST 地址窗口 1 使能（内部访问）.....	73
表 7-47 HT 总线 POST 地址窗口 1 基址（内部访问）.....	73
表 7-48 HT 总线可预取地址窗口 0 使能（内部访问）.....	74
表 7-49 HT 总线可预取地址窗口 0 基址（内部访问）.....	74
表 7-50 HT 总线可预取地址窗口 1 使能（内部访问）.....	74
表 7-51 HT 总线可预取地址窗口 1 基址（内部访问）.....	74
表 7-52 HT 总线 Uncache 地址窗口 0 使能（内部访问）.....	75
表 7-53 HT 总线 Uncache 地址窗口 0 基址（内部访问）.....	75
表 7-54 HT 总线 Uncache 地址窗口 1 使能（内部访问）.....	75
表 7-55 HT 总线 Uncache 地址窗口 1 基址（内部访问）.....	76
表 7-56 HT 总线 Uncache 地址窗口 2 使能（内部访问）.....	76
表 7-57 HT 总线 Uncache 地址窗口 2 基址（内部访问）.....	76
表 7-58 HT 总线 Uncache 地址窗口 3 使能（内部访问）.....	77
表 7-59 HT 总线 Uncache 地址窗口 3 基址（内部访问）.....	77
表 7-60 HT 总线 P2P 地址窗口 0 使能（外部访问）寄存器.....	77
表 7-61 HT 总线 P2P 地址窗口 0 基址（外部访问）寄存器.....	78
表 7-62 HT 总线 P2P 地址窗口 1 使能（外部访问）寄存器.....	78
表 7-63 HT 总线 P2P 地址窗口 1 基址（外部访问）寄存器.....	78
表 7-64 控制器参数配置寄存器 0.....	78
表 7-65 控制器参数配置寄存器 1.....	79
表 7-66 接收诊断寄存器.....	81
表 7-67 PHY 状态寄存器.....	81

表 7-68 软件频率配置寄存器.....	82
表 7-69 阻抗匹配控制寄存器.....	83
表 7-70 PHY 配置寄存器.....	83
表 7-71 链路初始化调试寄存器.....	84
表 7-72 LDT 调试寄存器 1.....	84
表 7-73 LDT 调试寄存器 2.....	84
表 7-74 LDT 调试寄存器 3.....	85
表 7-75 LDT 调试寄存器 4.....	85
表 7-76 LDT 调试寄存器 5.....	85
表 7-77 LDT 调试寄存器 5.....	85
表 7-78 post 请求响应完成计数寄存器.....	85
表 7-79 HT TX POST ID WIN0.....	86
表 7-80 HT TX POST ID WIN1.....	86
表 7-81 HT TX POST ID WIN2.....	86
表 7-82 HT TX POST ID WIN3.....	87
表 7-83 HT RX INT TRANS LO.....	87
表 7-84 HT RX INT TRANS Hi.....	87
表 7-85 扩展地址寄存器.....	87
表 7-86 unitid 控制寄存器 0.....	88
表 7-87 unitid 控制寄存器 1.....	88
表 7-88 unitid 控制寄存器 2.....	88
表 7-89 unitid 控制寄存器 3.....	89
表 8-1 MISC 低速设备块的 PCI 配置头 (MISC-D2:F0) .....	90
表 8-2 MISC 低速设备地址路由.....	91
表 8-3 MISC 低速设备地址路由 (续) .....	91
表 10-1 CAN 功能复用.....	97
表 10-2 CAN 模块物理地址构成.....	97
表 10-3 CAN 控制器标准模式下寄存器定义.....	98
表 10-4 CAN 控制器标准模式下的控制寄存器格式.....	99
表 10-5 CAN 控制器标准模式下命令寄存器格式.....	99
表 10-6 CAN 控制器标准模式下状态寄存器格式.....	100
表 10-7 CAN 控制器标准模式下中断寄存器格式.....	100
表 10-8 CAN 验收代码寄存器.....	101
表 10-9 CAN 验收屏蔽寄存器.....	101

表 10-10 CAN 控制器标准模式下发送缓冲区格式.....	101
表 10-11 扩展模式下 CAN 控制器的地址列表.....	102
表 10-12 CAN 控制器扩展模式下的模式寄存器格式.....	105
表 10-13 CAN 控制器扩展模式下命令寄存器格式.....	105
表 10-14 CAN 控制器扩展模式下状态寄存器格式.....	105
表 10-15 CAN 控制器扩展模式下中断寄存器格式.....	106
表 10-16 CAN 控制器扩展模式下中断使能寄存器格式.....	106
表 10-17 CAN 控制器扩展模式下仲裁丢失捕捉寄存器格式.....	107
表 10-18 CAN 错误劲爆限制寄存器.....	108
表 10-19 CAN 的 RX 错误计数寄存器.....	108
表 10-20 CAN 的 TX 错误计数寄存器.....	109
表 10-21 CAN 的 RX 错信息计数寄存器.....	109
表 10-22 CAN 总线定时寄存器 0.....	109
表 10-23 CAN 总线定时寄存器 1.....	110
表 10-24 CAN 输出控制寄存器.....	110
表 11-1 I2C 控制器访问地址及引脚复用.....	111
表 11-2 I2C 分频锁存器低字节寄存器.....	111
表 11-3 I2C 分频锁存器高字节寄存器.....	111
表 11-4 I2C 控制寄存器.....	112
表 11-5 I2C 发送数据寄存器.....	112
表 11-6 I2C 接收数据寄存器.....	112
表 11-7 I2C 命令控制寄存器.....	112
表 11-8 I2C 状态寄存器.....	113
表 11-9 I2C 从模式控制寄存器.....	113
表 12-1 PWM 控制器访问地址及引脚复用.....	114
表 12-2 PWM 寄存器列表.....	114
表 12-3 PWM 控制寄存器设置.....	114
表 13-1 HPET 控制器访问地址.....	117
表 13-2 HPET 寄存器列表.....	117
表 13-3 General Capabilities and ID Register.....	118
表 13-4 General Configuration Register.....	118
表 13-5 General Interrupt Status Register.....	118
表 13-6 Main Counter Value Register.....	119
表 13-7 Timer N Configuration and Capabilities Register.....	119

表 13-8 Timer N Comparator Value Register.....	120
表 14-1 ACPI 访问地址.....	121
表 14-2 ACPI 状态说明.....	121
表 15-1 RTC 访问地址.....	129
表 16-1 GPIO 控制寄存器.....	133
表 16-2 GPIO 模块内部寄存器物理地址.....	133
表 16-3 按位控制 GPIO 配置寄存器地址.....	134
表 16-4 按字节控制 GPIO 配置寄存器地址.....	134
表 17-1 GMAC 控制器的 PCI 配置头.....	138
表 18-1 USB-EHCI 控制器的 PCI 配置头.....	139
表 18-2 EHCI Capability 寄存器列表.....	140
表 18-3 EHCI Operational 寄存器列表.....	140
表 18-4 USB-OHCI 控制器的 PCI 配置头.....	141
表 18-5 USB-OHCI Operational 寄存器.....	142
表 18-6 USB-XHCI 控制器的 PCI 配置头.....	142
表 19-1 GPU 控制器的 PCI 配置头.....	144
表 19-2 内存控制器软件可见参数列表.....	145
表 20-1 DC 控制器的 PCI 配置头.....	156
表 20-2 单色光标模式.....	165
表 21-1 HDA 控制器的 PCI 配置头.....	178
表 22-1 I2S 控制器的 PCI 配置头.....	180
表 22-2 寄存器定义.....	181
表 22-3 标识寄存器.....	181
表 22-4 配置寄存器.....	181
表 22-5 控制寄存器.....	182
表 23-1 SATA 控制器的 PCI 配置头.....	187
表 23-2 SATA 控制寄存器.....	188
表 24-1 PCIE 控制器的 PCI 配置头.....	190
表 24-2 PCIE 端口 DID 表.....	191
表 25-1 SPI 控制器的 PCI 配置头.....	194
表 25-2 SPI 控制寄存器列表.....	195
表 25-3 SPI 分频系数.....	197
表 26-1 LPC 控制器的 PCI 配置头.....	202



# 1 引言

## 1.1 文档说明

### 1.1.1 文档约定

**注意：**文中寄存器描述中 Reserved 的位域，有的是只读属性，有的是读写属性，不管该位域的属性是什么，软件必须保证不改变该位域的值，也就是说，如果软件需要修改包含 Reserved 位域的寄存器，必须保证写入 Reserved 位域的值与从该位域读出的值一致。

为方便表述，本文使用以下缩写来表示寄存器的属性。

缩写	表示的寄存器属性
RO	只读
WO	只写
R/W	可读可写
R/WIC	只读，对应位写入 1 则将该位清 0
ROS	只读，并且寄存器的值不会因为热复位而发生变化
R/WS	可读可写，并且寄存器的值不会因为热复位而发生变化
R/WIP	可读可写，并且如果向寄存器写入 1，寄存器的值下 1 个时钟周期会自动清 0。因此读此寄存器仅能获得 0 值

## 1.2 桥片概述

龙芯 7A2000 型处理器桥片（后文也简称为 7A2000）是龙芯 7A1000 的升级产品。该芯片通过 HT 高速总线接口与龙芯 3 号系列处理器相连，内部集成 GPU、DDR4 显存控制器，以及 PCIe、SATA、USB、GMAC、HDMI、I2C、UART、GPIO 等接口，可为龙芯处理器提供丰富的南北桥功能。

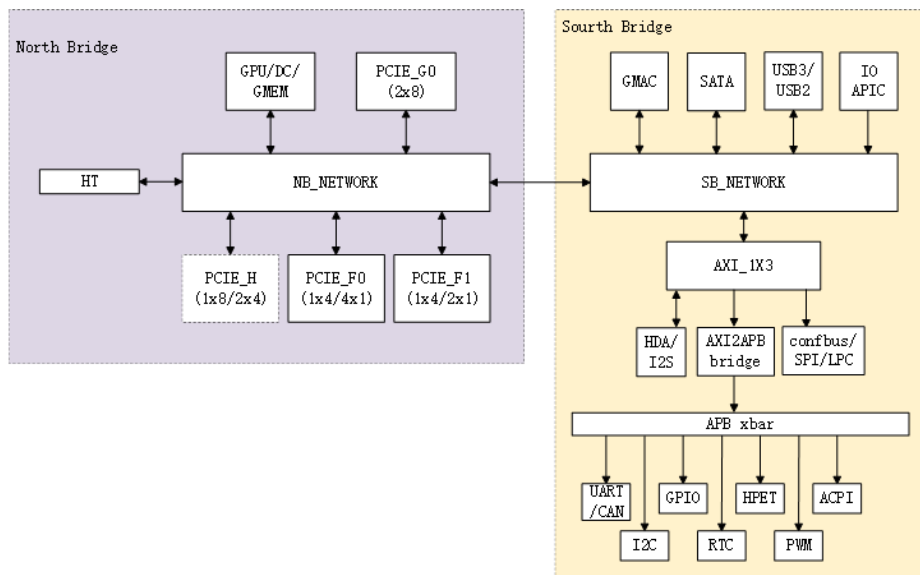


图 1-1 7A2000 顶层结构图



## 桥片主要特性

- 16 位 HT 3.0 接口
- **3D GPU**
- 支持双路显示
- 32 位 DDR4 显存控制器
- 3 个 x8 PCIE 3.0 接口；其中一个可拆分为 2 个独立 PCIE x4 接口
- 1 个 x4 PCIE 3.0 接口；可拆分为 4 个独立 PCIE x1 2.0 接口
- 1 个 x4 PCIE 3.0 接口；可拆分为 2 个独立 PCIE x1 2.0 接口
- 4 个 SATA 3.0
- 最多 4 个 USB 3.0，最多 12 个 USB2.0
- 1 个千兆网口
- HDA/I2S
- RTC/HPET 模块
- 1 个全功能 UART 控制器
- 4 个 CAN 控制器
- 6 个 I2C 控制器
- 1 个 LPC 控制器
- 1 个 SPI 控制器，支持 QSPI
- GPIO 接口
- 支持 ACPI 规范
- 内置温度传感器
- 采用 FC-BGA 封装
- 芯片正常工作表面温度区间为-40°C~85°C

## 1.3 桥片主要功能

### 1.3.1 HT 接口

- 16 位 HT3.0 接口

### 1.3.2 PCIE 接口

- 兼容 PCIE 3.0
- 3 个 x8 接口，2 个 x4 接口
- 1 个 x8 接口可配置为 2 个 x4 接口独立使用

- 其中 1 个 x4 接口可配置为 4 个 x1 接口独立使用
- 另外 1 个 x4 接口可配置为 2 个 x1 接口独立使用
- 输出 8 路 PCIE 差分参考时钟

### 1.3.3 GPU

- 集成一路 DMA
- 集成 MMU
- 支持 4x MSAA
- 支持内存压缩
- 支持动态功耗管理

### 1.3.4 显示接口

- 两路 HDMI 接口，一路 VGA 接口
- VGA 显示内容与一路 HDMI 相同
- 两路独立硬件光标
- 伽玛校正
- 输出抖动
- 支持 1080p
- 支持线性显示缓冲
- 上电序列控制
- 低功耗管理

### 1.3.5 显存接口

- 32 位 DDR4 显存控制器，接口最大速率 DDR4-2400Mbps.
- 最大显存容量 16GB
- 显存通道可配置为 32/16 位

### 1.3.6 USB 接口

- 采用 OHCI、EHCI 和 XHCI 协议
- 4 个独立的 USB3.0 HOST 端口
- 最多 12 个 USB2.0 HOST 端口
- 兼容 USB1.1、USB2.0 和 USB3.0
- 最高传输速度可达 5Gbps

### 1.3.7 SATA 接口

- 4 个独立 SATA3.0 端口
- 支持 SATA 1.5Gbps、3Gbps 和 6Gbps 的传输
- 兼容串行 ATA 3.3 规范和 AHCI 1.3.1 规范
- 低功耗设计

### 1.3.8 网络接口

- 10/100/1000Mbps 自适应以太网
- 兼容 IEEE 802.3
- 仅支持全双工，不支持半双工
- Timestamp 功能
- 支持 CRC 校验码的自动生成与校验，支持前置符生成与删除
- 支持网络唤醒

### 1.3.9 HDA 接口

- 支持 16、18 和 20 位采样精度，支持可变速率
- 最高采样频率 192KHz
- 7.1 频道环绕立体声输出
- 三路音频输入

### 1.3.10 I2S 接口

- 支持 master 模式下 I2S 输入
- 支持 master 模式下 I2S 输出
- 支持 8、16、18、20、24、32 位宽
- 支持单声道和立体声道音频数据
- 支持 (16、22.05、32、44.1、48)KHz 采样频率
- 支持 DMA 传输模式

### 1.3.11 SPI 接口

- 双缓冲接收器
- 极性和相位可编程的串行时钟
- 主模式支持
- 支持变长字节传输(最大 4 字节)

- 支持标准读、连续地址读、快速读、双路 I/O 等 SPI Flash 读模式
- 支持 QSPI

### 1.3.12 LPC 接口

- 符合 LPC1.1 规范
- 扩展支持 TPM 协议
- 支持 Serialized IRQ 规范，提供 17 个中断源

### 1.3.13 UART 接口

- 1 个全功能 UART 串口(支持流控)
- 在寄存器与功能上兼容 NS16550A
- 两路全双工异步数据接收/发送
- 可编程的数据格式
- 16 位可编程时钟计数器
- 支持接收超时检测
- 带仲裁的多中断系统
- 可配置为 4 个两线串口 (TXD, RXD)

### 1.3.14 CAN

- 符合 CAN2.0 规范
- 四路 CAN 接口
- 支持中断

### 1.3.15 I2C 接口

- 兼容 SMBUS (100Kbps)
- 与 PHILIPS I2C 标准相兼容
- 履行双向同步串行协议
- 主从设备支持
- 能够支持多主设备的总线
- 总线的时钟频率可编程
- 可以产生开始/停止/应答等操作
- 能够对总线的状态进行探测
- 支持低速和快速模式
- 支持 7 位寻址和 10 位寻址

- 支持时钟延伸和等待状态

### 1.3.16 PWM

- 4路 32 位可配置 PWM 定时器
- 支持定时器功能
- 支持计数器功能
- 支持防死区发生控制

### 1.3.17 HPET

- 支持 1 个周期性中断
- 支持 2 个非周期性中断

### 1.3.18 RTC

- 计时精确到 1 微秒
- 可产生 3 个定时中断
- 支持定时开关机功能

### 1.3.19 ACPI 接口

- USB/GMAC 可唤醒
- 来电可自动启动
- 支持 S0, S3, S4, S5 状态

### 1.3.20 GPIO 接口

- 2 个专用 GPIO 引脚，56 个复用 GPIO 引脚
- 输入中断功能
- 中断极性、触发类型可设置

### 1.3.21 JTAG 接口

- JTAG 测试接口

## 1.4 桥片工作模式

### 1.4.1 HT 主桥模式

HT 主桥模式作为 7A2000 的主要使用模式，将 7A2000 通过 HT 接口与 3A 系列处理器相连。7A2000 的 HT3.0 总线频率为 3.2GHz，通过 16 位总线与 3A 系列处理器连接。

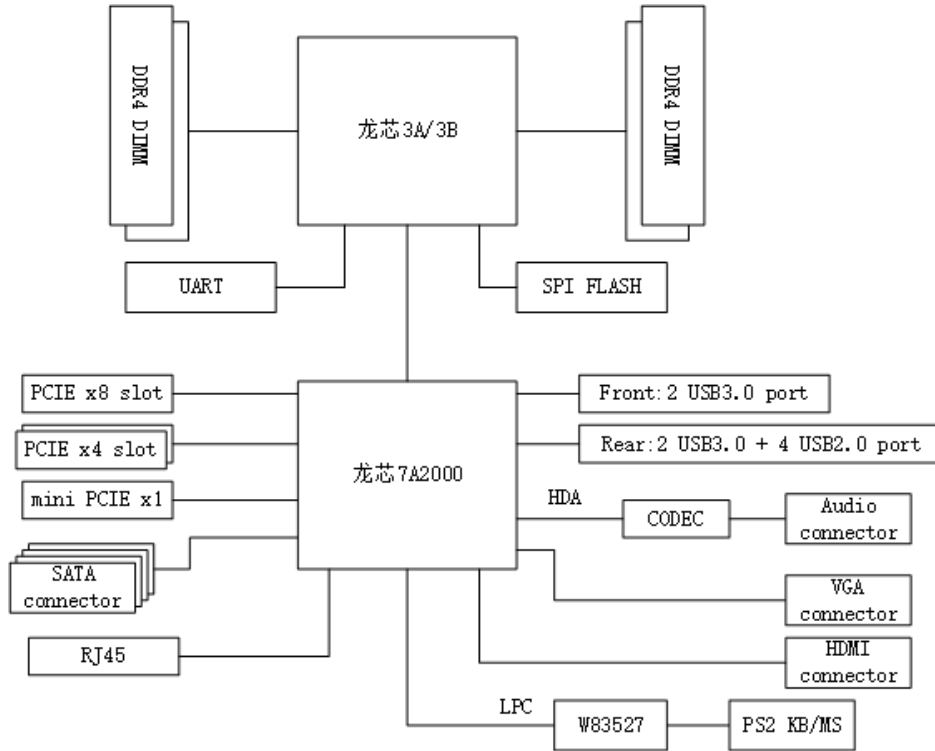


图 1-2 7A2000 单路应用示意图(HT 主桥)

## 2 桥片时钟结构

### 2.1 桥片时钟

7A2000 的时钟结构如图 2-1 所示。

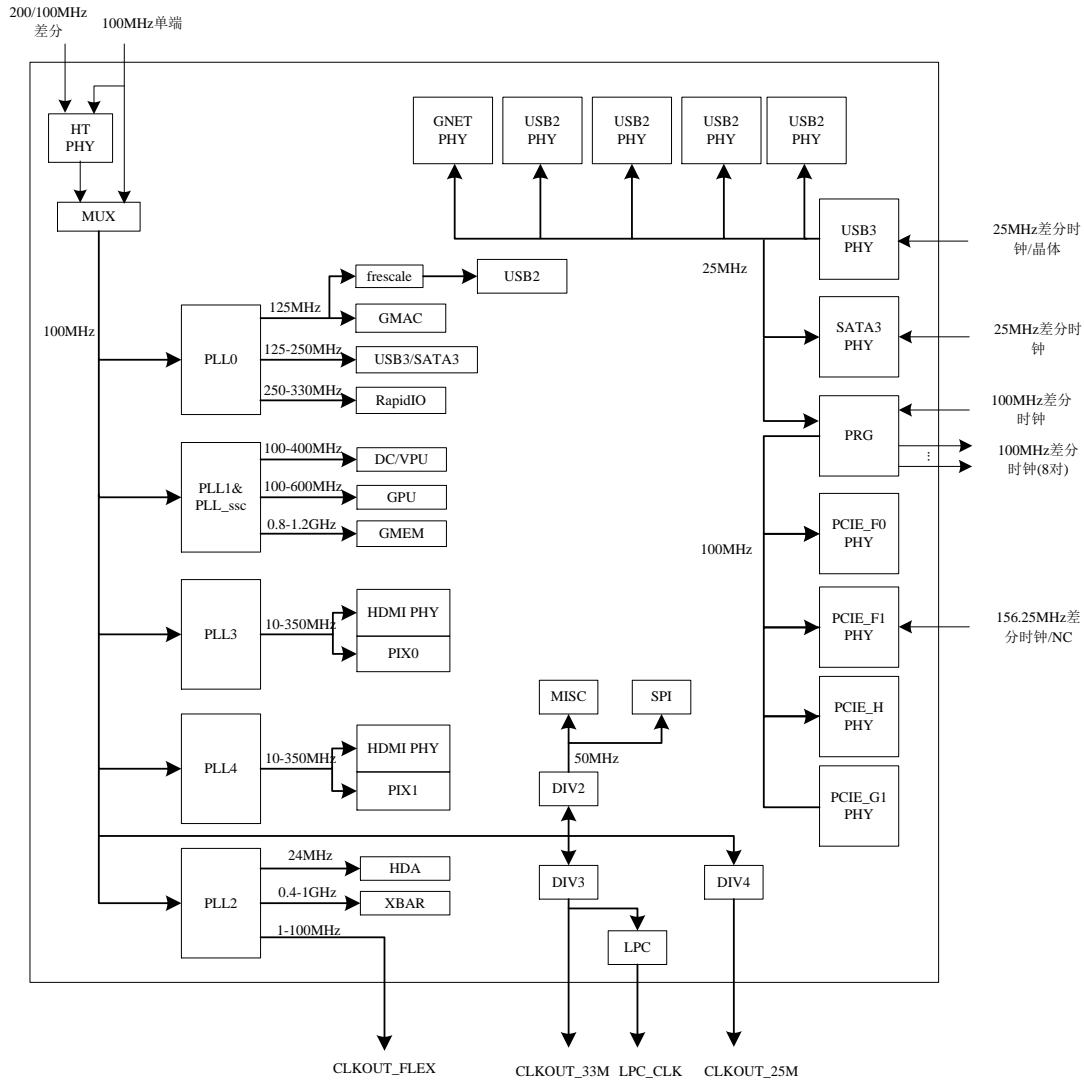


图 2-1 7A2000 的时钟结构

表 2-1 和表 2-2 分别列出了桥片所需要的时钟输入以及桥片的输出时钟。

表 2-1 桥片时钟输入

时钟	频率	说明
SYS_CLKIN	100MHz	桥片 100 MHz 主参考时钟
RTC_XI	32.768KHz	32.768KHz 晶体输入，或者外部 32.768KHz 时钟输入
RTC_XO	32.768KHz	32.768KHz 晶体输出
SYS_TESTCLK	-	保留，可不接

HTCLKp/n	100/200 MHz	HT 差分参考时钟，当 HT 使用内部参考时钟时，可不接
PCIE_REFCLKp/n	100MHz	PCIE 差分参考时钟，当 PCIE 使用 USB 的参考时钟时可不接
RAPIDIO_REFCLKp/n	-	保留，可不接
USB_CLKINp/n	25MHz	USB 差分参考时钟，当不使用 USB3/2 和 GMAC0 并且 PCIE 不使用该时钟时才可以不接
SATA_CLKINp/n	25MHz	SATA 差分参考时钟，当 SATA 使用 USB 的参考时钟时可不接

**注：**晶体参考时钟(RTC\_XI/XO)必须保留，否则影响正常开机以及会导致软件访问电源管理模块的寄存器时死机。

表 2-2 桥片时钟输出

时钟	频率	说明
CLKOUT33M <sup>1</sup>	33.3MHz	33.3 MHz 单端时钟输出
CLKOUT100M	100 MHz	100 MHz 单端时钟输出
CLKOUT25M <sup>1</sup>	25 MHz	25 MHz 单端时钟输出
CLKOUTFLEX <sup>1</sup>	可变	频率可变单端时钟输出
LPC_CLK	33.3MHz	LPC 接口输出参考时钟
PCIE_REFCLKOUTp/n[7:0]	100MHz	8 路 PCIE 输出参考时钟

**注：**CLKOUT33M、CLKOUT25M 和 CLKOUTFLEX 引脚可复用为 GPIO 功能。

## 2.2 时钟功能描述

### 2.2.1 系统参考时钟

桥片的系统参考时钟有两种选择方式，一种是选择单端输入时钟 SYS\_CLKIN，另一种是选择 HT 的差分输入时钟，选择信号为 SYS\_CLKSEL8。无论哪种方式，必须保证系统参考时钟的频率为 100MHz，所以当选择 HT 的差分输入时钟时，需要根据差分时钟的频率对系统配置信号 HT\_CLKSELO 进行设置。

### 2.2.2 内部控制器时钟

桥片内部包含了 5 个主要 PLL，这 5 个 PLL 以系统参考时钟作为输入，用于产生桥片内部控制器需要的各个时钟。每个 PLL 最多可以提供 3 个时钟输出。

这 5 个 PLL 的用途分别为：

一个设备 PLL，产生 USB/SATA、GMAC 控制器的时钟；

一个图形 PLL 用于产生 GPU、DC 以及显存的时钟；

一个系统 PLL 用于产生内部总线、HDA bitclk、flex clkout 的时钟；

两个 PIX PLL 用于产生两个独立的像素时钟，以支持双路独立显示。

PLL 的结构图如图 2-2 所示，



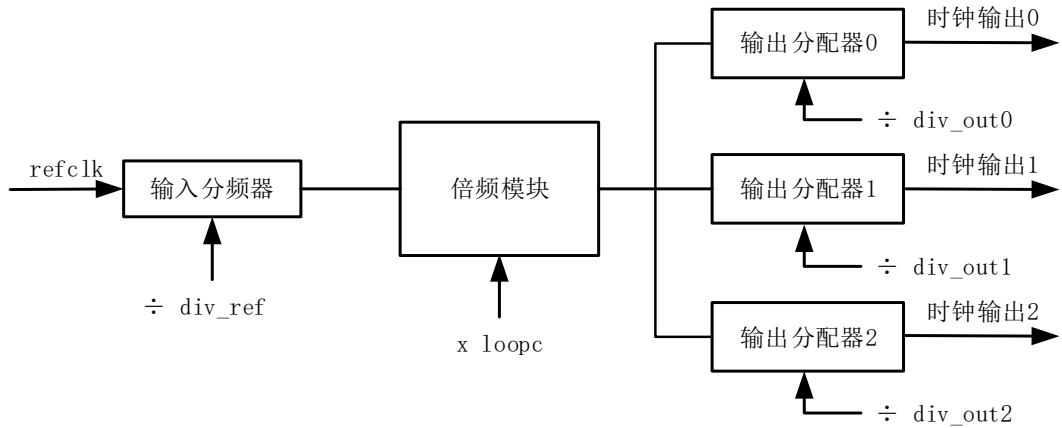


图 2-2 PLL 结构图

输出时钟频率的计算方式如下：

$$\text{clock\_out} = \text{refclk} / \text{div\_ref} * \text{loopc} / \text{divoutN};$$

其中，需要保证输入分频器的输出 ( $\text{refclk} / \text{div\_ref}$ ) 在 20 ~ 40MHz 的范围内，倍频模块倍频后的频率 ( $\text{refclk} / \text{div\_ref} * \text{loopc}$ ) 在 1.2GHz ~ 3.2GHz 的范围内。

PLL 相关的配置及说明见

表 2-3。这些配置通过 PLL0-4 配置寄存器进行控制。

表 2-3 PLL 相关配置信号说明表

信号	位数	方向	说明
pll_div_out0	7	R/W	PLL 输出时钟 0 分频数
pll_div_out1	7	R/W	PLL 输出时钟 1 分频数
pll_div_out2	7	R/W	PLL 输出时钟 2 分频数
pll_loopc	9	R/W	PLL 倍频乘数
pll_div_ref	7	R/W	PLL 输入分频数
pll_locked	1	RO	PLL 锁定
sel_pll_out0	1	R/W	选择 PLL 输出时钟 0
sel_pll_out1	1	R/W	选择 PLL 输出时钟 1
sel_pll_out2	1	R/W	选择 PLL 输出时钟 2
set_pll_param	1	R/W	设置 PLL 配置参数
pll_bypass	1	R/W	PLL 内部 bypass
pll_pd	1	R/W	PLL powerdown

当 SYS\_CLKSEL[1:0]为 00b 时表示可通过软件更改 PLL 的输出频率。这种配置下，桥片启动时默认的时钟频率为外部参考时钟频率，需要在处理器启动过程中对桥片时钟进行软件配置。通过软件修改时钟配置的过程如下：

1. 将 sel\_pll\_out\* 设置为 0;
2. 将 pll\_pd 信号设置为 1;
3. 将 set\_pll\_param 设置为 0,
4. 设置 pll\_div\_ref/pll\_loopc/pll\_div\_out\* 的值;
5. 将 set\_pll\_param 设置为 1;
6. 将 pll\_pd 信号设置为 0;
7. 等待 PLL 锁定信号 pll\_locked 变为 1;
8. 设置 sel\_pll\_out\* 为 1。

另外，针对图形 PLL，额外提供一个支持展频功能 (SSC) 的 PLL 作为备份，该 PLL 对应的配置寄存器为参见 SSCPLL 配置寄存器 0 和 SSCPLL 配置寄存器 1。

根据小数环路分频使能的设置，环路分频系数计算公式有所不同。

当小数环路分频未使能时，环路分频系数为：

$$DIV\_L00PC = 4 * loopc$$

当小数环路分频使能时，环路分频系数为：

$$DIV\_L00PC = 4 * (loopc + frac\_input / (2^{16}) + ! (dither\_disable[1]))$$

第 n 路输出时钟频率计算公式如下：

$$f\_outn = fref / ref\_div * DIV\_L00PC / (2 * div\_outn)$$

在参数配置时要求 VCO 的振荡频率 (fref/ref\_div\*DIV\_L00PC) 在 1.6-3.2GHz 之间。

SSC PLL 具体软件配置方法为：

1. 将 \*\_clk\_sel 设置为 0;
2. 将 pd 信号设置为 1;
3. 设置其他配置参数;
4. 将 pd 信号设置为 0;
5. 等待 PLL 锁定信号 lock 变为 1;
6. 设置 \*\_clk\_sel 为 1。

## 2.2.3 RTC 时钟

RTC 时钟频率要求为 32.768KHz。可选择外接晶体或者晶振，芯片内部 RTC 模块可以自适应这两种时钟输入，无需特别控制。

## 2.2.4 HT PHY 参考时钟

HT PHY 参考时钟有以下两种选择方式，通过芯片引脚 HT\_CLKSEL1 进行选择：

1. 外部 200MHz/100MHz 差分输入

## 2. 使用系统的 100MHz 输入

需要特别注意的是，系统参考时钟所使用的时钟和 HT PHY 参考时钟可以相互选择，在使用时需要遵守如下规则：

表 2-4 时钟使用规则

SYS_CLKSEL[8]	HT_CLKSEL1	说明
0	0	系统时钟选择 SYS_CLKIN HT PHY 参考时钟选择外部 200MHz/100MHz 差分输入
0	1	系统时钟选择 SYS_CLKIN HT PHY 参考时钟选择 SYS_CLKIN
1	0	系统时钟和 HT PHY 参考时钟均选择外部 200MHz/100MHz 差分输入
1	1	禁止设置

## 2.2.5 PCIE PHY 参考时钟

7A2000 的 PCIE 有 4 个 PHY，它们共用内部参考时钟。通过寄存器 CFG.0648[8]可以从下面两个时钟源对参考时钟进行选择：

1. 外部 100MHz 差分输入(PCIE REFCLKp/n)
2. USB PHY 的 25MHz 参考时钟(USB CLKINp/n)

## 2.2.6 USB PHY 参考时钟

USB 的参考时钟有以下两种选择方式，通过芯片引脚 USB\_CLKSEL 进行选择：

1. 外接 25MHz 晶体；
2. 外接 25MHz 差分输入；

USB 的参考时钟会提供给 GNET PHY 和 PCIE PHY 使用，所以只有在所有使用该参考时钟的模块都不工作时才可以不接 USB PHY 参考时钟。

## 2.2.7 SATA PHY 参考时钟

SATA PHY 参考时钟有以下两个来源，通过配置寄存器 CFG.0740[6]进行选择。

1. 外部 25MHz 差分输入(SATA CLKINp/n)
2. USB PHY 的 25MHz 参考时钟(USB CLKINp/n)

## 2.2.8 GMAC PHY 参考时钟

GMAC PHY 参考时钟有以下两个来源，通过配置寄存器 CFG.0770[22]进行选择。

1. 使用 USB PHY 的 25MHz 参考时钟(USB CLKINp/n)

2. 使用内部 PLL 生成的 GMAC 控制器时钟

## 2.2.9 PHY 时钟之间关系

对于 USB3/2、PCIE、SATA 和 GMAC PHY，在初始化之前需要保证其参考时钟已处于稳定状态，下表列出各参考时钟稳定的标志以及判断标准。

表 2-5 PHY 时钟之间关系

模块	参考时钟来源	参考时钟稳定标志
USB3 PHY	25MHz 晶体	USB3 osc 时钟稳定(USB3 PHY 配置寄存器 3bit24 置为 1)
	25MHz 差分输入	输入时钟稳定
USB2 PHY	USB3 PHY	参考 USB3 PHY
GMAC PHY	USB3 PHY	参考 USB3 PHY
	内部 PLL0	PLL0 lock(PLL0 配置寄存器 bit39 变为 1)
SATA PHY	USB3 PHY	参考 USB3 PHY
	25MHz 差分输入	输入时钟稳定
PRG	USB3 PHY	参考 USB3 PHY
	100MHz 差分输入	输入时钟稳定
PCIE	PRG	PRG 模块输出时钟稳定 (PRG 模块配置寄存器 0 bit48 变为 1)

### 3 地址空间

龙芯 7A 内部地址位宽为 48 位。作为龙芯 3A 处理器的配套桥片，龙芯 7A 的地址空间受限于主桥所支持的地址位数。在 HT 接口作为主桥时，默认情况下 HT 控制器仅支持 40 位地址，通过配置 HT 扩展地址寄存器，可将 HT 控制器的地址空间扩展到 64 位，从而可访问 7A 内部所有 48 位地址域。在配置时需要注意，必须保证 HT 链路两端的地址位宽一致。

#### 3.1 处理器的访问

从 CPU 的视角——即 CPU 可访问的设备地址空间——来看，桥片的地址空间包括 3 部分：配置空间、PCI IO 空间和 PCI MEM 空间。桥片的地址空间与 PCI 定义的地址空间形式相同。

1. 配置空间：该地址空间用来访问桥片内部设备（包括通过 PCIE 总线扩展的设备）的配置头，其地址组成符合 PCI 配置访问的地址组织形式；
2. IO 空间：该地址空间用于访问 PCI 协议定义的 IO 地址空间。在桥片中只有 PCIE 和 LPC 有这段地址空间，用于通过 IO 类型的请求访问 PCIE 控制器的下游设备。
3. MEM 空间：除了以上两种地址空间之外的所有地址空间为 MEM 空间。

桥片的配置空间对应于 HT 总线的 HT 总线配置空间，大小为 32MB。桥片的 PCI IO 空间对应于 HT 总线 IO 空间，大小为 32MB。桥片的 PCI MEM 空间用来访问桥片内部 PCIE 设备的 MEM 空间和除 PCIE 外的其他设备的地址空间，包括控制寄存器、显存、confbus 等。

##### 3.1.1 PCI 设备和功能

桥片内部设备按照 PCI 设备树进行组织，每个设备都包含了一个标准的 PCI 设备头。这些设备包括：GPU、DC、PCIE、USB、SATA、GMAC、HDA/I2S、MISC 低速设备块、confbus、LPC 和 SPI。各个设备的总线号、设备号、功能号见表 3-1。

表 3-1 各个设备的配置头访问对应关系

Bus: Device: Function	Function Description
Bus 0:Device 0:Function 0	HT
Bus 0:Device 2:Function 0	MISC
Bus 0:Device 3:Function 0	GMACO
Bus 0:Device 3:Function 1	保留
Bus 0:Device 4:Function 0	USB0 OHCI
Bus 0:Device 4:Function 1	USB0 EHCI
Bus 0:Device 5:Function 0	USB1 OHCI
Bus 0:Device 5:Function 1	USB1 EHCI

Bus 0:Device 6:Function 0	GPU
Bus 0:Device 6:Function 1	DC
Bus 0:Device 6:Function 2	HDA1
Bus 0:Device 6:Function 3	保留
Bus 0:Device 7:Function 0	HDA <sup>1</sup>
Bus 0:Device 7:Function 1	I2S <sup>1</sup>
Bus 0:Device 8:Function 0	SATA
Bus 0:Device 9:Function 0	PCIE_F0 Port0 <sup>2</sup>
Bus 0:Device 10:Function 0	PCIE_F0 Port1 <sup>2</sup>
Bus 0:Device 11:Function 0	PCIE_F0 Port2 <sup>2</sup>
Bus 0:Device 12:Function 0	PCIE_F0 Port3 <sup>2</sup>
Bus 0:Device 13:Function 0	PCIE_F1 Port0 <sup>3</sup>
Bus 0:Device 14:Function 0	PCIE_F1 Port1 <sup>3</sup>
Bus 0:Device 15:Function 0	PCIE_G0 port0 <sup>4</sup>
Bus 0:Device 16:Function 0	PCIE_G0 port1 <sup>4</sup>
Bus 0:Device 17:Function 0	保留
Bus 0:Device 18:Function 0	保留
Bus 0:Device 19:Function 0	PCIE_H port0 <sup>6</sup>
Bus 0:Device 20:Function 0	PCIE_H port1 <sup>6</sup>
Bus 0:Device 21:Function 0	保留
Bus 0:Device 22:Function 0	SPI
Bus 0:Device 23:Function 0	LPC <sup>7</sup>
Bus 0:Device 25:Function 0	USB XHCI

**注：**

1. 当 hda0\_i2s\_en 为 1 时，HDA 控制器才可以被发现；当 hda0\_i2s\_en 为 0 时，I2S 控制器才可以被发现。
2. 当 PCIE\_F0 工作在 x4 模式时，只有 Port 0 可见，Port 1-3 不可见；当 PCIE\_F0 工作在非 x4 模式时，Port 0-3 可见。
3. 当 PCIE\_F1 工作在 x4 模式时，只有 Port 0 可见，Port 1 不可见；当 PCIE\_F1 工作在非 x4 模式时，Port 0-1 可见。
4. 当 PCIE\_G0 只可工作在 2x8 模式，Port 0-1 可见。
5. 当 PCIE\_H 工作在 x8 模式时，只有 Port 0 可见，Port 1 不可见；当 PCIE\_H 工作在 x4 模式时，Port 0-1 可见。当 PCIE\_H 作为主桥时，Port0 和 1 均不可见。
6. 当 LPC 模块被使能后，LPC(D23:F0) 才可见。

当配置头访问的总线号、设备号、功能号和地址偏移无效时，写操作无效。读操作得到的数据为全 1。

### 3.1.2 PCI 配置访问地址

处理器可以通过两个地址空间来访问桥片的配置空间。一个是 HT 定义的标准配置访问空间 (0xFD\_FE00\_0000 - 0xFD\_FFFF\_FFFF)，另一个是 HT 的保留地址空间 (0xFE\_0000\_0000 - 0xFE\_1FFF\_FFFF)。通过 HT 标准配置访问空间访问的每个桥片设备的配置空间大小为 256 字节；通过保留地址空间访问的每个桥片设备的配置空间大小为 4K 字节。

当使用 HT 定义的标准配置访问空间 (0xFD\_FE00\_0000 - 0xFD\_FFFF\_FFFF) 来访问桥片时，每个设备的最大配置空间大小为 256 字节。其中，地址的 [39:24] 决定配置头类型 (0xFDFE 是 Type0, 0xFDFE 是 Type1)；[23:16] 表示总线号 (Bus Number)；[15:11] 表示设备号 (Device Number)；[10:8] 表示功能号 (Function Number)；[7:0] 表示偏移 (offset)。图 3-1 是 CPU 使用 HT 标准配置访问空间访问 PCI 配置空间的地址段含义示意图。

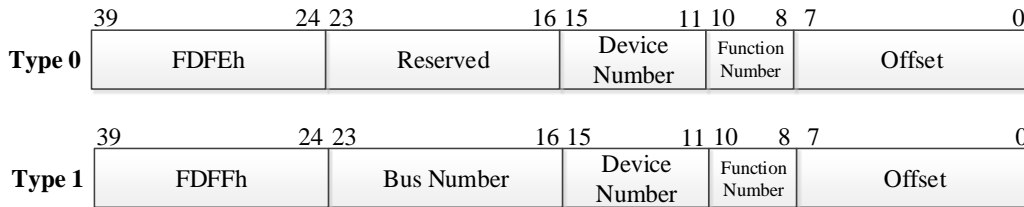


图 3-1 配置访问地址示意图

当使用 HT 的保留地址空间 (0xFE\_0000\_0000 - 0xFE\_1FFF\_FFFF) 来访问桥片时，每个设备的最大配置空间大小为 4K 字节。其中，地址的 [39:28] 决定配置头类型 (0xFE0 是 Type0, 0xFE1 是 Type1)；[23:16] 表示总线号 (Bus Number)；[15:11] 表示设备号 (Device Number)；[10:8] 表示功能号 (Function Number)；[27:24] 和 [7:0] 组合起来表示偏移 (offset)。图 3-2 是 CPU 使用 HT 保留地址空间访问 PCI 配置空间的地址段含义示意图。

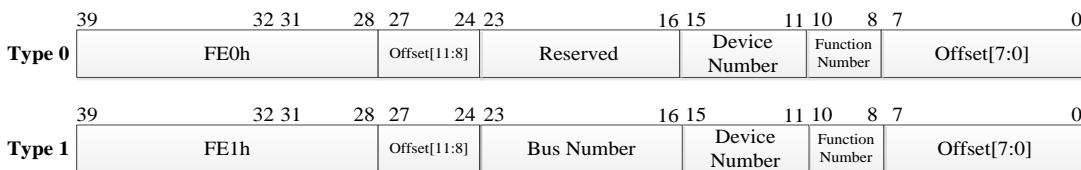


图 3-2 配置访问地址示意图

### 3.1.3 桥片设备地址空间分配示例

对桥片设备的访问主要通过 PCI MEM 空间来完成。软件可以在该地址段内任意分配桥片各个设备的访问地址。桥片的内部 PCI 设备包括：GPU、DC、GMEM、PCIE、USB、SATA、GMAC、had、I2S、LPC、SPI。这些设备（除 LPC 外）的访问地址可以由软件动态分配。一种分配方式如下：通过扫描 PCI 总线，读取各个设备（PCI 方式方位）的配置空间，获取各个设备使用的 MEM 空间和 I/O 空间大小，系统软件从 0x20000, 0000~0x7fff, ffff 这个地址内分配合

适大小的 MEM 空间, 从 0x1800, 0000~0x19ff, ffff 这个地址内分配合适大小的 I/O 空间(PCI 设备)。

除了这些 PCI 类型的设备外, 桥片内还包含一些使用固定地址访问的设备, 比如: 中断控制器、HPET 控制器、confbus 配置寄存器、MISC 低速设备块和 LPC。

表 3-2 和

表 3-3 给出了桥片固定地址设备和 PCI 设备的一种地址分配示例, 以及它们的地址空间大小和访问类型。访问类型中, B 表示字节访问 (1byte), H 表示半字访问 (2byte), W 表示字访问 (4byte), D 表示双字访问 (8byte), Q 表示 4 字访问 (16byte), C 表示 cacheline 访问。

表 3-2 桥片固定地址设备地址空间

模块	地址空间	地址空间大小	访问类型
INT	0x1000, 0000~0x1000, 0fff	4K	BHW
HPET	0x1000, 1000~0x1000, 1fff	4K	BW
CONF REG	0x1001, 0000~0x1001, ffff	64K	BHW
MISC	0x1008, 0000~0x100f, ffff	512K	BW
LPC REG	0x1000, 2000~0x1000, 2fff	4K	W
LPC MEM	0x1200, 0000~0x13ff, ffff	32M	BHWDQC
LPC I/O	0x1800, 0000~0x1800, ffff	64K	B
LPC TPM	0x1801, 0000~0x1801, ffff	64K	B

表 3-3 桥片 PCI 设备地址空间描述

模块	地址空间大小	访问类型
GPU	256M	W
Graphic Memory (共享显存)	可配置	BHWDQC
DC	64K	W
PCI E I/O	可配置	BHW
PCI E MEM	可配置	BHW
SPI MEM	16M	BHWDQC
USB0-EHCI	32K	W
USB0-OHCI	32K	W
USB1-EHCI	32K	W
USB1-OHCI	32K	W
USB-XHCI	1MB	W
SATA	1K	W
GMACO	32K	W
HDA	64K	BHW
HDA1	64K	BHW



I2S	64K	W
SPI REG	4K	B
IOMMU	1K	BHWD

上述设备中，除了 PCIEIO/MEM 和 Graphic Memory 外，其它设备的地址空间大小是固定不变的，软件可以改变地址空间的起始地址。PCIEIO/MEM 地址空间的大小需要根据所接设备来决定。

当使用桥片内部的显存时，Graphic Memory 的大小根据所接显存的容量来决定。BIOS 需要通过访问桥片通用配置寄存器来修改 GPU 配置头的 BAR 寄存器 2/3 的 MASK 值来配置 Graphic Memory 的大小，然后软件再通过 PCI 扫描的方式来获得显存的大小。

在使用 PCIE 外接独立显卡的情况下，独立显卡自带的独立显存空间位于 PCIE MEM 地址空间内，作为 PCIE 设备统一管理。

### 3.2 DMA 的访问

从 DMA 访问的视角——即桥片内部设备主动发起的访存地址空间——来看，可使用的地址空间包括：处理器的内存空间和桥片的显存空间。

桥片内部可以发起 DMA 操作的设备包括：GPU、DC、PCIE、USB、SATA、GMAC、HDA/I2S，其中 GPU、DC 可以根据地址路由规则分别访问处理器的内存空间和桥片的显存空间，而其他设备仅可以访问处理器的内存空间。

处理器的内存空间大小根据系统的结点个数而不同，总的 DMA 地址空间的大小为 256TB(使能 HT 64 位地址空间情况下)。为了支持多处理器系统，我们将桥片内部地址的某几位（最多 4 位）作为桥片内部设备 DMA 访问的目的结点号使用。也就是说，桥片最多可以支持 16 个结点的处理器系统。在多节点的情况下，每个节点所分配的最大地址空间与节点数有关，比如对于 4 个结点的系统，DMA 的地址空间必须位于结点地址空间的低 64TB 以内。

桥片的显存空间的基址通过 PCI 扫描程序进行动态分配。GPU、DC 分别有自己独立的地址路由窗口，命中在对应窗口的访问在地址转换后被路由到显存中，未命中的访问路由到处理器的内存空间。对于 GPU、DC 的地址窗口命中规则和地址转换规则如下表所示。

表 3-4 DC 访问地址窗口命中和转换规则

窗口	命中规则	转换规则
0	addr_in&gpu_bar1_mask ==gpu_bar1_base&gpu_bar1_mask	addr_in&gpu_bar1_mask
1	地址 bit48 为 1	仅保留低 48 位

表 3-5 GPU 访问地址窗口命中和转换规则

窗口	命中规则	转换规则
0	$addr\_in \& gpu\_bar1\_mask == gpu\_bar1\_base \& gpu\_bar1\_mask$	$addr\_in \& gpu\_bar1\_mask$
1	地址 bit48 为 1	仅保留低 48 位
2	$addr\_in \& \sim gpu\_win0\_mask == gpu\_win0\_base \& \sim gpu\_win0\_mask$	$addr\_in \& gpu\_win0\_mask   gpu\_win0\_mmap \& \sim gpu\_win0\_mask$
3	$addr\_in \& \sim gpu\_win1\_mask == gpu\_win1\_base \& \sim gpu\_win1\_mask$	$addr\_in \& gpu\_win1\_mask   gpu\_win1\_mmap \& \sim gpu\_win1\_mask$

### 3.3 龙芯 3A+7A 系统地址空间分配示例

下面以龙芯 3A+桥片系统为例来说明整个计算机系统的地址空间分配。3A+桥片的一种地址空间划分方式如表 3-6 所示：

表 3-6 龙芯 3 号处理器+龙芯 7A 桥片地址空间划分示例

类型	范围
7A MEM 高地址空间	$MEM\_UP\_LIMIT \sim 0xfc, ffff, ffff$
内存高地址空间	$0x8000, 0000 \sim MEM\_UP\_LIMIT$
7A MEM 低地址空间	$0x2000, 0000 \sim 0x7fff, ffff$
处理器 PCI 空间和其他	$0x1c00, 0000 \sim 0x1fff, ffff$
7A IO 空间和配置空间	$0x1800, 0000 \sim 0x1bff, ffff$
内存低地址空间	$0x0 \sim 0x0fff, ffff$

**注：**表中地址为低位地址，不包含结点号和高位地址。

在表 3-6 的地址空间分配方式中，

$0x0000, 0000 \sim 0x0fff, ffff$  为系统的低 256MB 内存空间。

$0x1800, 0000 \sim 0x19ff, ffff$  为桥片的 PCI IO 空间，软件可以将桥片内 PCIE 设备的 IO 空间分配到这段地址空间内，这段地址通过一级 XBAR 的配置窗口映射到  $0xefd, fc00, 0000 \sim 0xefd, fdff, ffff$  (HT1 的 IO 空间)。

$0x1a00, 0000 \sim 0x1bff, ffff$  为桥片的配置空间，用来访问桥片内部设备的配置头，访问方式与 PCI 协议兼容，地址位的 bit[23:8] 依次对应 bus 号、设备号、func 号，这段地址通过一级 XBAR 的配置窗口映射到  $0xefd, fe00, 0000 \sim 0xefd, ffff, ffff$  (HT1 的总线配置空间)。

$0x1c00, 0000 \sim 0x1fff, ffff$  为 3A 的低速设备空间和处理器配置空间。

$0x2000, 0000 \sim 0x7fff, ffff$  为桥片的 PCIMEM 空间，这段地址通过一级 XBAR 的配置窗口映射到  $0xe00, 2000, 0000 \sim 0xe00, 7fff, ffff$  (HT1 的 MEM 空间)。

$0x8000, 0000 \sim MEM\_UP\_LIMIT$  为 3A 的高内存地址空间。

$MEM\_UP\_LIMIT \sim 0xfc, ffff, ffff$  为桥片的 PCI MEM 空间，这段地址通过一级 XBAR 的配置窗口映射到  $0xe00, 0000, 0000 + MEM\_UP\_LIMIT \sim 0xfc, ffff, ffff$  (HT1 的 MEM 空间)。

## 4 桥片配置寄存器

桥片设置了一些寄存器，用来配置桥片的某些特性，而不是针对特定的接口（PCIE、USB 等）。这些寄存器统一安排在桥片的配置寄存器空间（不是 PCI 配置访问空间）。桥片配置寄存器的地址空间大小为 64KB，起始地址（桥片内部空间）由 BIOS 配置。

表 4-1 桥片配置寄存器列表

地址偏移	名称	访问类型	描述
0x0420	FUNC_CONFIG0	R/W, RO	功能配置寄存器 0
0x0430	FUNC_CONFIG1	R/W	功能配置寄存器 1
0x0440	PAD	R/W	PAD 复用配置
0x0448	USBOC_SEL	R/W	USB OC 复用配置
0x0460	FIX_ADDR	R/W	中断控制器和 HPET 默认地址配置
0x0480	PLL_0	R/W, RO	PLL0 配置
0x0490	PLL_1	R/W, RO	PLL1 配置
0x04a0	PLL_2	R/W	PLL2 配置
0x04b0	PLL_PIX_0	R/W, RO	PLL_PIX0 配置
0x04c0	PLL_PIX_1	R/W, RO	PLL_PIX1 配置
0x04d0	FREQSCALE	R/W	设备分频配置
0x04e0	SSCPPLL_0	R/W	SSC PLL 配置 0
0x04e8	SSCPPLL_1	R/W	SSC PLL 配置 1
0x0770	GMACO	R/W, R	GMACO 配置寄存器
0x3ff8	CHIP_ID	RO	芯片版本号

### 4.1 通用配置寄存器 0

地址偏移：0420h

默认值：cccc\_3ce0h

本寄存器包含 PCIE、图形处理单元（GPU、显示控制器、显存）相关的配置信息。

表 4-2 通用配置寄存器 0

位域	名称	访问	描述
63:62	Reserved	R/W	保留
61	gmem_lpconf_en	R/W	GMEM 配置空间低功耗使能 0: 不使能 1: 使能

60	gmem_lpmc_en	R/W	显存控制器低功耗使能 0: 不使能 1: 使能
59	Reserved	R/W	保留
58	Reserved	R/W	保留
57	disable_loop	R/W	禁止 7A 中 HT 的回环访问通路 0: 不禁止 1: 禁止
56	disable_gmem_confspace	R/W	禁止访问 GMEM 配置空间 0: 允许访问 1: 禁止访问
55:50	Reserved	R/W	保留
49	pcie_g0_p1_clk_ok	RO	pcie_g0 端口 1 时钟 ready 0: 没有时钟 1: 时钟正常
48	pcie_g0_p0_clk_ok	RO	pcie_g0 端口 0 时钟 ready 0: 没有时钟 1: 时钟正常
47	pcie_h_p1_clk_ok	RO	pcie_h 端口 1 时钟 ready 0: 没有时钟 1: 时钟正常
46	pcie_h_p0_clk_ok	RO	pcie_h 端口 0 时钟 ready 0: 没有时钟 1: 时钟正常
45	pcie_f1_p1_clk_ok	RO	pcie_f1 端口 1 时钟 ready 0: 没有时钟 1: 时钟正常
44	pcie_f1_p0_clk_ok	RO	pcie_f1 端口 0 时钟 ready 0: 没有时钟 1: 时钟正常
43	pcie_f0_p0_clk_ok	RO	pcie_f0 端口 3 时钟 ready 0: 没有时钟 1: 时钟正常
42	pcie_f0_p0_clk_ok	RO	pcie_f0 端口 2 时钟 ready 0: 没有时钟 1: 时钟正常
41	pcie_f0_p0_clk_ok	RO	pcie_f0 端口 1 时钟 ready 0: 没有时钟 1: 时钟正常
40	pcie_f0_p0_clk_ok	RO	pcie_f0 端口 0 时钟 ready 0: 没有时钟 1: 时钟正常
39:37	Reserved	R/W	保留
36	pcie_g0_uca_en	R/W	pcie_g0 uncached 访问加速使能 0: 关闭访问加速 1: 打开访问加速
35	pcie_h_uca_en	R/W	pcie_h uncached 访问加速使能 0: 关闭访问加速 1: 打开访问加速
34	pcie_f1_uca_en	R/W	pcie_f1 uncached 访问加速使能 0: 关闭访问加速 1: 打开访问加速
33	pcie_f0_uca_en	R/W	pcie_f0 uncached 访问加速使能 0: 关闭访问加速 1: 打开访问加速

32	graphic_uca_en	R/W	GPU/DC uncache 访问加速时能 0: 关闭访问加速 1: 打开访问加速
31:28	Reserved	R/W	保留
27	pcie_g0_p1_clken	R/W	使能 pcie_g0 的 port1 时钟 0: 关闭时钟 1: 打开时钟
26	pcie_g0_p0_clken	R/W	使能 pcie_g0 的 port0 时钟 0: 关闭时钟 1: 打开时钟
25	pcie_g0_enable	R/W	使能 pcie_g0 控制器 0: 禁止访问 1: 允许访问
24	pcie_g0_soft_reset	R/W	pcie_g0 的软件复位 0: 解除复位 1: 保持复位
23	pcie_h_p1_clken	R/W	使能 pcie_h 的 port1 时钟 0: 关闭时钟 1: 打开时钟
22	pcie_h_p0_clken	R/W	使能 pcie_h 的 port0 时钟 0: 关闭时钟 1: 打开时钟
21	pcie_h_enable	R/W	使能 pcie_h 控制器 0: 禁止访问 1: 允许访问
20	pcie_h_soft_reset	R/W	pcie_h 的软件复位 0: 解除复位 1: 保持复位
19	pcie_f1_p1_clken	R/W	使能 pcie_f1 的 port1 时钟 0: 关闭时钟 1: 打开时钟
18	pcie_f1_p0_clken	R/W	使能 pcie_f1 的 port0 时钟 0: 关闭时钟 1: 打开时钟
17	pcie_f1_enable	R/W	使能 pcie_f1 控制器 0: 禁止访问 1: 允许访问
16	pcie_f1_soft_reset	R/W	pcie_f1 的软件复位 0: 解除复位 1: 保持复位
15:14	Reserved	R/W	保留
13	pcie_f0_p3_clken	R/W	使能 pcie_f0 的 port3 时钟 0: 关闭时钟 1: 打开时钟
12	pcie_f0_p2_clken	R/W	使能 pcie_f0 的 port2 时钟 0: 关闭时钟 1: 打开时钟
11	pcie_f0_p1_clken	R/W	使能 pcie_f0 的 port1 时钟 0: 关闭时钟 1: 打开时钟
10	pcie_f0_p0_clken	R/W	使能 pcie_f0 的 port0 时钟 0: 关闭时钟 1: 打开时钟
9	pcie_f0_enable	R/W	使能 pcie_f0 控制器 0: 禁止访问 1: 允许访问

8	pcie_f0_soft_reset	R/W	pcie_f0 的软件复位 0: 解除复位 1: 保持复位
7	dc_clken	R/W	使能 dc 的时钟 0: 关闭时钟 1: 打开时钟
6	gpu_clken	R/W	使能 gpu 的时钟 0: 关闭时钟 1: 打开时钟
5	gmem_clken	R/W	使能 gmem 的时钟 0: 关闭时钟 1: 打开时钟
4	gpu_soft_reset	R/W	gpu 的软件复位, 高有效
3	Reserved	R/W	保留
2	pcie_clkssel	R/W	pcie f1 的时钟选择。 0: 选择内部参考时钟 1: 选择 PAD 输入时钟
1	Reserved	R/W	保留
0	default_route_cfg0	R/W	该位必须设置为 0。

## 4.2 通用配置寄存器 1

地址偏移: 0430h

默认值: 0009\_99f2h

本寄存器包含 USB、SATA、GMAC、HDA/I2S、LPC、SPI 相关的配置信息。

表 4-3 通用配置寄存器 1

位域	名称	访问	描述
63	hda1_int_route	R/W	HDA1 中断引脚控制 0: 使用 hda1_int 1: 使用 hda0_int
62	lpc_ioaddr_new	R/W	需要设置成 1
61	rtc_access_speed	R/W	RTC 访问通路控制 0: 低速访问通路, 读取 RTC 时间延迟长 1: 低速访问通路, 读取 RTC 时间延迟短
60	hda_dma_64	R/W	使能 HDA64 位 DMA 地址模式 0: 使用 32 位 DMA 地址模式 1: 使用 64 位 DMA 地址模式
59:56	Reserved	R/W	保留
52	Reserved	R/W	保留
51:44	Reserved	R/W	保留
43	sata_clken	R/W	SATA 时钟使能 0: 没有时钟 1: 时钟正常
42	sata_en	R/W	SATA 访问使能 0: 禁止访问 1: 允许访问
41	Reserved	R/W	保留

40	sata_cntl_soft_reset	R/W	SATA 控制器软件复位 0: 解除复位 1: 保持复位
39	gpio50_int_remap	R/W	GPIO50 引脚中断重映射 0: 默认映射方式 1: 将 GPIO50 中断映射到 GPIO3 对应的中断引脚
38	gpio15_int_remap	R/W	GPIO15 引脚中断重映射 0: 默认映射方式 1: 将 GPIO15 中断映射到 GPIO2 对应的中断引脚
37	gpio14_int_remap	R/W	GPIO14 引脚中断重映射 0: 默认映射方式 1: 将 GPIO14 中断映射到 GPIO1 对应的中断引脚
36	gpio13_int_remap	R/W	GPIO13 引脚中断重映射 0: 默认映射方式 1: 将 GPIO13 中断映射到 GPIO0 对应的中断引脚
35:34	Reserved	R/W	保留
33	hpet_int_ctrl	R/W	hpet 中断分离控制 0: HPET 的中断全部分配到 hpet0_int 1: HPET 的中断分别分配到 hpet0/1/2_int
32	lpc_en	R/W	LPC 控制器使能 0: 禁止访问 1: 允许访问
31	lpc_uca_en	R/W	LPC uncached 加速使能 0: 关闭访问加速 1: 打开访问加速
30	spi_uca_en	R/W	SPI uncached 加速使能 0: 关闭访问加速 1: 打开访问加速
29	conf_uca_en	R/W	配置寄存器 uncached 加速使能 0: 关闭访问加速 1: 打开访问加速
28	misc_uca_en	R/W	低速 misc 设备 uncached 加速使能 0: 关闭访问加速 1: 打开访问加速
27	aud_uca_en	R/W	HDA0/HDA1/I2S uncached 加速使能 0: 关闭访问加速 1: 打开访问加速
26	gmac_uca_en	R/W	gmac uncached 加速使能 0: 关闭访问加速 1: 打开访问加速
25	sata_uca_en	R/W	sata uncached 加速使能 0: 关闭访问加速 1: 打开访问加速
24	usb_uca_en	R/W	usb uncached 加速使能 0: 关闭访问加速 1: 打开访问加速
23:22	Reserved	R/W	保留
21	hda1_en	R/W	HDA1 控制器访问使能 0: 禁止访问 1: 允许访问
20	hda0_i2s_en	R/W	HDA0/I2S 控制器使能 0: 使能 I2S 控制器 1: 使能 HDA0 控制器
19	usb3_clken	R/W	usb3 时钟使能 0: 没有时钟 1: 时钟正常

18	usb3_en	R/W	usb3 访问使能 0: 禁止访问 1: 允许访问
17	usb3_phy_soft_reset	R/W	usb3 PHY 软件复位 0: 解除复位 1: 保持复位
16	usb3_cntl_soft_reset	R/W	usb3 控制器软件复位 0: 解除复位 1: 保持复位
15	usb21_clken	R/W	usb21 时钟使能 0: 没有时钟 1: 时钟正常
14	usb21_en	R/W	usb21 访问使能 0: 禁止访问 1: 允许访问
13	Reserved	R/W	保留
12	usb21_cntl_soft_reset	R/W	usb21 控制器软件复位 0: 解除复位 1: 保持复位
11	usb20_clken	R/W	usb20 时钟使能 0: 没有时钟 1: 时钟正常
10	usb20_en	R/W	usb20 访问使能 0: 禁止访问 1: 允许访问
9	usb2_phy_soft_reset	R/W	usb2 PHY 软件复位 0: 解除复位 1: 保持复位
8	usb20_cntl_soft_reset	R/W	usb20 控制器软件复位 0: 解除复位 1: 保持复位
7	Reserved	R/W	保留
6	Reserved	R/W	保留
5	gmac0_clken	R/W	gmac0 时钟使能 0: 没有时钟 1: 时钟正常
4	gmac0_sdb_flowctrl	R/W	gmac0 流控使能 0: 流控关闭 1: 流控打开
3:2	Reserved	R/W	保留
1	usb_ehci_dma64_en	R/W	使能 usb EHCI 64 位 DMA 地址模式 0: 使用 32 位 DMA 地址模式 1: 使用 64 位 DMA 地址模式
0	default_route_cfg1	R/W	使用固定地址访问 USB、SATA、GMAC 等设备。 0: 使用 PCI 配置头对设备地址进行配置 1: 使用固定地址访问设备 如果 fix_default_route 为 1, 那么该信号恒为 1。 该位必须设置为 0。

### 4.3 引脚复用配置寄存器 0

地址偏移: 0440h

默认值: FFFF\_FFFFh



本寄存器包含引脚复用的相关配置信息。

表 4-4 引脚复用配置寄存器 0

位域	名称	访问	描述
31	uart3_enable	R/W	引脚 UART_RI/DCD 的 UART 工作模式选择(当选择为 UART 模式时) 0: 是否使用由 bit28 决定 1: 工作在 UART3 模式
30	uart2_enable	R/W	引脚 UART_DTR/DSR 的 UART 工作模式选择(当选择为 UART 模式时) 0: 是否使用由 bit28 决定 1: 工作在 UART2 模式
29	uart1_enable	R/W	引脚 UART_RTS/CTS 的 UART 工作模式选择(当选择为 UART 模式时) 0: 是否使用由 bit28 决定 1: 工作在两线 UART1 模式
28	uart0_full_func	R/W	引脚 UART_TXD/RXD 的 UART 工作模式选择(当选择为 UART 模式时) 0: 工作在两线 UART0 模式 1: 工作在全功能 UART0 模式 当 bit[31:29] 不等于 0 时, 不管该位的值是什么, UART_TXD/RXD 都工作在两线 UART 模式。
27	clkout_flex_sel	R/W	引脚 CLKOUT_FLEX 的工作模式选择 0: 工作 GPIO 模式 1: 工作 CLKOUT_FLEX 模式
26	clkout_25m_sel	R/W	引脚 CLKOUT_25M 的工作模式选择 0: 工作 GPIO 模式 1: 工作 CLKOUT_25M 模式
25	lpc_sel	R/W	LPC 引脚(LPC_ADO~3/LPC_SERIRQ/LPC_FRAME <sub>n</sub> ) 的工作模式选择 0: 工作在 GPIO 模式 1: 工作在 LPC 模式
24	uart_dtr_dsr_sel_i2c	R/W	引脚 UART_DTR/DSR 的工作模式选择(与 bit20 共同决定) [bit24, bit20]: 00b: 工作在 GPIO 模式 x1b: 工作在 UART 模式(由 bit[31:28] 决定是否工作在全功能模式) 10b: 工作在 I2C 模式
23	uart_ri_dcd_sel_i2c	R/W	引脚 UART_RI/DCD 的工作模式选择(与 bit19 共同决定) [bit23, bit19]: 00b: 工作在 GPIO 模式 x1b: 工作在 UART 模式(由 bit[31:28] 决定是否工作在全功能模式) 10b: 工作在 I2C 模式
22	uart_txd_rxd_sel_uart	R/W	引脚 UART_TXD/RXD 的工作模式选择 0: 工作在 GPIO 模式 1: 工作在 UART 模式(由 bit[31:28] 决定是否工作在全功能模式)
21	uart_rts_cts_sel_uart	R/W	引脚 UART_RTS/CTS 的工作模式选择 0: 工作在 GPIO 模式 1: 工作在 UART 模式(由 bit[31:28] 决定是否工作在全功能模式)

20	uart_dtr_dsr_sel_uart	R/W	引脚 UART_DTR/DSR 的工作模式选择(与 bit24 共同决定) [bit24, bit20]: 00b: 工作在 GPIO 模式 x1b: 工作在 UART 模式(由 bit[31:28] 决定是否工作在全功能模式) 10b: 工作在 I2C 模式
19	uart_ri_dcd_sel_uart	R/W	引脚 UART_RI/DCD 的工作模式选择(与 bit23 共同决定) [bit23, bit19]: 00b: 工作在 GPIO 模式 x1b: 工作在 UART 模式(由 bit[31:28] 决定是否工作在全功能模式) 10b: 工作在 I2C 模式
18	usb_oc3_sel	R/W	引脚 USB_OC3 的工作模式选择 0: 工作在 GPIO 模式 1: 工作在 USB_OC 模式
17	usb_oc2_sel	R/W	引脚 USB_OC2 的工作模式选择 0: 工作在 GPIO 模式 1: 工作在 USB_OC 模式
16	usb_oc1_sel	R/W	引脚 USB_OC1 的工作模式选择 0: 工作在 GPIO 模式 1: 工作在 USB_OC 模式
15	usb_oc0_sel	R/W	引脚 USB_OC0 的工作模式选择 0: 工作在 GPIO 模式 1: 工作在 USB_OC 模式
14	clkout_33m_sel	R/W	引脚 CLKOUT_33M 的工作模式选择 0: 工作 GPIO 模式 1: 工作 CLKOUT_33M 模式
13	sata_ledn_sel	R/W	引脚 SATA_LEDn 的工作模式选择 0: 工作在 GPIO 模式 1: 工作在 SATA 模式
12:11	hda_i2s_sel	R/W	HDA 引脚的工作模式选择。 00b: GPIO 模式 01b: HDA0 模式 10b: I2S 模式 11b: HDA1 模式
10	spi_sel	R/W	引脚 SPI_SCK/SDI/SDO 的工作模式选择 0: 工作在 GPIO 模式 1: 工作在 SPI 模式
9:8	spi_csn_23_sel_spi	R/W	引脚 SPI_CS <sub>n</sub> 2/3 的工作模式选择 00b: 工作在 GPIO 模式 01b: 工作在 SPI 模式 10b: 工作在 I2C 模式 11b: 保留
7:6	spi_csn_01_sel_spi	R/W	引脚 SPI_CS <sub>n</sub> 0/1 的工作模式选择 00b: 工作在 GPIO 模式 01b: 工作在 SPI 模式 10b: 工作在 I2C 模式 11b: 保留
5	i2c1_sel	R/W	引脚 I2C1_SCL/SDA 的工作模式选择 0: 工作在 GPIO 模式 1: 工作在 I2C 模式
4	i2c0_sel	R/W	引脚 I2C0_SCL/SDA 的工作模式选择 0: 工作在 GPIO 模式 1: 工作在 I2C 模式
3	pwm3_sel	R/W	引脚 PWM3 的工作模式选择 0: 工作在 GPIO 模式 1: 工作在 PWM 模式

2	pwm2_sel	R/W	引脚 PWM2 的工作模式选择 0: 工作在 GPIO 模式 1: 工作在 PWM 模式
1	pwm1_sel	R/W	引脚 PWM1 的工作模式选择 0: 工作在 GPIO 模式 1: 工作在 PWM 模式
0	pwm0_sel	R/W	引脚 PWM0 的工作模式选择 0: 工作在 GPIO 模式 1: 工作在 PWM 模式

## 4.4 引脚复用配置寄存器 1

地址偏移: 0444h

默认值: 0000\_0000h

本寄存器包含引脚复用的相关配置信息。

表 4-5 引脚复用配置寄存器 1

位域	名称	访问	描述
31:4	Reserved	R/W	保留
3	can3_sel	R/W	引脚 UART_RI/DCD 的工作模式选择 1b: 工作在 CAN 模式 0b: 工作在非 CAN 模式, 由引脚复用配置寄存器 0 的 bit[31:28] 和 bit[24:19] 决定其工作模式
2	can2_sel	R/W	引脚 UART_DTR/DSR 的工作模式选择 1b: 工作在 CAN 模式 0b: 工作在非 CAN 模式, 由引脚复用配置寄存器 0 的 bit[31:28] 和 bit[24:19] 决定其工作模式
1	can1_sel	R/W	引脚 UART_RTS/CTS 的工作模式选择 1b: 工作在 CAN 模式 0b: 工作在非 CAN 模式, 由引脚复用配置寄存器 0 的 bit[31:28] 和 bit[24:19] 决定其工作模式
0	can0_sel	R/W	引脚 UART_TXD/RXD 的工作模式选择 1b: 工作在 CAN 模式 0b: 工作在非 CAN 模式, 由引脚复用配置寄存器 0 的 bit[31:28] 和 bit[24:19] 决定其工作模式

## 4.5 固定地址配置寄存器

本寄存器用来配置中断控制器和 HPET 模块的固定访问地址（可由 BIOS 更改）。如果不使用该固定地址，操作系统可以通过配置访问获取配置寄存器或者低速 MISC 设备块的基地址，然后加上固定偏移来获得中断控制器和 HPET 模块的访问地址。使用固定地址来访问中断控制器和 HPET 模块可以加快操作系统的中断处理过程。该固定地址由操作系统决定。如果操作系统使用了与本桥片默认设置的固定地址不同的地址，BIOS 需修改本寄存器来与操作系统保持一致。

中断控制器固定地址配置（中断控制器的地址空间大小为 4KB）

地址偏移: 0460h

默认值：5FFF\_F004h

表 4-6 固定地址配置寄存器

位域	名称	访问	描述
63:44	hpet_fix_addr	R/W	固定地址的 31 到 12 位
43:35	Reserved	R/W	保留
34	hpet_fix_addr_en	R/W	使能该固定地址配置
33:32	Reserved	R/W	保留
31:12	apic_fix_addr	R/W	固定地址的 31 到 12 位
11:3	Reserved	R/W	保留
2	apic_fix_addr_en	R/W	使能该固定地址配置
1:0	Reserved	R/W	保留

HPET 固定地址配置（HPET 的地址空间大小为 4KB）

## 4.6 PLL0 配置寄存器

PLL 的具体用法请参考第 2.2.2 节。

该寄存器用来设置 PLL0，其中输出时钟 0 用来控制 USB3/SATA3 的控制器时钟，输出时钟 1 用来产生 GMAC 需要的 125MHz 时钟。

地址偏移：0480h

默认值：0000\_0000h

表 4-7 PLL0 配置寄存器

位域	名称	访问	描述
63:46	Reserved	R/W	保留
45	pll_pd	R/W	PLL powerdown
44	pll_bypass	R/W	PLL 内部 bypass
43	set_pll_param	R/W	设置 PLL 配置参数
42	sel_pll_out2	R/W	选择 PLL 输出时钟 2
41	sel_pll_out1	R/W	选择 PLL 输出时钟 1
40	sel_pll_out0	R/W	选择 PLL 输出时钟 0
39	pll_locked	RO	PLL 锁定
38:32	pll_div_ref	R/W	PLL 输入分频数
31:30	Reserved	R/W	保留
29:21	pll_loopc	R/W	PLL 倍频乘数
20:14	pll_div_out2	R/W	PLL 输出时钟 2 分频数
13:7	pll_div_out1	R/W	PLL 输出时钟 1 分频数
6:0	pll_div_out0	R/W	PLL 输出时钟 0 分频数

## 4.7 PLL1 配置寄存器

PLL 的具体用法请参考第 2.2.2 节。

该寄存器用来设置 PLL1, 其中输出时钟 0 用来作为 DC 的时钟, 输出时钟 1 用来作为 GMEM 的时钟, 输出时钟 2 用来作为 GPU 的时钟。

地址偏移: 0490h

默认值: 0000\_0000h

表 4-8 PLL1 配置寄存器

位域	名称	访问	描述
63:46	Reserved	R/W	保留
45	pll_pd	R/W	PLL powerdown
44	pll_bypass	R/W	PLL 内部 bypass
43	set_pll_param	R/W	设置 PLL 配置参数
42	sel_pll_out2	R/W	选择 PLL 输出时钟 2
41	sel_pll_out1	R/W	选择 PLL 输出时钟 1
40	sel_pll_out0	R/W	选择 PLL 输出时钟 0
39	pll_locked	RO	PLL 锁定
38:32	pll_div_ref	R/W	PLL 输入分频数
31:30	Reserved	R/W	保留
29:21	pll_loopc	R/W	PLL 倍频乘数
20:14	pll_div_out2	R/W	PLL 输出时钟 2 分频数
13:7	pll_div_out1	R/W	PLL 输出时钟 1 分频数
6:0	pll_div_out0	R/W	PLL 输出时钟 0 分频数

## 4.8 PLL2 配置寄存器

PLL 的具体用法请参考第 2.2.2 节。

该寄存器用来设置 PLL2, 其中输出时钟 0 用来产生 HDA 需要的 24MHz bitclk 时钟, 输出时钟 1 用来作为内部总线的时钟, 输出时钟 2 用来作为 CLKOUT\_FLEX 的时钟。

地址偏移: 04A0h

默认值: 0000\_0000h

表 4-9 PLL2 配置寄存器

位域	名称	访问	描述
63:46	Reserved	R/W	保留
45	pll_pd	R/W	PLL powerdown

44	pll_bypass	R/W	PLL 内部 bypass
43	set_pll_param	R/W	设置 PLL 配置参数
42	sel_pll_out2	R/W	选择 PLL 输出时钟 2
41	sel_pll_out1	R/W	选择 PLL 输出时钟 1
40	sel_pll_out0	R/W	选择 PLL 输出时钟 0
39	pll_locked	RO	PLL 锁定
38:32	pll_div_ref	R/W	PLL 输入分频数
31:30	Reserved	R/W	保留
29:21	pll_loopc	R/W	PLL 倍频乘数
20:14	pll_div_out2	R/W	PLL 输出时钟 2 分频数
13:7	pll_div_out1	R/W	PLL 输出时钟 1 分频数
6:0	pll_div_out0	R/W	PLL 输出时钟 0 分频数

## 4.9 PLL\_PIX\_0 配置寄存器

PLL 的具体用法请参考第 2.2.2 节。

该寄存器用来设置 PLL\_PIX\_0，其中输出时钟 0 用来产生 PIX0 时钟。

地址偏移：04B0h

默认值：0000\_0000h

表 4-10 PLL\_PIX\_0 配置寄存器列表

位域	名称	访问	描述
63:46	Reserved	R/W	保留
45	pll_pd	R/W	PLL powerdown
44	pll_bypass	R/W	PLL 内部 bypass
43	set_pll_param	R/W	设置 PLL 配置参数
42:41	Reserved	R/W	保留
40	sel_pll_out0	R/W	选择 PLL 输出时钟 0
39	pll_locked	RO	PLL 锁定
38:32	pll_div_ref	R/W	PLL 输入分频数
31:30	Reserved	R/W	保留
29:21	pll_loopc	R/W	PLL 倍频乘数
20:7	Reserved	R/W	保留
6:0	pll_div_out0	R/W	PLL 输出时钟 0 分频数

## 4.10 PLL\_PIX\_1 配置寄存器

PLL 的具体用法请参考第 2.2.2 节。

该寄存器用来设置 PLL\_PIX\_1，其中输出时钟 0 用来产生 PIX1 时钟。

地址偏移：04C0h

默认值：0000\_0000h

表 4-11 PLL\_PIX\_1 配置寄存器

位域	名称	访问	描述
63:46	Reserved	R/W	保留
45	pll_pd	R/W	PLL powerdown
44	pll_bypass	R/W	PLL 内部 bypass
43	set_pll_param	R/W	设置 PLL 配置参数
42:41	Reserved	R/W	保留
40	sel_pll_out0	R/W	选择 PLL 输出时钟 0
39	pll_locked	RO	PLL 锁定
38:32	pll_div_ref	R/W	PLL 输入分频数
31:30	Reserved	R/W	保留
29:21	pll_loopc	R/W	PLL 倍频乘数
20:7	Reserved	R/W	保留
6:0	pll_div_out0	R/W	PLL 输出时钟 0 分频数

## 4.11 SSCPLL 配置寄存器 0

PLL 的具体用法请参考第 2.2.2 节。

该寄存器用于配置 SSC PLL。

地址偏移：04E0h

默认值：0000\_0000\_0000\_0001h

表 4-12 SSCPLL 配置寄存器 0

位域	名称	访问	描述
61:56	Reserved	R/W	保留
53:48	div_out2	R/W	输出分频系数 2
45:40	div_out1	R/W	输出分频系数 1
37:32	div_out0	R/W	输出分频系数 0
31:24	loopc	R/W	环路分频系数, 可配值范围为 8-200(包含)
18:16	ref_div	R/W	参考时钟分频设置
10:08	odf_disable	R/W	输出分频器禁止, 每位对应一个输出分频器, 高有效
07:06	dither_disable	R/W	07:为高时禁止长方形 PDF dither 06:为高时禁止三角形 PDF dither
05	strb_bypass	R/W	选通信号 bypass, 具体描述参见 strb.

04	strb	R/W	展频或小数分频配置(包括 spread_sel, frac_en, frac_input, mode_period, inc_step)的选通信号。 该位配置的要求如下： 1. 只有 strb_bypass 为 0 时才起作用； 2. 先置高再置低后以上配置信号才生效 3. 只有在 lock 生效后才起作用。 当 strb_bypass 设置为 1 时，以上配置参数只可以在低功耗模式下进行修改。
03	spread_sel	R/W	展频方式选择, 0-中心展频, 1-下展频
02	frac_en	R/W	小数环路分频使能, 高有效
01	sscg_en	R/W	展频使能, 高有效
00	pd	R/W	低功耗模式, 高有效

## 4.12 SSCPLL 配置寄存器 1

PLL 的具体用法请参考第 2.2.2 节。

该寄存器用于配置 SSC PLL。

地址偏移：04E8h

默认值：0000\_0000\_0000\_0000h

表 4-13 SSCPLL 配置寄存器 1

位域	名称	访问	描述
60	lock		PLL 锁定信号, 高有效
58	gpu_clk_sel		GPU 时钟选择设置, 0-选择 PLL2 的输出, 1-选择 SSC PLL 的输出
57	gmem_clk_sel		GMEM 时钟选择设置, 0-选择 PLL2 的输出, 1-选择 SSC PLL 的输出
56	dc_clk_sel		DC 时钟选择设置, 0-选择 PLL2 的输出, 1-选择 SSC PLL 的输出
52:48	cp		电荷泵设置, loopc 设置越大, 该寄存器设置越大
47:32	frac_input		小数环路分频设置
30:16	inc_step		展频深度设置
12:00	mod_period		展频调制周期设置

## 4.13 FREQSCALE 配置寄存器

该寄存器用于配置时钟的分频系数。

地址偏移：04D0h

默认值：FFFF\_FFFFh



表 4-14 FREQSCALE 配置寄存器

位域	名称	访问	描述
31:26	Reserved	R/W	保留
25:23	ht_freqscale	R/W	HT 控制器时钟分频数
22:19	Reserved	R/W	保留
18:16	usb_freqscale	R/W	USB 控制器时钟分频数
15	Reserved	R/W	保留
14:12	sata_freqscale	R/W	SATA 控制器时钟分频数
11:7	Reserved	R/W	保留
6:4	usb2_frescale	R/W	USB2 控制器时钟分频数
2:0	sys_freqscale	R/W	内部总线时钟分频数

Freqscale 分频计算公式为:  $f_{out} = f_{in} * (freqscale + 1) / 8$ .

注:

1. 保留的位域必须保持为 1.

## 4.14 PCIE 配置访问路由控制寄存器

地址偏移: 0638h

默认值: 0000\_0000h

本寄存器用来控制 PCIE 配置访问的路由。当配置访问命中在 PCIE 控制器二级总线 (secondary bus) 上, 但设备号非 0 时, 可通过配置该寄存器禁止该配置访问被转发到二级总线上。每个 PCIE 端口分别有一个配置位, 配置为 0 代表禁止, 配置位与 PCIE 端口具体对应关系如下表:

表 4-15 配置位与 PCIE 端口对应关系

位域	名称	访问	描述
31: 14	-	R/W	保留
13:0	pcie_route	R/W	每个 PCIE 控制器有一个控制位, 对应关系如下: 03:00 对应 PCIE_F0 的 port3-0 05:04 对应 PCIE_F1 的 port1-0 09:08 对应 PCIE_H 的 port1-0 13:12 对应 PCIE_G0 的 port1-0 其他位, 保留

## 4.15 GMAC0 配置寄存器

本组寄存器用来操作 gmac0 以及对应 PHY 的相关配置。

地址偏移: 0770h

默认值: 01c2\_0e0fh

表 4-16 GMAC0 配置寄存器

位域	名称	访问	描述
63:33	Reserved	R/W	保留
32	gmac_reset	R/W	gmac 控制器复位信号, 高有效
31	phy_pll_lock	RO	PHY PLL 锁定信号
30:25	Reserved	R/W	保留
24	phy_resetn	R/W	PHY 复位控制信号, 低有效
23	ref_clk_mode	R/W	参考时钟模式 0: 单端时钟 1: 差分时钟
22	pll_25_125	R/W	PHY 参考时钟频率选择 0: 125M(必须将 ref_clk_mode 设置为 0) 1: 25M(必须将 ref_clk_mode 设置为 1)
21:0	Reserved	R/W	保留

## 4.16 桥片标识寄存器

本寄存器包含桥片的标识 ID。

地址偏移: 3FF8h

默认值: 7a10h

表 4-17 桥片标识寄存器

位域	名称	访问	描述
63:32	variable_id	RO	桥片可变 ID (芯片出厂时设定)
31:20	fix_id	RO	桥片固定 ID (0x7A1)
19:0	variable_id	RO	桥片可变 ID (芯片出厂时设定)

## 5 中断控制器

桥片内部集成了高级可编程中断控制器。桥片内部的中断控制器最多支持 64 个中断源，并支持两路中断输出。桥片支持通过中断线和 HT 消息两种方式传输中断。在中断线方式下，桥片的中断输出引脚连接到处理器的中断输入引脚上，处理器通过中断引脚被中断。在 HT 消息方式下，不需要额外的中断引脚连接，桥片通过 HT 数据包的方式，将中断向量发送到处理器的 HT 控制器寄存器中，处理器通过 HT 控制器中断被中断。桥片只支持使用这两种中断方式中的一种，且对所有 64 个中断源同时生效。

桥片支持输出双路中断，也就是说可以将中断信息路由给两个处理器。中断源和中断输出的对应关系是可配置，这个配置对中断线和 HT 消息包两种中断方式都有效。

在中断线中断模式下，桥片内部所有具有中断功能的设备的中断线直接送给桥片的中断控制器，外接 PCIE 设备通过 legacy 中断的方式，将中断发送到桥片内部的 PCIE 控制器，然后 PCIE 控制器通过中断线将中断信息送给桥片的中断控制器，桥片的中断控制器最终通过桥片的中断引脚信号中断处理器。

在 HT 消息中断模式下，桥片内部除了 PCIE 外的所有其他设备通过中断线将中断信息送给桥片的中断控制器，中断控制器将中断信息转换成 HT 消息包，然后通过 HT 总线发给处理器。对于 PCIE 设备，有两种中断模式，一种是设备仍然使用 legacy 中断，通过桥片的 PCIE 控制器的中断线将中断信息送给桥片的中断控制器，由中断控制器负责将中断信息转换成 HT 消息包；另一种是使能 PCIE 设备的 MSI 中断功能，设备的 MSI 中断消息经过桥片 PCIE 控制器内部的转换模块转换成 HT 消息包。需要注意，在后一种模式下，PCIE 设备的中断消息包不支持中断路由，只能发送给桥片的 HT IO 控制器。

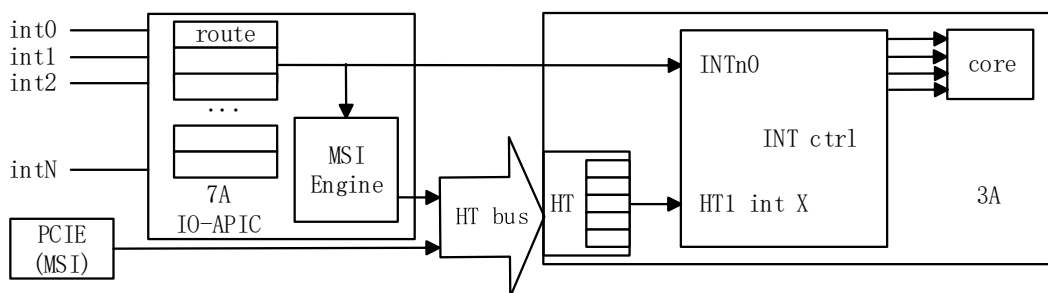


图 5-1 3A+7A 的中断系统示意图

图 5-1 以 3A+7A 计算机系统为例来说明中断涉及的硬件模块及过程。图中表示了两种中断方式的过程，上部表示的是通过中断线 INTn0 来中断，下部表示的是通过 HT 消息包来中断。设备（除了工作在 MSI 模式的 PCIE 设备）发出的中断 intX 送给 7A 内部中断控制器，经过中断路由后送到桥片引脚或者转换成 HT 消息包发给 3A 的 HT 控制器，3A 的中断控制器通过外部中断引脚或者 HT 控制器中断接收到该中断，并经过中断路由中断某个处理器核。

桥片中断控制器的地址空间大小为 4KB，起始地址（桥片内部空间）由 BIOS 配置。

## 5.1 中断源分配

桥片内部设备都通过中断线连接到中断控制器。中断控制器的中断引脚分配见表 5-1。其中，四个 UART 控制器共用一个中断引脚，

6 个 I2C 控制器共用一个中断引脚，

HPET 的三个 timer 共用一个中断引脚，

I2S 中断分为控制器中断和 DMA 中断，它们各占一个中断引脚，

GPIO0~3 分别独占一个中断引脚，其他的 GPIO 共用一个中断引脚。

表 5-1 桥片中断控制器中断源分配

中断引脚	中断源	中断引脚	中断源
0	-	32	pcie_f0_0
1	-	33	pcie_f0_1
2	-	34	pcie_f0_2
3	-	35	pcie_f0_3
4	prg	36	pcie_f1_0
5	sataphy	37	pcie_f1_1
6	usb3phy	38	pcie_h_lo
7	gmac0_phy	39	pcie_h_hi
8	uart[3:0]	40	pcie_g0_lo
9	i2c[5:0]	41	pcie_g0_hi
10	-	42	-
11	can	43	-
12	gmac0_sbd	44	toy[0]
13	gmac0_pmt	45	toy[1]
14	-	46	toy[2]
15	-	47	acpi_int
16	sata	48	usb_0_ehci
17	-	49	usb_0_ohci
18	-	50	usb_1_ehci
19	lpc	51	usb_1_ohci
20	hpet[1]	52	rtc[0]
21	hpet[2]	53	rtc[1]
22	usb3	54	rtc[2]
23	hda1	55	hpet[0]
24	pwm[0]	56	i2s_dma[0]
25	pwm[1]	57	i2s_dma[1]

26	pwm[2]	58	i2s/hda
27	pwm[3]	59	gpio_hi
28	dc	60	gpio[0]/gpio[13]
29	gpu	61	gpio[1]/gpio[14]
30	gmem	62	gpio[2]/gpio[15]
31	thsens	63	gpio[3]/gpio[50]

## 5.2 中断相关寄存器描述

桥片的中断控制器针对每一个中断源，有一套控制和状态寄存器。

表 5-2 中断相关寄存器

寄存器名	位宽	访问类型	说明	默认值
INT_MASK	1	R/W	中断屏蔽寄存器。 0: 使能该中断; 1: 屏蔽该中断。	1
HTMSI_EN	1	R/W	HT 消息包中断使能寄存器。 0: 关闭 HT 消息包方式; 1: 使能 HT 消息包方式。	0
INTEEDGE	1	R/W	触发方式设置寄存器。 0: 电平触发中断; 1: 边沿触发中断。	0
INTCLR	1	WO	脉冲触发中断清除寄存器。 写 1 清除该中断, 写 0 无效。	N/A
SOFT_INT	1	R/W	软件触发中断寄存器。 使用软件来模拟设备中断, 等价于设备的中断输出。	0
ROUTE_ENTRY	8	R/W	中断路由寄存器。 用来配置将该中断路由给哪个处理器, 该寄存器按照位图的形式组织。 bit0: 路由给 INTn0/HT 控制器 lo; bit1: 路由给 INTn1/HT 控制器 hi。 bit7:2: 保留。	0
HTMSI_VECTOR	8	R/W	HT 消息包中断向量寄存器。	见下文
INTISR_CHIPO	1	RO	路由到 INTn0 的中断状态 (在服务) 寄存器。 0: 无中断; 1: 有中断。	0
INTISR_CHIP1	1	RO	路由到 INTn1 的中断状态 (在服务) 寄存器。 0: 无中断; 1: 有中断。	0
INTIRR	1	RO	中断请求寄存器。 0: 没有中断请求; 1: 有中断请求。	0
INTISR	1	RO	中断状态 (在服务) 寄存器。 0: 没有中断被接收; 1: 有中断被接收。	0

INT_POLARITY	1	R/W	中断电平触发极性选择寄存器。 对于电平触发类型： 0：高电平触发； 1：低电平触发。	0
--------------	---	-----	---	---

桥片中断控制器的访问地址是固定的，访问基地址为 0x5fff, f000，地址空间大小为 4KB。中断控制器相关寄存器的地址分布见表 5-3。

表 5-3 中断寄存器地址分布

寄存器名	地址偏移	访问	说明
INT_APIC_ID	0x000	RO	中断控制器标识寄存器
INT_MASK	0x020	R/W	中断屏蔽寄存器
HTMSI_EN	0x040	R/W	HT 消息包中断使能寄存器
INTEDGE	0x060	R/W	触发方式设置寄存器
INTCLR	0x080	WO	脉冲触发中断清除寄存器
SOFT_INT	0x0a0	R/W	软件触发中断寄存器
ROUTE_ENTRY_0	0x100	R/W	中断路由寄存器[ 7- 0]
ROUTE_ENTRY_8	0x108	R/W	中断路由寄存器[15- 8]
ROUTE_ENTRY_16	0x110	R/W	中断路由寄存器[23-16]
ROUTE_ENTRY_24	0x118	R/W	中断路由寄存器[31-24]
ROUTE_ENTRY_32	0x120	R/W	中断路由寄存器[39-32]
ROUTE_ENTRY_40	0x128	R/W	中断路由寄存器[47-40]
ROUTE_ENTRY_48	0x130	R/W	中断路由寄存器[55-48]
ROUTE_ENTRY_56	0x138	R/W	中断路由寄存器[63-56]
HTMSI_VECTOR0	0x200	R/W	HT 中断向量寄存器[ 7- 0]
HTMSI_VECTOR8	0x208	R/W	HT 中断向量寄存器[15- 8]
HTMSI_VECTOR16	0x210	R/W	HT 中断向量寄存器[23-16]
HTMSI_VECTOR24	0x218	R/W	HT 中断向量寄存器[31-24]
HTMSI_VECTOR32	0x220	R/W	HT 中断向量寄存器[39-32]
HTMSI_VECTOR40	0x228	R/W	HT 中断向量寄存器[47-40]
HTMSI_VECTOR48	0x230	R/W	HT 中断向量寄存器[55-48]
HTMSI_VECTOR56	0x238	R/W	HT 中断向量寄存器[63-56]
INTISR_0	0x300	RO	路由到 INTn0 的中断状态（在服务）寄存器
INTISR_1	0x320	RO	路由到 INTn1 的中断状态（在服务）寄存器
INTIRR	0x380	RO	中断请求寄存器
INTISR	0x3a0	RO	中断状态（在服务）寄存器
INT_POLARITY	0x3e0	R/W	中断触发电平选择寄存器

### 中断控制器标识寄存器

地址偏移：000-003h

属性：RO

默认值：07000000h

大小：32 位

位域	名称	访问	描述
31:24	apic_id	RO	中断控制器 ID
23:0	Reserved	RO	保留

地址偏移：004-007h

属性：RO

默认值：003F0001h

大小：32 位

位域	名称	访问	描述
31:24	Reserved	RO	保留
23:16	int_num	RO	支持的中断源个数。实际中断个数等于该字段的值加 1。
15:8	Reserved	RO	保留
7:0	apic_version	RO	中断控制器版本号

### 中断掩码寄存器

地址偏移：020-023h

属性：R/W

默认值：FFFFFFFFh

大小：32 位

位域	名称	访问	描述
31:0	int_mask	R/W	中断掩码寄存器的低 32 位 (bit[31:0])

地址偏移：024-027h

属性：R/W

默认值：FFFFFFFFh

大小：32 位

位域	名称	访问	描述
31:0	int_mask	R/W	中断掩码寄存器的高 32 位 (bit[63:32])

### HT 中断消息包使能寄存器

地址偏移：040-043h

属性：R/W

默认值：00000000h

大小：32 位

位域	名称	访问	描述
31:0	htmsi_en	R/W	HT 中断消息包使能寄存器的低 32 位 (bit[31:0])

地址偏移：044-047h

属性：R/W

默认值：00000000h

大小：32 位

位域	名称	访问	描述
31:0	htmsi_en	R/W	HT 中断消息包使能寄存器的高 32 位 (bit[63:32])

### 中断触发控制寄存器

地址偏移: 060-063h

属性: R/W

默认值: 00000000h

大小: 32 位

位域	名称	访问	描述
31:0	int_edge	R/W	中断触发控制寄存器的低 32 位 (bit[31:0])

地址偏移: 064-067h

属性: R/W

默认值: 00000000h

大小: 32 位

位域	名称	访问	描述
31:0	int_edge	R/W	中断触发控制寄存器的高 32 位 (bit[63:32])

### 中断清除寄存器

地址偏移: 080-083h

属性: WO

默认值: N/A

大小: 32 位

位域	名称	访问	描述
31:0	int_clear	WO	中断清除寄存器的低 32 位 (bit[31:0])

地址偏移: 084-087h

属性: WO

默认值: 00000000h

大小: 32 位

位域	名称	访问	描述
31:0	int_clear	WO	中断清除寄存器的高 32 位 (bit[63:32])

### 软中断寄存器

地址偏移: 0A0-0A3h

属性: R/W

默认值: 00000000h

大小: 32 位

位域	名称	访问	描述
31:0	soft_int	R/W	软中断寄存器的低 32 位 (bit[31:0])

地址偏移: 0A4-0A7h

属性: R/W

默认值: 00000000h

大小: 32 位

位域	名称	访问	描述
31:0	soft_int	R/W	软中断寄存器的高 32 位 (bit[63:32])

### 中断路由配置寄存器

地址偏移: 100-103h

属性: R/W

默认值: 00000000h

大小: 32 位



位域	名称	访问	描述
31:9	Reserved	R/W	保留
1:0	Reserved	R/W	保留

地址偏移：104-107h

属性：R/W

默认值：00000000h

大小：32 位

位域	名称	访问	描述
25:24	gmac0_phy	R/W	GMACO_PHY 中断路由配置寄存器
17:16	usb3phy	R/W	USB3PHY 中断路由配置寄存器
9:8	sataphy	R/W	SATAPHY 中断路由配置寄存器
1:0	prg	R/W	PRG 中断路由配置寄存器

地址偏移：108-10Bh

属性：R/W

默认值：00000000h

大小：32 位

位域	名称	访问	描述
25:24	can_int_route	R/W	CAN 中断路由配置寄存器
17:16	Reserved	R/W	保留
9:8	i2c_int_route	R/W	I2C 中断路由配置寄存器
1:0	uart_int_route	R/W	UART 中断路由配置寄存器

地址偏移：10C-10Fh

属性：R/W

默认值：00000000h

大小：32 位

位域	名称	访问	描述
25:24	Reserved	R/W	保留
17:16	Reserved	R/W	保留
9:8	gmac0_pmt_int_route	R/W	GMACO_PMT 中断路由配置寄存器
1:0	gmac0_sbd_int_route	R/W	GMACO_SBD 中断路由配置寄存器

地址偏移：110-113h

属性：R/W

默认值：00000000h

大小：32 位

位域	名称	访问	描述
25:24	lpc_int_route	R/W	LPC 中断路由配置寄存器
1:0	SATA_int_route	R/W	SATA 中断路由配置寄存器

地址偏移：114-117h

属性：R/W

默认值：00000000h

大小：32 位

位域	名称	访问	描述
25:24	hda1_int_route	R/W	HDA1 中断路由配置寄存器
17:16	usb3_int_route	R/W	USB3 中断路由配置寄存器

9:8	hpet2_int_route	R/W	HPET2 中断路由配置寄存器
1:0	hpet1_int_route	R/W	HPET1 中断路由配置寄存器

地址偏移: 118-11Bh

属性: R/W

默认值: 00000000h

大小: 32 位

位域	名称	访问	描述
25:24	pwm3_int_route	R/W	PWM3 中断路由配置寄存器
17:16	pwm2_int_route	R/W	PWM2 中断路由配置寄存器
9:8	pwm1_int_route	R/W	PWM1 中断路由配置寄存器
1:0	pwm0_int_route	R/W	PWM0 中断路由配置寄存器

地址偏移: 11C-11Fh

属性: R/W

默认值: 00000000h

大小: 32 位

位域	名称	访问	描述
25:24	thsens_int_route	R/W	Thsensor 中断路由配置寄存器
17:16	gmem_int_route	R/W	GMEM 中断路由配置寄存器
9:8	gpu_int_route	R/W	GPU 中断路由配置寄存器
1:0	dc_int_route	R/W	DC 中断路由配置寄存器

地址偏移: 120-123h

属性: R/W

默认值: 00000000h

大小: 32 位

位域	名称	访问	描述
25:24	pcie_f0_p3_int_route	R/W	PCIE_F0 控制器 3 中断路由配置寄存器
17:16	pcie_f0_p2_int_route	R/W	PCIE_F0 控制器 2 中断路由配置寄存器
9:8	pcie_f0_p1_int_route	R/W	PCIE_F0 控制器 1 中断路由配置寄存器
1:0	pcie_f0_p0_int_route	R/W	PCIE_F0 控制器 0 中断路由配置寄存器

地址偏移: 124-127h

属性: R/W

默认值: 00000000h

大小: 32 位

位域	名称	访问	描述
25:24	pcie_h_p1_int_route	R/W	PCIE_H 控制器 1 中断路由配置寄存器
17:16	pcie_h_p0_int_route	R/W	PCIE_H 控制器 0 中断路由配置寄存器
9:8	pcie_f1_p1_int_route	R/W	PCIE_F1 控制器 1 中断路由配置寄存器
1:0	pcie_f1_p0_int_route	R/W	PCIE_F1 控制器 0(或 FC) 中断路由配置寄存器

地址偏移: 128-12Bh

属性: R/W

默认值: 00000000h

大小: 32 位

位域	名称	访问	描述
25:24	pcie_g1_p1_int_route	R/W	PCIE_G1 控制器 1 中断路由配置寄存器

17:16	pcie_g1_p0_int_route	R/W	PCIE_G1 控制器 0 中断路由配置寄存器
9:8	pcie_g0_p1_int_route	R/W	PCIE_G0 控制器 1 中断路由配置寄存器
1:0	pcie_g0_p0_int_route	R/W	PCIE_G0 控制器 0 中断路由配置寄存器

地址偏移：12C-12Fh

属性：R/W

默认值：00000000h

大小：32 位

位域	名称	访问	描述
25:24	acpi_int_route	R/W	ACPI 中断路由配置寄存器
17:16	toy2_int_route	R/W	TOY2 中断路由配置寄存器
9:8	toy1_int_route	R/W	TOY1 中断路由配置寄存器
1:0	toy0_int_route	R/W	TOY0 中断路由配置寄存器

地址偏移：130-133h

属性：R/W

默认值：00000000h

大小：32 位

位域	名称	访问	描述
25:24	usb1_ohci_int_route	R/W	USB1 OHCI 控制器中断路由配置寄存器
17:16	usb1_ehci_int_route	R/W	USB1 EHCI 控制器中断路由配置寄存器
9:8	usb0_ohci_int_route	R/W	USB0 OHCI 控制器中断路由配置寄存器
1:0	usb0_ehci_int_route	R/W	USB0 EHCI 控制器中断路由配置寄存器

地址偏移：134-137h

属性：R/W

默认值：00000000h

大小：32 位

位域	名称	访问	描述
25:24	hpet_int_route	R/W	HPET 中断路由配置寄存器
17:16	rtc2_int_route	R/W	RTC2 中断路由配置寄存器
9:8	rtc1_int_route	R/W	RTC1 中断路由配置寄存器
1:0	rtc0_int_route	R/W	RTC0 中断路由配置寄存器

地址偏移：138-13Bh

属性：R/W

默认值：00000000h

大小：32 位

位域	名称	访问	描述
25:24	gpio_hi_int_route	R/W	GPIO 高位 (bit[56:4]) 中断路由配置寄存器
17:16	i2s/hda_int_route	R/W	I2S/HDA 控制器中断路由配置寄存器
9:8	i2s_dma1_int_route	R/W	I2S DMA1 中断路由配置寄存器
1:0	i2s_dma0_int_route	R/W	I2S DMA0 中断路由配置寄存器

地址偏移：13C-13Fh

属性：R/W

默认值：00000000h

大小：32 位

位域	名称	访问	描述
25:24	gpio3_int_route/gpio50_int_route	R/W	GPI03/50 中断路由配置寄存器
17:16	gpio2_int_route/gpio15_int_route	R/W	GPI02/15 中断路由配置寄存器
9:8	gpio1_int_route/gpio14_int_route	R/W	GPI01/14 中断路由配置寄存器
1:0	gpio0_int_route/gpio13_int_route	R/W	GPI00/13 中断路由配置寄存器

### HT 消息包中断向量配置寄存器

地址偏移: 200-203h                                  属性: R/W  
默认值: 03020100h                                 大小: 32 位

位域	名称	访问	描述
31:8	Reserved	R/W	保留
7: 0	Reserved	R/W	保留

地址偏移: 204-207h                                  属性: R/W  
默认值: 070605040h                                大小: 32 位

位域	名称	访问	描述
31:24	gmac0_phy_int_vector	R/W	GMACO PHY HT 中断向量配置寄存器
23:16	usb3phy_int_vector	R/W	USB3 PHY HT 中断向量配置寄存器
15:8	gsataphy_int_vector	R/W	SATA PHY HT 中断向量配置寄存器
7:0	prg_int_vector	R/W	PRG HT 中断向量配置寄存器

地址偏移: 208-20Bh                                 属性: R/W  
默认值: 0B0A0908h                                大小: 32 位

位域	名称	访问	描述
31:24	can_int_vector	R/W	CAN HT 中断向量配置寄存器
23:16	Reserved	R/W	保留
15:8	i2c_int_vector	R/W	I2C HT 中断向量配置寄存器
7:0	uart_int_vector	R/W	UART HT 中断向量配置寄存器

地址偏移: 20C-20Fh                                 属性: R/W  
默认值: 0E0F0D0Ch                                大小: 32 位

位域	名称	访问	描述
31:24	Reserved	R/W	保留
23:16	Reserved	R/W	保留
15:8	gmac0_pmt_int_vector	R/W	GMACO_PMT HT 中断向量配置寄存器
7:0	gmac0_sbd_int_vector	R/W	GMACO_SBD HT 中断向量配置寄存器

地址偏移: 210-213h                                 属性: R/W  
默认值: 13121110h                                大小: 32 位

位域	名称	访问	描述
31:24	lpc_int_vector	R/W	LPC HT 中断向量配置寄存器
7:0	SATA_int_vector	R/W	SATA HT 中断向量配置寄存器

地址偏移：214-217h

属性：R/W

默认值：17161514h

大小：32 位

位域	名称	访问	描述
31:24	can_int_vector	R/W	CAN HT 中断向量配置寄存器
23:16	Reserved	R/W	保留
15:8	i2c_int_vector	R/W	I2C HT 中断向量配置寄存器
7:0	uart_int_vector	R/W	UART HT 中断向量配置寄存器

地址偏移：218-21Bh

属性：R/W

默认值：1B1A1918h

大小：32 位

位域	名称	访问	描述
31:24	pwm3_int_vector	R/W	PWM3 HT 中断向量配置寄存器
23:16	pwm2_int_vector	R/W	PWM2 HT 中断向量配置寄存器
15:8	pwm1_int_vector	R/W	PWM1 HT 中断向量配置寄存器
7:0	pwm0_int_vector	R/W	PWM0 HT 中断向量配置寄存器

地址偏移：21C-21Fh

属性：R/W

默认值：1E1F1D1Ch

大小：32 位

位域	名称	访问	描述
31:24	thsens_int_vector	R/W	Thsensor HT 中断向量配置寄存器
23:16	gmem_int_vector	R/W	GMEM HT 中断向量配置寄存器
15:8	gpu_int_vector	R/W	GPU HT 中断向量配置寄存器
7:0	dc_int_vector	R/W	DC HT 中断向量配置寄存器

地址偏移：220-223h

属性：R/W

默认值：43424140h

大小：32 位

位域	名称	访问	描述
31:24	pcie_f0_p3_int_vector	R/W	PCIE_F0 控制器 3 HT 中断向量配置寄存器
23:16	pcie_f0_p2_int_vector	R/W	PCIE_F0 控制器 2 HT 中断向量配置寄存器
15:8	pcie_f0_p1_int_vector	R/W	PCIE_F0 控制器 1 HT 中断向量配置寄存器
7:0	pcie_f0_p0_int_vector	R/W	PCIE_F0 控制器 0 HT 中断向量配置寄存器

地址偏移：224-227h

属性：R/W

默认值：47464544h

大小：32 位

位域	名称	访问	描述
31:24	hda1_int_vector	R/W	HDA1 控制器 1 HT 中断向量配置寄存器
23:16	usb3_int_vector	R/W	USB3 控制器 0 HT 中断向量配置寄存器
15:8	hpet2_int_vector	R/W	HPET2 HT 中断向量配置寄存器
7:0	hpet1_int_vector	R/W	HPET1 HT 中断向量配置寄存器

地址偏移：228-22Bh

属性：R/W

默认值：4B4A4948h

大小：32 位

位域	名称	访问	描述
31:24	pcie_g1_p1_int_vector	R/W	PCIE_G1 控制器 1 HT 中断向量配置寄存器
23:16	pcie_g1_p0_int_vector	R/W	PCIE_G1 控制器 0 HT 中断向量配置寄存器
15:8	pcie_g0_p1_int_vector	R/W	PCIE_G0 控制器 1 HT 中断向量配置寄存器
7:0	pcie_g0_p0_int_vector	R/W	PCIE_G0 控制器 0 HT 中断向量配置寄存器

地址偏移：22C-22Fh

属性：R/W

默认值：4F4E4D4Ch

大小：32 位

位域	名称	访问	描述
31:24	acpi_int_vector	R/W	ACPI HT 中断向量配置寄存器
23:16	toy2_int_vector	R/W	TOY2 HT 中断向量配置寄存器
15:8	toy1_int_vector	R/W	TOY1 HT 中断向量配置寄存器
7:0	toy0_int_vector	R/W	TOY0 HT 中断向量配置寄存器

地址偏移：230-233h

属性：R/W

默认值：53525150h

大小：32 位

位域	名称	访问	描述
31:24	usb1_ohci_int_vector	R/W	USB1 OHCI 控制器 HT 中断向量配置寄存器
23:16	usb1_ehci_int_vector	R/W	USB1 EHCI 控制器 HT 中断向量配置寄存器
15:8	usb0_ohci_int_vector	R/W	USB0 OHCI 控制器 HT 中断向量配置寄存器
7:0	usb0_ehci_int_vector	R/W	USB0 EHCI 控制器 HT 中断向量配置寄存器

地址偏移：234-237h

属性：R/W

默认值：57565554h

大小：32 位

位域	名称	访问	描述
31:24	hpet_int_vector	R/W	HPET HT 中断向量配置寄存器
23:16	rtc2_int_vector	R/W	RTC2 HT 中断向量配置寄存器
15:8	rtc1_int_vector	R/W	RTC1 HT 中断向量配置寄存器
7:0	rtc0_int_vector	R/W	RTC0 HT 中断向量配置寄存器

地址偏移：238-23Bh

属性：R/W

默认值：5B5A5958h

大小：32 位

位域	名称	访问	描述
31:24	gpio_hi_int_vector	R/W	GPIO 高位 (bit[56:4]) HT 中断向量配置寄存器
23:16	i2s/hda_int_vector	R/W	I2S/HDA 控制器 HT 中断向量配置寄存器
15:8	i2s_dma1_int_vector	R/W	I2S DMA1 HT 中断向量配置寄存器
7:0	i2s_dma0_int_vector	R/W	I2S DMA0 HT 中断向量配置寄存器

地址偏移: 23C-23Fh

属性: R/W

默认值: 5F5E5D5Ch

大小: 32 位

位域	名称	访问	描述
31:24	gpio3_int_vector/gpio50_int_vector	R/W	GPIO3/50 HT 中断向量配置寄存器
23:16	gpio2_int_vector/gpio15_int_vector	R/W	GPIO2/15 HT 中断向量配置寄存器
15:8	gpio1_int_vector/gpio14_int_vector	R/W	GPIO1/14 HT 中断向量配置寄存器
7:0	gpio0_int_vector/gpio13_int_vector	R/W	GPIO0/13 HT 中断向量配置寄存器

### 路由到 INTn0 的中断在服务状态寄存器

地址偏移: 300-303h

属性: R/W

默认值: 00000000h

大小: 32 位

位域	名称	访问	描述
31:0	int_isr_0	R/W	路由到 INTn0 的中断在服务状态寄存器的低 32 位 (bit[31:0])

地址偏移: 304-307h

属性: R/W

默认值: 00000000h

大小: 32 位

位域	名称	访问	描述
31:0	int_isr_0	R/W	路由到 INTn0 的中断在服务状态寄存器的高 32 位 (bit[63:32])

### 路由到 INTn1 的中断在服务状态寄存器

地址偏移: 320-323h

属性: R/W

默认值: 00000000h

大小: 32 位

位域	名称	访问	描述
31:0	int_isr_1	R/W	路由到 INTn1 的中断在服务状态寄存器的低 32 位 (bit[31:0])

地址偏移: 324-327h

属性: R/W

默认值: 00000000h

大小: 32 位

位域	名称	访问	描述
31:0	int_isr_1	R/W	路由到 INTn1 的中断在服务状态寄存器的高 32 位 (bit[63:32])

### 中断请求寄存器

地址偏移：380-383h 属性：R/W  
默认值：00000000h 大小：32 位

位域	名称	访问	描述
31:0	int_irr	R/W	中断请求寄存器的低 32 位 (bit[31:0])

地址偏移：384-387h 属性：R/W  
默认值：00000000h 大小：32 位

位域	名称	访问	描述
31:0	int_irr	R/W	中断请求寄存器的高 32 位 (bit[63:32])

### 中断在服务状态寄存器

地址偏移：3A0-3A3h 属性：R/W  
默认值：00000000h 大小：32 位

位域	名称	访问	描述
31:0	int_isr	R/W	中断在服务状态寄存器的低 32 位 (bit[31:0])

地址偏移：3A4-3A7h 属性：R/W  
默认值：00000000h 大小：32 位

位域	名称	访问	描述
31:0	int_isr	R/W	中断在服务状态寄存器的高 32 位 (bit[63:32])

### 中断电平触发极性寄存器

地址偏移：3E0-3E3h 属性：R/W  
默认值：00000000h 大小：32 位

位域	名称	访问	描述
31:0	int_polarity	R/W	中断电平触发极性寄存器的低 32 位 (bit[31:0])

地址偏移：3E4-3E7h 属性：R/W  
默认值：00000000h 大小：32 位

位域	名称	访问	描述
31:0	int_polarity	R/W	中断电平触发极性寄存器的高 32 位 (bit[63:32])

## 5.3 设备中断类型

桥片内部设备中断都连接到了桥片的中断控制器上。对于桥片来说，I2S DMA 中断为脉冲触发类型，gpio 中断根据需要可以配置成电平触发或者脉冲触发，其余中断均为电平触发，且高电平有效。



对于 PCIE 设备，一种方式是通过中断线的方式将 PCIE 控制器的中断送给桥片的中断控制器；另一种方式是直接使用 PCIE 设备的 MSI 中断。

在 PCIE MSI 中断方式下，桥片内部的 PCIE 控制器接收到 MSI 中断时，会将它直接转换成 HT 中断消息包发给 HT 控制器。因此，软件需要注意将 PCIE 设备的 MSI 中断向量和桥片内部设备配置的 HT 中断向量区分开来。

## 5.4 中断处理过程详细说明

在中断线中断方式下，桥片的中断控制器接收到设备中断后，会根据中断路由配置将对应的中断输出引脚置低，处理器通过中断输入引脚接收到该中断，处理器通过读取桥片中断控制器内对应的 ISR（中断在服务寄存器）寄存器获得当前路由给自己的中断来源。中断处理过程为：

1. 桥片的中断控制器接收到设备中断；（硬件）
2. 桥片的中断控制器将中断输出引脚置低；（硬件）
3. 处理器的中断控制器接收到中断引脚中断；（硬件）
4. 处理器关中断。（软件）
5. 处理器读取自己的中断控制器得知是外部中断；（软件）
6. 处理器读桥片的中断控制器得到中断向量；（软件）
7. 处理器写桥片的中断控制器关闭对应源的中断使能；（软件）
8. 处理器开中断。（软件）
9. 处理器调用中断服务程序处理中断；（软件）
10. 处理器写桥片的中断控制器清除边沿触发的中断（目前桥片不需要，桥片内部全部为电平触发中断）；（软件）
11. 处理器写桥片的中断控制器打开对应源的中断使能。（软件）
12. 处理器中断返回。（软件）

在 HT 消息包中断方式下，桥片的中断控制器和 PCIE 控制器接收到设备中断后，可以直接将中断向量发送到处理器的 HT 控制器，从而避免了处理器的 HT 控制器查询中断向量的过程。中断处理过程为：

1. 桥片的中断控制器接收到设备中断；（硬件）
2. 桥片的中断控制器将中断向量发送给处理器的 HT 控制器；（硬件）
3. 处理器的中断控制器接收到 HT 中断；（硬件）
4. 处理器关中断。（软件）
5. 处理器读自己的中断控制器得知是 HT 中断；（软件）
6. 处理器读自己的 HT 控制器得到中断向量；（软件）
7. 处理器写自己的 HT 控制器清除中断；（软件）
8. 处理器开中断。（软件）

9. 处理器调用中断服务程序处理中断；（软件）
10. 处理器写桥片的中断控制器清除中断（如果是 PCIE 设备发出的 MSI 中断，则不需要）。  
（软件）
11. 处理器中断返回。（软件）  
软件可以配置桥片内部设备对应的 HT 中断向量。

## 6 温度传感器

龙芯 7A2000 内部集成一个温度传感器，可以通过采样寄存器进行观测，同时内部实现了灵活的高低温中断报警机制。

下面列出温度传感器相关的寄存器以及说明，这些寄存器的基地址与中断控制器一致。

### 6.1 温度传感器配置寄存器

本寄存器用来配置温度传感器的一些控制参数。

地址偏移：0400h

默认值：0000\_0000\_0000\_0001h

表 6-1 温度传感器配置寄存器

位域	名称	访问	描述
63:7	reserved	R/W	保留
6:4	cluster_sel	R/W	采样点选择：0 为本地监测点，对于电压检测可选值有 1-5，对于温度检测可选值有 1-4。
3:2	mode	R/W	工作模式控制： 00-温度采样 10-电压采样 其他-保留
1	rate	R/W	检测速率控制： 0-低速模式(10~20Hz) 1-高速模式(325~650Hz)
0	powerdown	R/W	低功耗控制，为 1 代表进入低功耗模式

### 6.2 温度传感器中断控制寄存器

本寄存器用来对温度传感器中断进行控制。

地址偏移：0408h

默认值：0000\_0000\_0000\_0001h

表 6-2 温度传感器中断控制寄存器

位域	名称	访问	描述
63:41	reserved	R/W	保留
40	low_int_en	R/W	低温中断使能
39:32	temp_low	R/W	低温中断触发温度设置： [7]-温度正负指示，0 代表正值，1 代表负值； [6:0]-摄氏温度值
31:9	reserved	R/W	保留
8	high_int_en	R/W	高温中断使能
7:0	temp_high	R/W	高温中断触发温度设置： [7]-温度正负指示，0 代表正值，1 代表负值； [6:0]-摄氏温度值

## 6.3 温度传感器中断状态/清除寄存器

本寄存器用来对温度传感器中断进行控制。

地址偏移：0410h

默认值：0000\_0000\_0000\_0000h

表 6-3 温度传感器中断状态/清除寄存器

位域	名称	访问	描述
63:56	temperature	RO	摄氏温度值
55	reserved	R/W	保留
54:52	thsens_outcluster	RO	传感器配置的监测点
51:49	reserved	R/W	保留
48	thsens_outmode	RO	传感器配置的监测模式 0：温度模式；1：电压模式
47	reserved	R/W	保留
46	thsens_overflow	R/W	传感器监测值溢出标志
45:32	thsens_data	R/W	传感器的原始数据
31:2	reserved	R/W	保留
1	int_high	R/W1C	中断状态指示，读取时获取高温中断状态，写 1 清零
0	int_low	R/W1C	中断状态指示，读取时获取低温中断状态，写 1 清零

## 7 HT 控制器

桥片 HT 接口最高支持双向 16 位数据宽度及 2.0 GHz 的运行频率。在系统自动初始化建立连接后，用户可以通过修改协议中相应的配置寄存器，实现对宽度和运行频率的更改，并重新进行初始化。

桥片 HT 接口的主要特征如下：

- 支持HT1.0/3.0协议
- 支持8/16位宽度

### 7.1 HT 使用说明

HT 模块内部设置了几个地址窗口用于对 CPU 访问和 DMA 访问进行配置。对于 CPU 访问，桥片作为被访问者，相应的配置窗口称为接收窗口；对于 DMA 访问，桥片作为访问的发起者，相应的窗口称为发送窗口。

接收窗口包括两类：P2P 访问窗口和正常访问窗口。落在 P2P 访问窗口内的访问将作为 P2P 命令直接转发回 HT 总线而不会发给桥片内部设备；落在正常访问窗口内的访问将作为对桥片内部设备的访问发给内部设备。其中，P2P 访问窗口的优先级高于正常访问窗口。对于在这两类接收窗口中都没有命中的访问，则作为 P2P 命令直接转发回 HT 总线。

桥片内部设置了 Post 发送窗口用于将 DMA 访问通过 HT 的 Post 通道发出去。也就是说，在 Post 发送窗口命中的 DMA 访问通过 HT 的 Post 通道发给 HT 总线，不在 Post 发送窗口命中的 DMA 访问通过 HT 的 non-Post 通道发给 HT 总线。

### 7.2 HT 配置寄存器

表 7-1 HT 配置寄存器

Enable	0x00	Device ID		Vendor ID		
	0x04	Status		Command		
	0x08	Class Code			Revision ID	
	0x0c	BIST	Header Type	Latency Timer	Cache Line Size	
	0x10					
	0x14					
	0x18					
	0x1c					
	0x20					
	0x24					
	0x28	Cardbus CIS Pointer				
	0x2c	Subsystem ID		Subsystem Vendor ID		

	0x30	Expansion ROM Enable Address			
	0x34	Reserved			Capabilities Pointer
	0x38	Reserved			
	0x3c	Bridge Control	Interrupt Pin		Interrupt Line
Cap 0 PRI	0x40	Command	Capabilities Pointer		Capability ID
	0x44	Link Config 0	Link Control 0		
	0x48	Link Config 1	Link Control 1		
	0x4C	LinkFreqCap0	Link Error0/Link Freq 0	Revision ID	
	0x50	LinkFreqCap1	Link Error1/Link Freq 1	Feature	
	0x54	Error Handling	Enumeration Scratchpad		
	0x58	Reserved	Mem Limit Upper	Mem Enable Upper	
Cap 1 Retry	0x60	Capability Type	Reserved	Capability Pointer	Capabiliter ID
	0x64	Status 1	Control 1	Status 0	Control 0
	0x68	Retry Count 1		Retry Count 0	
CAP 3	0x6C	Capability Type	Revision ID	Capability Pointer	Capabiliter ID
CAP 4 Interrupt	0x70	Capability Type	Index	Capability Pointer	Capabiliter ID
	0x74	Dataport			
	0x78	IntrInfo[31:0]			
	0x7C	IntrInfo[63:32]			
Int Vector	0x80	INT Vector[31:0]			
	0x84	INT Vector[63:32]			
	0x88	INT Vector[95:64]			
	0x8C	INT Vector[127:96]			
	0x90	INT Vector[159:128]			
	0x94	INT Vector[191:160]			
	0x98	INT Vector[223:192]			
	0x9C	INT Vector[255:224]			
	0xA0	INT Enable[31:0]			
	0xA4	INT Enable[63:32]			
	0xA8	INT Enable[95:64]			
	0xAC	INT Enable[127:96]			
	0xB0	INT Enable[159:128]			
	0xB4	INT Enable[191:160]			
	0xB8	INT Enable[223:192]			
0xBC	INT Enable[255:224]				
CAP 5 Gen3	0xC0	Capability Type	Cap Enum/Index	Capability Pointer	Capabiliter ID
	0xC4	Global Link Training			
	0xC8	Transmitter Configuration 0			

	0xCC	Receiver Configuration 0
	0xD0	Link Training 0
	0xD4	Frequency Extension
	0xD8	Transmitter Configuration 1
	0xDC	Receiver Configuration 1
	0xE0	Link Training 1
	0xE4	BIST Control

Enable	0x100	Device ID	Vendor ID		
	0x104	Status	Command		
	0x108	Class Code			Revision ID
	0x10c	BIST	Header Type	Latency Timer	Cache Line Size
	0x110				
	0x114				
	0x118				
	0x11c				
	0x120				
	0x124				
	0x128	Cardbus CIS Pointer			
	0x12c	Subsystem ID	Subsystem Vendor ID		
	0x130	Expansion ROM Enable Address			
	0x134	Reserved		Capabilities Pointer	
	0x138	Reserved			
	0x13c	Bridge Control	Interrupt Pin	Interrupt Line	
Receive Windows	0x140	HT RX Enable 0			
	0x144	HT RX Mask 0			
	0x148	HT RX Enable 1			
	0x14C	HT RX Mask 1			
	0x150	HT RX Enable 2			
	0x154	HT RX Mask 2			
	0x158	HT RX Enable 3			
	0x15C	HT RX Mask 3			
	0x160	HT RX Enable 4			
	0x164	HT RX Mask 4			
Header Trans	0x168	HT RX Header Trans			
	0x16C	HT RX EXT Header Trans			
Post Windows	0x170	HT TX Post Enable 0			
	0x174	HT TX Post Mask 0			
	0x178	HT TX Post Enable 1			
	0x17C	HT TX Post Mask 1			

Prefetchable Windows	0x180	HT TX Prefetchable Enable 0
	0x184	HT TX Prefetchable Mask 0
	0x188	HT TX Prefetchable Enable 1
	0x18C	HT TX Prefetchable Mask 1
Uncache Windows	0x190	HT RX Uncache Enable 0
	0x194	HT RX Uncache Mask 0
	0x198	HT RX Uncache Enable 1
	0x19C	HT RX Uncache Mask 1
	0x1A0	HT RX Uncache Enable 2
	0x1A4	HT RX Uncache Mask 2
	0x1A8	HT RX Uncache Enable 3
P2P Windows	0x1B0	HT RX P2P Enable 0
	0x1B4	HT RX P2P Mask 0
	0x1B8	HT RX P2P Enable 1
	0x1BC	HT RX P2P Mask 1
APP Config	0x1C0	APP Configuration 0
	0x1C4	APP Configuration 1
	0x1C8	RX Bus Value
	0x1CC	PHY status
Buffer	0x1D0	TX Buffer 0
	0x1D4	TX Buffer 1 / Rx buffer hi
	0x1D8	TX Buffer turning
	0x1DC	RX Buffer lo
Training	0x1E0	Training 0 Counter Short
	0x1E4	Training 0 Counter Long
	0x1E8	Training 1 Counter
	0x1EC	Training 2 Counter
	0x1F0	Training 3 Counter
PLL	0x1F4	PLL Configuration
PHY	0x1F8	IO Configuration
	0x1FC	PHY Configuration

DEBUG	0x240	HT3 DEBUG 0
	0x244	HT3 DEBUG 1
	0x248	HT3 DEBUG 2
	0x24C	HT3 DEBUG 3
	0x250	HT3 DEBUG 4
	0x254	HT3 DEBUG 5
	0x258	HT3 DEBUG 6



POST ID WINDOWS	0x260	HT TX POST ID WIN0
	0x264	HT TX POST ID WIN1
	0x268	HT TX POST ID WIN2
	0x26C	HT TX POST ID WIN3
POST ID WINDOWS	0x270	INT TRANS WIN lo
	0x274	INT TRANS WIN hi

每个寄存器的具体含义如下节如示：

### 7.2.1 Bridge Control

偏移量： 0x3C

复位值： 0x00000000

名称： Bus Reset Control

表 7-2 Bus Reset Control 寄存器定义

位域	位域名称	位宽	复位值	访问	描述
31:23	Reserved	4	0x0		保留
22	Reset	12	0x0	R/W	总线复位控制： 0->1: HT_RSTn 置 0, 总线复位 1->0: HT_RSTn 置 1, 总线解复位
21:0	Reserved	5	0x0		保留

### 7.2.2 Capability Registers

偏移量： 0x40

复位值： 0x20010008

名称： Command, Capabilities Pointer, Capability ID

表 7-3 Command, Capabilities Pointer, Capability ID 寄存器定义

位域	位域名称	位宽	复位值	访问	描述
31:29	Slave/Pri	3	0x0	R	Command 格式为 HOST/Sec
28:26	Reserved	2	0x0	R	保留
25:21	Unit Count	5	0x0	R/W	提供给软件用于记录当前的 Unit 个数
20:16	Unit ID	5	0x0		HOST 模式时: 可用于记录使用 ID 个数 SLAVE 模式时: 记录自身 Unit ID HOST/SLAVE 模式由 act_as_slave 寄存器控制
15:08	Capabilities Pointer	8	0x60	R	下一个 Cap 寄存器偏移地址
7:0	Capability ID	8	0x08	R	HyperTransport capability ID

偏移量： 0x44

复位值： 0x00112000

名称： Link Config, Link Control

表 7-4 Link Config, Link Control 寄存器定义

位域	位域名称	位宽	复位值	访问	描述
30:28	Link Width Out	3	0x0	R/W	发送端宽度
					冷复位后的值为当前连接的最大宽度，写入此寄存器的值将会在下次热复位或是 HT Disconnect 之后生效
					000: 8 位方式
					001: 16 位方式
27	Reserved	1	0x0		保留
26:24	Link Width In	3	0x0	R/W	接收端宽度
					冷复位后的值为当前连接的最大宽度，写入此寄存器的值将会在下次热复位或是 HT Disconnect 之后生效
23	Dw Fc out	1	0x0	R	发送端不支持双字流控
22:20	Max Link Width out	3	0x1	R	HT 总线发送端最大宽度: 16bits
19	Dw Fc In	1	0x0	R	接收端不支持双字流控
18:16	Max Link Width In	3	0x1	R	HT 总线接收端最大宽度: 16bits
15:14	Reserved	2	0x0		保留
13	LDTSTOP#	1	0x1	R/W	当 HT 总线进入 HT Disconnect 状态时，是否关闭 HT PHY
	Tristate Enable				1: 关闭
					0: 不关闭
12:10	Reserved	3	0x0		保留
9	CRC Error (hi)	1	0x0	R/W	高 8 位发生 CRC 错
8	CRC Error (lo)	1	0x0	R/W	低 8 位发生 CRC 错
7	Trans off	1	0x0	R/W	HT PHY 关闭控制
					处于 16 位总线工作方式时
					1: 关闭高/低 8 位 HT PHY 0: 使能低 8 位 HT PHY, 高 8 位 HT PHY 由 bit 0 控制
6	End of Chain	0	0x0	R	HT 总线末端
5	Init Complete	1	0x0	R	HT 总线初始化是否完成
4	Link Fail	1	0x0	R	指示连接失败
3:2	Reserved	2	0x0		保留
1	CRC Flood Enable	1	0x0	R/W	发生 CRC 错误时，是否 flood HT 总线
0	Trans off (hi)	1	0x0	R/W	使用 16 位 HT 总线运行 8 位协议时，高 8 位 PHY 关闭控制
					1: 关闭高 8 位 HT PHY
					0: 使能高 8 位 HT PHY

偏移量: 0x4C

复位值: 0x80250023

名称: Revision ID, Link Freq, Link Error, Link Freq Cap

表 7-5 Revision ID, Link Freq, Link Error, Link Freq Cap 寄存器定义

位域	位域名称	位宽	复位值	访问	描述
31:16	Link Freq Cap	16	0x0000	R	支持的 HT 总线频率，根据外部 PLL 的设置产生不同的值（当使用软件配置 PLL（0x1F4）时，该位无意义） {3.2G, 2.6G, 2.4G, 2.2G, 2.0G, 1.8G, 1.6G, 1.4G, 1.2G, 1.0G, 800M, 600M, 500M, 400M, 300M, 200M}
15:14	Reserved	2	0x0		保留
13	Over Flow Error	1	0x0	R	HT 总线包溢出
12	Protocol Error	1	0x0	R/W	协议错误，指 HT 总线上收到不可识别的命令
11:8	Link Freq	4	0x0	R/W	HT 总线工作频率，写入此寄存器的值后将在下次热复位或是 HT Disconnect 之后生效，设置的值与 Link Freq Cap 的位相对应 (当使用软件配置 PLL（0x1F4）时，该位无意义)
7:0	Revision ID	8	0x60	R/W	版本号：3.0

偏移量： 0x50

复位值： 0x00000002

名称： Feature Capability

表 7-6 Feature Capability 寄存器定义

位域	位域名称	位宽	复位值	访问	描述
31:9	Reserved	25	0x0		保留
8	ExtendedRegister	1	0x0	R	没有
7:4	Reserved	3	0x0		保留
3	ExtendedCTLTime	1	0x0	R	不需要
2	CRCTestMode	1	0x0	R	不支持
1	LDTSTOP#	1	0x1	R	支持 LDTSTOP#
0	IsochronousMode	1	0x0	R	不支持

### 7.2.3 Error Retry 控制寄存器

用于 HyperTransport 3.0 模式下错误重传使能，配置 Short Retry 的最大次数，显示 Retry 计数器是否翻转。

偏移量： 0x64

复位值： 0x00000000

名称： Error Retry 控制寄存器

表 7-7 Error Retry 控制寄存器

位域	位域名称	位宽	复位值	访问	描述
31:10	Reserved	22	0x0	R	保留

9	Retry Count Rollover	1	0x0	R	Retry 计数器计数翻转
8	Reserved	1	0x0	R	保留
7:6	Short Retry Attempts	2	0x0	R/W	允许的最大 Short Retry 次数
5:1	Reserved	5	0x0	R	
0	Link Retry Enable	1	0x0	R/W	出错重连功能使能控制

## 7.2.4 Retry Count 寄存器

用于 HyerTransport 3.0 模式下错误重传计数。

偏移量: 0x68

复位值: 0x00000000

名称: Retry Count 寄存器

表 7-8 Retry Count 寄存器

位域	位域名称	位宽	复位值	访问	描述
31:16	Reserved	12	0x0	R	保留
15:0	Retry Count	16	0x0	R	Retry 计数

## 7.2.5 Revision ID 寄存器

用于配置控制器版本, 配置成新的版本号, 通过 Warm Reset 生效。

偏移量: 0x6C

复位值: 0x00200000

名称: RevisionID 寄存器

表 7-9 Revision ID 寄存器

位域	位域名称	位宽	复位值	访问	描述
31:24	Reserved	8	0x0	R	保留
23:16	Revision ID	8	0x20	R/W	Revision ID 控制寄存器 0x20: HyperTransport 1.00 0x60: HyperTransport 3.00
15:0	Reserved	16	0x0	R	保留

## 7.2.6 Interrupt Discovery & Configuration

偏移量: 0x70

复位值: 0x80000008

名称: Interrupt Capability

表 7-10 Interrupt Capability 寄存器定义

位域	位域名称	位宽	复位值	访问	描述
----	------	----	-----	----	----

位域	位域名称	位宽	复位值	访问	描述
31:24	CapabilitiesPointer	8	0x80	R	Interrupt discovery and configuration block
23:16	Index	8	0x0	R/W	读寄存器偏移地址
15:8	CapabilitiesPointer	8	0x0	R	Capabilities Pointer
7:0	CapabilityID	8	0x08	R	Hypertransport Capablity ID

偏移量: 0x74

复位值: 0x00000000

名称: Dataport

表 7-11 Dataport 寄存器定义

位域	位域名称	位宽	复位值	访问	描述
31:0	Dataport	32	0x0	R/W	当上一寄存器 Index 为 0x10 时, 本寄存器读写结果为 0xa8 寄存器, 否则为 0xac

偏移量: 0x78

复位值: 0xF8000000

名称: IntrInfo[31:0]

表 7-12 IntrInfo 寄存器定义 (1)

位域	位域名称	位宽	复位值	访问	描述
31:24	IntrInfo[31:24]	32	0xF8	R	保留
23:2	IntrInfo[23:2]	22	0x0	R/W	IntrInfo[23:2], 当发出 PIC 中断时, IntrInfo 的值用来表示中断向量
1:0	Reserved	2	0x0	R	保留

偏移量: 0x7c

复位值: 0x00000000

名称: IntrInfo[63:32]

表 7-13 IntrInfo 寄存器定义 (2)

位域	位域名称	位宽	复位值	访问	描述
31:0	IntrInfo[63:32]	32	0x0	R	保留

## 7.2.7 中断向量寄存器

中断向量寄存器共 256 个, 其中除去 HT 总线上的 Fix、Arbiter 以及 PIC 中断直接映射到此 256 个中断向量之中, 其它的中断, 如 SMI, NMI, INIT, INTA, INTB, INTC, INTD 可以通过寄存器 0x50 的 [28:24] 映射到任意一个 8 位中断向量上去, 映射的顺序为 {INTD, INTC, INTB, INTA, 1'b0, INIT, NMI, SMI}。此时中断向量对应值为 {Interrupt Index, 内部向量[2:0]}。

默认情况下可以将 256 位中断分发到 4 位中断线上。在不使用高 8 位 HT 控制器的中断时, 也可以通过设置 ht\_int\_8bit 将 256 位中断分发到 8 位中断线上。

256 个中断向量根据中断路由方式选择寄存器配置的不同映射到不同的中断线上，具体的映射方式为：

表 7-14 中断向量映射方式

中断数	Strip	0	1	2	3	4	5	6	7
4 X = [63:0]	1	[X]	[X+64]	[X+128]	[X+192]	-	-	-	-
	2	[2X]	[2X+1]	[2X+128]	[2X+129]	-	-	-	-
	4	[4X]	[4X+1]	[4X+2]	[4X+3]	-	-	-	-
8 X = [31:0] Y = [63:32]	1	[X]	[Y]	[X+64]	[Y+64]	[X+128]	[Y+128]	[X+192]	[Y+192]
	2	[2X]	[2Y]	[2X+1]	[2Y+1]	[2X+128]	[2Y+128]	[2X+129]	[2Y+129]
	4	[4X]	[4X+32]	[4X+1]	[4X+33]	[4X+2]	[4X+34]	[4X+3]	[4X+35]

以使用 4 位中断线为例，不同的映射方式如下。

ht\_int\_stripe\_1:

- [0, 1, 2, 3……63]对应中断线 0 /HT HI 对应中断线 4
- [64, 65, 66, 67……127]对应中断线 1 /HT HI 对应中断线 5
- [128, 129, 130, 131……191]对应中断线 2 /HT HI 对应中断线 6
- [192, 193, 194, 195……255]对应中断线 3 /HT HI 对应中断线 7

ht\_int\_stripe\_2:

- [0, 2, 4, 6……126]对应中断线 0 /HT HI 对应中断线 4
- [1, 3, 5, 7……127]对应中断线 1 /HT HI 对应中断线 5
- [128, 130, 132, 134……254]对应中断线 2 /HT HI 对应中断线 6
- [129, 131, 133, 135……255]对应中断线 3 /HT HI 对应中断线 7

ht\_int\_stripe\_4:

- [0, 4, 8, 12……252]对应中断线 0 /HT HI 对应中断线 4
- [1, 5, 9, 13……253]对应中断线 1 /HT HI 对应中断线 5
- [2, 6, 10, 14……254]对应中断线 2 /HT HI 对应中断线 6
- [3, 7, 11, 15……255]对应中断线 3 /HT HI 对应中断线 7

以下中断向量的描述对应于 ht\_int\_stripe\_1，另外两种方式可由以上说明得到。

偏移量： 0x80

复位值： 0x00000000

名称： HT 总线中断向量寄存器[31:0]

表 7-15 HT 总线中断向量寄存器 1

位域	位域名称	位宽	复位值	访问	描述
31:0	Interrupt_case[31:0]	32	0x0	R/W	HT 总线中断向量寄存器[31:0]，对应中断线 0/HTHI 对应中断线 4

偏移量: 0x84  
 复位值: 0x00000000  
 名称: HT 总线中断向量寄存器[63:32]

表 7-16 HT 总线中断向量寄存器 2

位域	位域名称	位宽	复位值	访问	描述
31:0	Interrupt_case[63:32]	32	0x0	R/W	HT 总线中断向量寄存器[63:32], 对应中断线 0/HTHI 对应中断线 4

偏移量: 0x88  
 复位值: 0x00000000  
 名称: HT 总线中断向量寄存器[95:64]

表 7-17 HT 总线中断向量寄存器 3

位域	位域名称	位宽	复位值	访问	描述
31:0	Interrupt_case[95:64]	32	0x0	R/W	HT 总线中断向量寄存器[95:64], 对应中断线 1/HTHI 对应中断线 5

偏移量: 0x8c  
 复位值: 0x00000000  
 名称: HT 总线中断向量寄存器[127:96]

表 7-18 HT 总线中断向量寄存器 4

位域	位域名称	位宽	复位值	访问	描述
31:0	Interrupt_case[127:96]	32	0x0	R/W	HT 总线中断向量寄存器[127:96], 对应中断线 1/HTHI 对应中断线 5

偏移量: 0x90  
 复位值: 0x00000000  
 名称: HT 总线中断向量寄存器[159:128]

表 7-19 HT 总线中断向量寄存器 5

位域	位域名称	位宽	复位值	访问	描述
31:0	Interrupt_case[159:128]	32	0x0	R/W	HT 总线中断向量寄存器[159:128], 对应中断线 2/HTHI 对应中断线 6

偏移量: 0x94  
 复位值: 0x00000000  
 名称: HT 总线中断向量寄存器[191:160]

表 7-20 HT 总线中断向量寄存器 6

位域	位域名称	位宽	复位值	访问	描述
31:0	Interrupt_case[191:160]	32	0x0	R/W	HT 总线中断向量寄存器[191:160], 对应中断线 2/HTHI 对应中断线 6

偏移量: 0x98

复位值: 0x00000000  
名称: HT 总线中断向量寄存器[223:192]

表 7-21 HT 总线中断向量寄存器 7

位域	位域名称	位宽	复位值	访问	描述
31:0	Interrupt_case[223:192]	32	0x0	R/W	HT 总线中断向量寄存器[223:192], 对应中断线 3/HTHI 对应中断线 7

偏移量: 0x9c  
复位值: 0x00000000  
名称: HT 总线中断向量寄存器[255:224]

表 7-22 HT 总线中断向量寄存器 8

位域	位域名称	位宽	复位值	访问	描述
31:0	Interrupt_case[255:224]	32	0x0	R/W	HT 总线中断向量寄存器[255:224], 对应中断线 3/HTHI 对应中断线 7

## 7.2.8 中断使能寄存器

中断使能寄存器共 256 个，与中断向量寄存器一一对应。置 1 为对应中断打开，置 0 则为中断屏蔽。

256 个中断向量根据中断路由方式选择寄存器配置的不同映射到不同的中断线上，具体的映射方式为：

ht\_int\_stripe\_1:

- [0, 1, 2, 3……63]对应中断线 0 /HT HI 对应中断线 4
- [64, 65, 66, 67……127]对应中断线 1 /HT HI 对应中断线 5
- [128, 129, 130, 131……191]对应中断线 2 /HT HI 对应中断线 6
- [192, 193, 194, 195……255]对应中断线 3 /HT HI 对应中断线 7

ht\_int\_stripe\_2:

- [0, 2, 4, 6……126]对应中断线 0 /HT HI 对应中断线 4
- [1, 3, 5, 7……127]对应中断线 1 /HT HI 对应中断线 5
- [128, 130, 132, 134……254]对应中断线 2 /HT HI 对应中断线 6
- [129, 131, 133, 135……255]对应中断线 3 /HT HI 对应中断线 7

ht\_int\_stripe\_4:

- [0, 4, 8, 12……252]对应中断线 0 /HT HI 对应中断线 4
- [1, 5, 9, 13……253]对应中断线 1 /HT HI 对应中断线 5
- [2, 6, 10, 14……254]对应中断线 2 /HT HI 对应中断线 6
- [3, 7, 11, 15……255]对应中断线 3 /HT HI 对应中断线 7

以下中断向量的描述对应于 ht\_int\_stripe\_1，另外两种方式可由以上说明得到。



偏移量: 0xa0

复位值: 0x00000000

名称: HT 总线中断使能寄存器[31:0]

表 7-23 HT 总线中断使能寄存器 1

位域	位域名称	位宽	复位值	访问	描述
31:0	Interrupt_mask[31:0]	32	0x0	R/W	HT 总线中断使能寄存器[31:0], 对应中断线 0/HTHI 对应中断线 4

偏移量: 0xa4

复位值: 0x00000000

名称: HT 总线中断使能寄存器[63:32]

表 7-24 HT 总线中断使能寄存器 2

位域	位域名称	位宽	复位值	访问	描述
31:0	Interrupt_mask[63:32]	32	0x0	R/W	HT 总线中断使能寄存器[63:32], 对应中断线 0/HTHI 对应中断线 4

偏移量: 0xa8

复位值: 0x00000000

名称: HT 总线中断使能寄存器[95:64]

表 7-25 HT 总线中断使能寄存器 3

位域	位域名称	位宽	复位值	访问	描述
31:0	Interrupt_mask[95:64]	32	0x0	R/W	HT 总线中断使能寄存器[95:64], 对应中断线 1/HTHI 对应中断线 5

偏移量: 0xac

复位值: 0x00000000

名称: HT 总线中断使能寄存器[127:96]

表 7-26 HT 总线中断使能寄存器 4

位域	位域名称	位宽	复位值	访问	描述
31:0	Interrupt_mask[127:96]	32	0x0	R/W	HT 总线中断使能寄存器[127:96], 对应中断线 1 /HT HI 对应中断线 5

偏移量: 0xb0

复位值: 0x00000000

名称: HT 总线中断使能寄存器[159:128]

表 7-27 HT 总线中断使能寄存器 5

位域	位域名称	位宽	复位值	访问	描述
31:0	Interrupt_mask[159:128]	32	0x0	R/W	HT 总线中断使能寄存器[159:128], 对应中断线 2/HTHI 对应中断线 6

偏移量: 0xb4

复位值: 0x00000000  
名称: HT 总线中断使能寄存器[191:160]

表 7-28 HT 总线中断使能寄存器 6

位域	位域名称	位宽	复位值	访问	描述
31:0	Interrupt_mask[191:160]	32	0x0	R/W	HT 总线中断使能寄存器[191:160], 对应中断线 2/HTHI 对应中断线 6

偏移量: 0xb8  
复位值: 0x00000000  
名称: HT 总线中断使能寄存器[223:192]

表 7-29 HT 总线中断使能寄存器 7

位域	位域名称	位宽	复位值	访问	描述
31:0	Interrupt_mask[223:192]	32	0x0	R/W	HT 总线中断使能寄存器[223:192], 对应中断线 3/HTHI 对应中断线 7

偏移量: 0xbc  
复位值: 0x00000000  
名称: HT 总线中断使能寄存器[255:224]

表 7-30 HT 总线中断使能寄存器 8

位域	位域名称	位宽	复位值	访问	描述
31:0	Interrupt_mask[255:224]	32	0x0	R/W	HT 总线中断使能寄存器[255:224], 对应中断线 3/HTHI 对应中断线 7

## 7.2.9 Link Train 寄存器

HyperTransport 3.0 链路初始化及链路训练控制寄存器。

偏移量: 0xD0  
复位值: 0x00000070  
名称: Link Train 寄存器

表 7-31 Link Train 寄存器

位域	位域名称	位宽	复位值	访问	描述
31:23	Reserved	9	0x0	R	保留
22:21	Transmitter LS select	2	0x0	R/W	发送端在 Disconnected 或 Inactive 状态下的链路状态: 2' b00 LS1 2' b01 LS0 2' b10 LS2 2' b11 LS3
14	Dsiable Cmd Throttling	1	0x0	R/W	在 HyperTransport 3.0 模式下, 默认任意 4 个连续的 DWS 中只能出现一个 Non-info CMD; 1' b0 使能 Cmd Throttling 1' b1 禁用 Cmd Throttling
13:10	Reserved	4	0x0	R	保留

8: 7	Receiver LS select	2	0x0	R/W	接收端在 Disconnected 或 Inactive 状态下的链路状态: 2' b00 LS1 2' b01 LS0 2' b10 LS2 2' b11 LS3
6:4	Long Retry Count	3	0x7	R/W	Long Retry 最大次数
3	Scrambling Enable	1	0x0	R/W	是否使能 Scramble 0: 禁用 Scramble 1: 使能 Scramble
2	8B10B Enable	1	0x0	R/W	是否使能 8B10B 0: 禁用 8B10B 1: 使能 8B10B
1	AC	1	0x0	R	是否检测到 AC mode 0: 没有检测到 AC mode 1: 检测到 AC mode
0	Reserved	1	0x0	R	保留

### 7.2.10 接收地址窗口配置寄存器

HT 控制器中的地址窗口命中公式如下:

$$\text{hit} = (\text{BASE} \& \text{MASK}) == (\text{ADDR} \& \text{MASK})$$

$$\text{addr\_out\_trans} = \text{TRANS\_EN} ? \text{TRANS} | \text{ADDR} \& \sim \text{MASK} : \text{ADDR}$$

$$\text{addr\_out} = \text{Multi\_node\_en} ?$$

$$\text{addr\_out\_trans}[39:37], \text{addr\_out\_trans}[43:40], 3' \text{ b0}, \text{addr\_out}[36:0]:$$

$$\text{addr\_out\_trans};$$

需要说明的是, 配置地址窗口寄存器时, MASK 高位应全为 1, 低位应全为 0。MASK 中 0 的实际位数表示的就是地址窗口的大小。

接收地址窗口的地址为 HT 总线上接收的地址。落在 P2P 窗口内的 HT 地址将作为 P2P 命令转发回 HT 总线, 落在正常接收窗口内且不在 P2P 窗口内的 HT 地址将被发往桥片内, 其它地址的命令将作为 P2P 命令被转发回 HT 总线。

偏移量: 0x140

复位值: 0x00000000

名称: HT 总线接收地址窗口 0 使能 (外部访问)

表 7-32 HT 总线接收地址窗口 0 使能 (外部访问) 寄存器定义

位域	位域名称	位宽	复位值	访问	描述
31	ht_rx_image0_en	1	0x0	R/W	HT 总线接收地址窗口 0, 使能信号
30	ht_rx_image0_trans_en	1	0x0	R/W	HT 总线接收地址窗口 0, 映射使能信号
29	ht_rx_image0_multi_node_en	1	0x0	R/W	HT 总线接收地址窗口 0, 多结点地址映射使能 将地址的 [39:37] 转换到 [46:44]
28	ht_rx_image0_conf_hit_en	1	0x0	R/W	HT 总线接收地址窗口 0, 协议地址命中使能 必须置 0
25:0	ht_rx_image0_trans[49:24]	26	0x0	R/W	HT 总线接收地址窗口 0, 映射后地址的 [49:24], 使能 64 位地址空间时, 高位地址

					通过寄存器 284[29:24] 和 280[31:24] 拼接补充
--	--	--	--	--	------------------------------------

偏移量: 0x144

复位值: 0x00000000

名称: HT 总线接收地址窗口 0 基址 (外部访问)

表 7-33 HT 总线接收地址窗口 0 基址 (外部访问) 寄存器定义

位域	位域名称	位宽	复位值	访问	描述
31:16	ht_rx_image0_base[39:24]	16	0x0	R/W	HT 总线接收地址窗口 0, 地址基址的 [39:24], 使能 64 位地址空间时, 高位基址通过寄存器 280[23:0] 拼接补充
15:0	ht_rx_image0_mask[39:24]	16	0x0	R/W	HT 总线接收地址窗口 0, 地址屏蔽的 [39:24], 使能 64 位地址空间时, 高位掩码通过寄存器 284[23:0] 拼接补充

偏移量: 0x148

复位值: 0x00000000

名称: HT 总线接收地址窗口 1 使能 (外部访问)

表 7-34 HT 总线接收地址窗口 1 使能 (外部访问) 寄存器定义

位域	位域名称	位宽	复位值	访问	描述
31	ht_rx_image1_en	1	0x0	R/W	HT 总线接收地址窗口 1, 使能信号
30	ht_rx_image1_trans_en	1	0x0	R/W	HT 总线接收地址窗口 1, 映射使能信号
29	ht_rx_image1_multi_node_en	1	0x0	R/W	HT 总线接收地址窗口 1, 多结点地址映射使能 将地址的 [39:37] 转换到 [46:44]
28	ht_rx_image1_conf_hit_en	1	0x0	R/W	HT 总线接收地址窗口 1, 协议地址命中使能 必须置 0
25:0	ht_rx_image1_trans[49:24]	26	0x0	R/W	HT 总线接收地址窗口 1, 映射后地址的 [49:24], 使能 64 位地址空间时, 高位地址通过寄存器 288[29:24] 和 28C[31:24] 拼接补充

偏移量: 0x14c

复位值: 0x00000000

名称: HT 总线接收地址窗口 1 基址 (外部访问)

表 7-35 HT 总线接收地址窗口 1 基址 (外部访问) 寄存器定义

位域	位域名称	位宽	复位值	访问	描述
31:16	ht_rx_image1_base[39:24]	16	0x0	R/W	HT 总线接收地址窗口 1, 地址基址的 [39:24], 使能 64 位地址空间时, 高位基址通过寄存器 288[23:0] 拼接补充
15:0	ht_rx_image1_mask[39:24]	16	0x0	R/W	HT 总线接收地址窗口 1, 地址屏蔽的 [39:24], 使能 64 位地址空间时, 高位掩码通过寄存器 28C[23:0] 拼接补充

偏移量: 0x150

复位值: 0x00000000

名称: HT 总线接收地址窗口 2 使能 (外部访问)

表 7-36 HT 总线接收地址窗口 2 使能（外部访问）寄存器定义

位域	位域名称	位宽	复位值	访问	描述
31	ht_rx_image2_en	1	0x0	R/W	HT 总线接收地址窗口 2, 使能信号
30	ht_rx_image2_trans_en	1	0x0	R/W	HT 总线接收地址窗口 2, 映射使能信号
29	ht_rx_image2_multi_node_en	1	0x0	R/W	HT 总线接收地址窗口 2, 多结点地址映射使能 将地址的[39:37]转换到[46:44]
28	ht_rx_image2_conf_hit_en	1	0x0	R/W	HT 总线接收地址窗口 2, 协议地址命中使能 必须置 0
25:0	ht_rx_image2_trans[49:24]	26	0x0	R/W	HT 总线接收地址窗口 2, 映射后地址的 [49:24], 使能 64 位地址空间时, 高位地址 通过寄存器 290[29:24]和 294[31:24]拼接 补充

偏移量: 0x154

复位值: 0x00000000

名称: HT 总线接收地址窗口 2 基址（外部访问）

表 7-37 HT 总线接收地址窗口 2 基址（外部访问）寄存器定义

位域	位域名称	位宽	复位值	访问	描述
31:16	ht_rx_image2_base[39:24]	16	0x0	R/W	HT 总线接收地址窗口 2, 地址基址的[39:24], 使能 64 位地址空间时, 高位基址通过寄存器 290[23:0]拼接补充
15:0	ht_rx_image2_mask[39:24]	16	0x0	R/W	HT 总线接收地址窗口 2, 地址屏蔽的[39:24], 使能 64 位地址空间时, 高位掩码通过寄存器 294[23:0]拼接补充

偏移量: 0x158

复位值: 0x00000000

名称: HT 总线接收地址窗口 3 使能（外部访问）

表 7-38 HT 总线接收地址窗口 3 使能（外部访问）寄存器定义

位域	位域名称	位宽	复位值	访问	描述
31	ht_rx_image3_en	1	0x0	R/W	HT 总线接收地址窗口 3, 使能信号
30	ht_rx_image3_trans_en	1	0x0	R/W	HT 总线接收地址窗口 3, 映射使能信号
29	ht_rx_image3_multi_node_en	1	0x0	R/W	HT 总线接收地址窗口 3, 多结点地址映射使能 将地址的[39:37]转换到[46:44]
28	ht_rx_image3_conf_hit_en	1	0x0	R/W	HT 总线接收地址窗口 3, 协议地址命中使能 必须置 0
25:0	ht_rx_image3_trans[49:24]	26	0x0	R/W	HT 总线接收地址窗口 3, 映射后地址的 [49:24], 使能 64 位地址空间时, 高位地址 通过寄存器 29C[29:24]和 298[31:24]拼接 补充

偏移量: 0x15C

复位值: 0x00000000

名称: HT 总线接收地址窗口 3 基址（外部访问）

表 7-39 HT 总线接收地址窗口 3 基址（外部访问）寄存器定义

位域	位域名称	位宽	复位值	访问	描述
31:16	ht_rx_image3_base[39:24]	16	0x0	R/W	HT 总线接收地址窗口 3, 地址基址的 [39:24], 使能 64 位地址空间时, 高位基址通过寄存器 298[23:0] 拼接补充
15:0	ht_rx_image3_mask[39:24]	16	0x0	R/W	HT 总线接收地址窗口 3, 地址屏蔽的 [39:24], 使能 64 位地址空间时, 高位掩码通过寄存器 29C[23:0] 拼接补充

偏移量: 0x160

复位值: 0x00000000

名称: HT 总线接收地址窗口 4 使能（外部访问）

表 7-40 HT 总线接收地址窗口 4 使能（外部访问）寄存器定义

位域	位域名称	位宽	复位值	访问	描述
31	ht_rx_image4_en	1	0x0	R/W	HT 总线接收地址窗口 4, 使能信号
30	ht_rx_image4_trans_en	1	0x0	R/W	HT 总线接收地址窗口 4, 映射使能信号
29	ht_rx_image4_multi_node_en	1	0x0	R/W	HT 总线接收地址窗口 4, 多结点地址映射使能 将地址的 [39:37] 转换到 [46:44]
28	ht_rx_image4_conf_hit_en	1	0x0	R/W	HT 总线接收地址窗口 4, 协议地址命中使能 必须置 0
25:0	ht_rx_image4_trans[49:24]	26	0x0	R/W	HT 总线接收地址窗口 4, 映射后地址的 [49:24], 使能 64 位地址空间时, 高位地址通过寄存器 2A0[29:24] 和 2A4[31:24] 拼接补充

偏移量: 0x164

复位值: 0x00000000

名称: HT 总线接收地址窗口 4 基址（外部访问）

表 7-41 HT 总线接收地址窗口 4 基址（外部访问）寄存器定义

位域	位域名称	位宽	复位值	访问	描述
31:16	ht_rx_image4_base[39:24]	16	0x0	R/W	HT 总线接收地址窗口 4, 地址基址的 [39:24], 使能 64 位地址空间时, 高位基址通过寄存器 2A0[23:0] 拼接补充
15:0	ht_rx_image4_mask[39:24]	16	0x0	R/W	HT 总线接收地址窗口 4, 地址屏蔽的 [39:24], 使能 64 位地址空间时, 高位掩码通过寄存器 2A4[23:0] 拼接补充

## 7.2.11 配置空间转换寄存器

用于对 HT 的配置空间进行各种转换。

偏移量: 0x168

复位值: 0x00000000

名称: 配置空间扩展地址转换

表 7-42 配置空间扩展地址转换寄存器定义

位域	位域名称	位宽	复位值	访问	描述
31	ht_rx_header_trans_ext	1	0x1	R/W	将配置空间 (0xFD_FE000000) 转换后的地址 type1 标志位由 24 位调整到 28 位, 用于与 EXT HEADER 空间统一
30	ht_rx_header_trans_en	1	0x1	R/W	使能配置空间 (0xFD_FE000000) 的高位地址 ([39:24]) 转换
29:0	ht_rx_header_trans[53:24]	30	0xFE00	R/W	配置空间转换后的高地址 [53:24] (实际只有 [53:25] 可用), 使能 64 位地址空间时, 高位地址通过寄存器 2A8[9:0] 拼接补充

偏移量: 0x16C

复位值: 0x00000000

名称: 扩展地址转换

表 7-43 扩展地址转换寄存器定义

位域	位域名称	位宽	复位值	访问	描述
30	ht_rx_ext_header_trans_en	1	0x0	R/W	使能扩展配置空间 (0xFE_00000000) 的高位地址 ([39:28]) 转换
29:0	ht_rx_ext_header_trans[53:24]	30	0x0	R/W	扩展配置空间转换后的高地址 [53:24] (实际只有 [53:29] 可用), 使能 64 位地址空间时, 高位地址通过寄存器 2A8[25:16] 拼接补充

## 7.2.12 POST 地址窗口配置寄存器

地址窗口命中公式详见 7.2.10 节。

本窗口的地址是 AXI 总线上接收到的地址。落在本窗口的所有写访问将立即在 AXI B 通道返回, 并以 POST WRITE 的命令格式发给 HT 总线。而不在本窗口的写请求则以 NONPOST WRITE 的方式发送到 HT 总线, 并等待 HT 总线响应后再返回 AXI 总线。

偏移量: 0x170

复位值: 0x00000000

名称: HT 总线 POST 地址窗口 0 使能 (内部访问)

表 7-44 HT 总线 POST 地址窗口 0 使能 (内部访问)

位域	位域名称	位宽	复位值	访问	描述
31	ht_post0_en	1	0x0	R/W	HT 总线 POST 地址窗口 0, 使能信号
30	ht_split0_en	1	0x0	R/W	HT 访问拆包使能 (对应于 CPU 核的对外 uncached ACC 操作窗口)
29:23	Reserved	14	0x0		保留
15:0	ht_post0_trans[39:24]	16	0x0	R/W	HT 总线 POST 地址窗口 0, 转译后地址的 [39:24]

偏移量: 0x174

复位值: 0x00000000



名称： HT 总线 POST 地址窗口 0 基址（内部访问）

表 7-45 HT 总线 POST 地址窗口 0 基址（内部访问）

位域	位域名称	位宽	复位值	访问	描述
31:16	ht_post0_base[39:24]	16	0x0	R/W	HT 总线 POST 地址窗口 0，地址基址的 [39:24]，使能 64 位地址空间时，高位基址通过寄存器 2B0[23:0] 拼接补充
15:0	ht_post0_mask[39:24]	16	0x0	R/W	HT 总线 POST 地址窗口 0，地址屏蔽的 [39:24]，使能 64 位地址空间时，高位掩码通过寄存器 2B4[23:0] 拼接补充

偏移量： 0x178

复位值： 0x00000000

名称： HT 总线 POST 地址窗口 1 使能（内部访问）

表 7-46 HT 总线 POST 地址窗口 1 使能（内部访问）

位域	位域名称	位宽	复位值	访问	描述
31	ht_post1_en	1	0x0	R/W	HT 总线 POST 地址窗口 1，使能信号
30	ht_split1_en	1	0x0	R/W	HT 访问拆包使能(对应于 CPU 核的对外 uncached ACC 操作窗口)
29:16	Reserved	14	0x0		保留
15:0	ht_post1_trans[39:24]	16	0x0	R/W	HT 总线 POST 地址窗口 1，转译后地址的 [39:24]

偏移量： 0x17c

复位值： 0x00000000

名称： HT 总线 POST 地址窗口 1 基址（内部访问）

表 7-47 HT 总线 POST 地址窗口 1 基址（内部访问）

位域	位域名称	位宽	复位值	访问	描述
31:16	ht_post1_base[39:24]	16	0x0	R/W	HT 总线 POST 地址窗口 1，地址基址的 [39:24]，使能 64 位地址空间时，高位基址通过寄存器 2B8[23:0] 拼接补充
15:0	ht_post1_mask[39:24]	16	0x0	R/W	HT 总线 POST 地址窗口 1，地址屏蔽的 [39:24]，使能 64 位地址空间时，高位掩码通过寄存器 2BC[23:0] 拼接补充

### 7.2.13 可预取地址窗口配置寄存器

地址窗口命中公式详见 7.2.10 节。

本窗口的地址是 AXI 总线上接收到的地址。落在本窗口的取指指令以及 CACHE 访问才会被发往 HT 总线，其它的取指或 CACHE 访问将不会被发往 HT 总线，而是立即返回，如果是读命令，则会返回相应个数的无效读数据。

偏移量： x180

复位值： 0x00000000

名称： HT 总线可预取地址窗口 0 使能（内部访问）



表 7-48 HT 总线可预取地址窗口 0 使能（内部访问）

位域	位域名称	位宽	复位值	访问	描述
31	ht_prefetch0_en	1	0x0	R/W	HT 总线可预取地址窗口 0, 使能信号
30:23	Reserved	15	0x0		保留
15:0	ht_prefetch0_trans[39:24]	16	0x0	R/W	HT 总线可预取地址窗口 0, 转译后地址的 [39:24]

偏移量: 0x184

复位值: 0x00000000

名称: HT 总线可预取地址窗口 0 基址（内部访问）

表 7-49 HT 总线可预取地址窗口 0 基址（内部访问）

位域	位域名称	位宽	复位值	访问	描述
31:16	ht_prefetch0_base[39:24]	16	0x0	R/W	HT 总线可预取地址窗口 0, 地址基址的 [39:24] 位地址, 使能 64 位地址空间时, 高位基址通过寄存器 2C0[23:0] 拼接补充
15:0	ht_prefetch0_mask[39:24]	16	0x0	R/W	HT 总线可预取地址窗口 0, 地址屏蔽的 [39:24], 使能 64 位地址空间时, 高位掩码通过寄存器 2C4[23:0] 拼接补充

偏移量: 0x188

复位值: 0x00000000

名称: HT 总线可预取地址窗口 1 使能（内部访问）

表 7-50 HT 总线可预取地址窗口 1 使能（内部访问）

位域	位域名称	位宽	复位值	访问	描述
31	ht_prefetch1_en	1	0x0	R/W	HT 总线可预取地址窗口 1, 使能信号
30:23	Reserved	15	0x0		保留
15:0	ht_prefetch1_trans[39:24]	16	0x0	R/W	HT 总线可预取地址窗口 1, 转译后地址的 [39:24]

偏移量: 0x18c

复位值: 0x00000000

名称: HT 总线可预取地址窗口 1 基址（内部访问）

表 7-51 HT 总线可预取地址窗口 1 基址（内部访问）

位域	位域名称	位宽	复位值	访问	描述
31:16	ht_prefetch1_base[39:24]	16	0x0	R/W	HT 总线可预取地址窗口 1, 地址基址的 [39:24], 使能 64 位地址空间时, 高位基址通过寄存器 2C8[23:0] 拼接补充
15:0	ht_prefetch1_mask[39:24]	16	0x0	R/W	HT 总线可预取地址窗口 1, 地址屏蔽的 [39:24], 使能 64 位地址空间时, 高位掩码通过寄存器 2CC[23:0] 拼接补充

## 7.2.14 UNCACHE 地址窗口配置寄存器

地址窗口命中公式详见 7.2.10 节。

本窗口的地址是 HT 总线上接收到的地址。落在本窗口地址的读写命令，将不会被送往 SCACHE，也不会使一级 CACHE 发生失效，而是会被直接送至内存或是其它的地址空间，也即该地址窗口中的读写命令将不会维持 IO 的 CACHE 一致性。该窗口主要针对一些不会在 CACHE 中命中所以可以提高访存效率的操作，如显存的访问等。

偏移量： 0x190

复位值： 0x00000000

名称： HT 总线 Uncache 地址窗口 0 使能（内部访问）

表 7-52 HT 总线 Uncache 地址窗口 0 使能（内部访问）

位域	位域名称	位宽	复位值	访问	描述
31	ht_uncache0_en	1	0x0	R/W	HT 总线 uncache 地址窗口 0，使能信号
30	ht_uncache0_trans_en	1	0x0	R/W	HT 总线 uncache 地址窗口 0，映射使能信号
29	ht_uncache0_multi_node_en	1	0x0	R/W	HT 总线 uncache 接收地址窗口 0，多结点地址映射使能
28	ht_uncache0_conf_hit_en	1	0x0	R/W	HT 总线 uncache 接收地址窗口 0，协议地址命中使能
25:0	ht_uncache0_trans[49:24]	26	0x0	R/W	HT 总线 uncache 地址窗口 0，转译后地址的 [49:24]，使能 64 位地址空间时，高位地址通过寄存器 2D4[29:24] 和 2D0[31:24] 拼接补充

偏移量： 0x194

复位值： 0x00000000

名称： HT 总线 Uncache 地址窗口 0 基址（内部访问）

表 7-53 HT 总线 Uncache 地址窗口 0 基址（内部访问）

位域	位域名称	位宽	复位值	访问	描述
31:16	ht_uncache0_base[39:24]	16	0x0	R/W	HT 总线 uncache 地址窗口 0，地址基址的 [39:24]，使能 64 位地址空间时，高位基址通过寄存器 2D0[23:0] 拼接补充
15:0	ht_uncache0_mask[39:24]	16	0x0	R/W	HT 总线 uncache 地址窗口 0，地址屏蔽的 [39:24]，使能 64 位地址空间时，高位掩码通过寄存器 2D4[23:0] 拼接补充

偏移量： 0x198

复位值： 0x00000000

名称： HT 总线 Uncache 地址窗口 1 使能（内部访问）

表 7-54 HT 总线 Uncache 地址窗口 1 使能（内部访问）

位域	位域名称	位宽	复位值	访问	描述
31	ht_uncache1_en	1	0x0	R/W	HT 总线 uncache 地址窗口 1，使能信号
30	ht_uncache1_trans_en	1	0x0	R/W	HT 总线 uncache 地址窗口 1，映射使能信号
29	ht_uncache1_multi_node_en	1	0x0	R/W	HT 总线 uncache 接收地址窗口 1，多结点地址映射使能
28	ht_uncache1_conf_hit_en	1	0x0	R/W	HT 总线 uncache 接收地址窗口 1，协议地址命中使能

25:0	ht_uncache1_trans[49:24]	26	0x0	R/W	HT 总线 uncache 地址窗口 1, 转译后地址的[49:24], 使能 64 位地址空间时, 高位地址通过寄存器 2DC[29:24] 和 2D8[31:24] 拼接补充
------	--------------------------	----	-----	-----	---

偏移量: 0x19c

复位值: 0x00000000

名称: HT 总线 Uncache 地址窗口 1 基址 (内部访问)

表 7-55 HT 总线 Uncache 地址窗口 1 基址 (内部访问)

位域	位域名称	位宽	复位值	访问	描述
31:16	ht_uncache1_base[39:24]	16	0x0	R/W	HT 总线 uncache 地址窗口 1, 地址基址的 [39:24], 使能 64 位地址空间时, 高位基址通过寄存器 2D8[23:0] 拼接补充
15:0	ht_uncache1_mask[39:24]	16	0x0	R/W	HT 总线 uncache 地址窗口 1, 地址屏蔽的 [39:24], 使能 64 位地址空间时, 高位掩码通过寄存器 2DC[23:0] 拼接补充

偏移量: 0x1A0

复位值: 0x00000000

名称: HT 总线 Uncache 地址窗口 2 使能 (内部访问)

表 7-56 HT 总线 Uncache 地址窗口 2 使能 (内部访问)

位域	位域名称	位宽	复位值	访问	描述
31	ht_uncache2_en	1	0x0	R/W	HT 总线 uncache 地址窗口 2, 使能信号
30	ht_uncache2_trans_en	1	0x0	R/W	HT 总线 uncache 地址窗口 2, 映射使能信号
29	ht_uncache2_multi_node_en	1	0x0	R/W	HT 总线 uncache 接收地址窗口 2, 多结点地址映射使能
28	ht_uncache2_conf_hit_en	1	0x0	R/W	HT 总线 uncache 接收地址窗口 2, 协议地址命中使能
25:0	ht_uncache2_trans[49:24]	26	0x0	R/W	HT 总线 uncache 地址窗口 2, 转译后地址的 [49:24], 使能 64 位地址空间时, 高位地址通过寄存器 2E4[29:24] 和 2E0[31:24] 拼接补充

偏移量: 0x1A4

复位值: 0x00000000

名称: HT 总线 Uncache 地址窗口 2 基址 (内部访问)

表 7-57 HT 总线 Uncache 地址窗口 2 基址 (内部访问)

位域	位域名称	位宽	复位值	访问	描述
31:16	ht_uncache2_base[39:24]	16	0x0	R/W	HT 总线 uncache 地址窗口 2, 地址基址的 [39:24], 使能 64 位地址空间时, 高位基址通过寄存器 2E0[23:0] 拼接补充
15:0	ht_uncache2_mask[39:24]	16	0x0	R/W	HT 总线 uncache 地址窗口 2, 地址屏蔽的 [39:24], 使能 64 位地址空间时, 高位掩码通过寄存器 2 E4[23:0] 拼接补充

偏移量: 0x1A8

复位值： 0x00000000

名称： HT 总线 Uncache 地址窗口 3 使能（内部访问）

表 7-58 HT 总线 Uncache 地址窗口 3 使能（内部访问）

位域	位域名称	位宽	复位值	访问	描述
31	ht_uncache3_en	1	0x0	R/W	HT 总线 uncache 地址窗口 3, 使能信号
30	ht_uncache3_trans_en	1	0x0	R/W	HT 总线 uncache 地址窗口 3, 映射使能信号
29	ht_uncache3_multi_node_en	1	0x0	R/W	HT 总线 uncache 接收地址窗口 3, 多结点地址映射使能
28	ht_uncache3_conf_hit_en	1	0x0	R/W	HT 总线 uncache 接收地址窗口 3, 协议地址命中使能
25:0	ht_uncache3_trans[49:24]	26	0x0	R/W	HT 总线 uncache 地址窗口 3, 转译后地址的 [49:24], 使能 64 位地址空间时, 高位地址通过寄存器 2EC[29:24] 和 2E8[31:24] 拼接补充

偏移量： 0x1AC

复位值： 0x00000000

名称： HT 总线 Uncache 地址窗口 3 基址（内部访问）

表 7-59 HT 总线 Uncache 地址窗口 3 基址（内部访问）

位域	位域名称	位宽	复位值	访问	描述
31:16	ht_uncache3_base[39:24]	16	0x0	R/W	HT 总线 uncache 地址窗口 3, 地址基址的 [39:24], 使能 64 位地址空间时, 高位基址通过寄存器 2E8[23:0] 拼接补充
15:0	ht_uncache3_mask[39:24]	16	0x0	R/W	HT 总线 uncache 地址窗口 3, 地址屏蔽的 [39:24], 使能 64 位地址空间时, 高位掩码通过寄存器 2 EC[23:0] 拼接补充

## 7.2.15 P2P 地址窗口配置寄存器

地址窗口命中公式详见 7.2.10 节。

本窗口的地址是 HT 总线上接收到的地址。落在本窗口地址的读写命令, 直接作为 P2P 命令转发回总线, 相对于正常接收窗口和 Uncache 窗口, 该窗口具有最高优先级。

偏移量： 0x1B0

复位值： 0x00000000

名称： HT 总线 P2P 地址窗口 0 使能（外部访问）

表 7-60 HT 总线 P2P 地址窗口 0 使能（外部访问）寄存器

位域	位域名称	位宽	复位值	访问	描述
31	ht_rx_p2p0_en	1	0x0	R/W	HT 总线 P2P 地址窗口 0, 使能信号
29:0	ht_rx_p2p0_trans[53:24]	16	0x0	R/W	HT 总线 P2P 地址窗口 0, 转译后地址的 [53:24], 使能 64 位地址空间时, 高位地址通过寄存器 2F4[28:24] 和 2F0[28:24] 拼接补充

偏移量: 0x1B4  
 复位值: 0x00000000  
 名称: HT 总线 P2P 地址窗口 0 基址 (外部访问)

表 7-61 HT 总线 P2P 地址窗口 0 基址 (外部访问) 寄存器

位域	位域名称	位宽	复位值	访问	描述
31:16	ht_rx_p2p0_base[39:24]	16	0x0	R/W	HT 总线 P2P 地址窗口 1, 地址基址的 [39:24], 使能 64 位地址空间时, 高位基址通过寄存器 2F0[23:0] 拼接补充
15:0	ht_rx_p2p0_mask[39:24]	16	0x0	R/W	HT 总线 P2P 地址窗口 1, 地址屏蔽的 [39:24], 使能 64 位地址空间时, 高位掩码通过寄存器 2F4[23:0] 拼接补充

偏移量: 0x1B8  
 复位值: 0x00000000  
 名称: HT 总线 P2P 地址窗口 1 使能 (外部访问)

表 7-62 HT 总线 P2P 地址窗口 1 使能 (外部访问) 寄存器

位域	位域名称	位宽	复位值	访问	描述
31	ht_rx_p2p1_en	1	0x0	R/W	HT 总线 P2P 地址窗口 1, 使能信号
29:0	ht_rx_p2p1_trans[53:24]	16	0x0	R/W	HT 总线 P2P 地址窗口 1, 转译后地址的 [53:24], 使能 64 位地址空间时, 高位地址通过寄存器 2FC[28:24] 和 2F8[28:24] 拼接补充

偏移量: 0x1BC  
 复位值: 0x00000000  
 名称: HT 总线 P2P 地址窗口 1 基址 (外部访问)

表 7-63 HT 总线 P2P 地址窗口 1 基址 (外部访问) 寄存器

位域	位域名称	位宽	复位值	访问	描述
31:16	ht_rx_p2p1_base[39:24]	16	0x0	R/W	HT 总线 P2P 地址窗口 1, 地址基址的 [39:24], 使能 64 位地址空间时, 高位基址通过寄存器 2F8[23:0] 拼接补充
15:0	ht_rx_p2p1_mask[39:24]	16	0x0	R/W	HT 总线 P2P 地址窗口 1, 地址屏蔽的 [39:24], 使能 64 位地址空间时, 高位掩码通过寄存器 2FC[23:0] 拼接补充

## 7.2.16 控制器参数配置寄存器

偏移量: 0x1C0  
 复位值: 0x03804321  
 名称: APP CONFIG 0

表 7-64 控制器参数配置寄存器 0

位域	位域名称	位宽	复位值	访问	描述
31:30	Reserved	1	0x0		保留
29	Ldt Stop Gen	1	0x0	R/W	使总线进入 LDT DISCONNECT 模式

					正确的方法是：0 -> 1
28	Ldt Req Gen	1	0x0	R/W	从 LDT DISCONNECT 中唤醒 HT 总线，设置 LDT_REQ_n
					正确的方法是先置 0 再置 1：0 -> 1
					除此之外，直接向总线发出读写请求也可以自动唤醒总线
27	rx sample en	1	0x0	R/W	使能采样输入的 cad 和 ctl1，在 (0x1c8) 寄存器中显示，用于调试
26	Dword Write	1	0x1	R/W	对于 32/64/128/256 位 MEM 写访问，是否采用 Dword Write 命令格式 (Byte Write 方式的写在接收时会转换为 128 位带 MASK 的写)
25	Dword Write cfg	1	0x1	R/W	对于配置空间的写访问，是否采用 Dword Write 命令格式 (Byte Write 方式的写在接收时会转换为 128 位带 MASK 的写)
24	Dword Write IO	1	0x1	R/W	对于 IO 空间的写访问，是否采用 Dword Write 命令格式 (Byte Write 方式的写在接收时会转换为 128 位带 MASK 的写)
23	axi aw resize	1	0x0	RW	是否对 128 位带 MASK 写按 Mask 进行 size 的重新设置
22	Coherent Mode	1	0x0	RW	是否是处理器一致性模式，初始值由 ICCEN 引脚决定，复位后生效
21	Coherent_split	1	0x0	RW	一致性模式下，将所有包拆为 32byte 处理
20	Not Care Seqid	1	0x0	R/W	接收端是否不关心 HT 序关系
19:16	Reserved	4	0x0	R/W	保留
15:12	Priority Nop	4	0x4	R/W	HT 总线 Nop 流控包优先级
11:8	Priority NPC	4	0x3	R/W	Non Post 通道读写优先级
7:4	Priority RC	4	0x2	R/W	Response 通道读写优先级
3:0	Priority PC	4	0x1	R/W	Post 通道读写优先级
					0x0：最高优先级
					0xF：最低优先级
					对于各个通道的优先级均采用根据时间变化提高的优先级策略，该组寄存器用于配置各个通道的初始优先级

偏移量： 0x1C4

复位值： 0x70a30800

名称： APP CONFIG1

表 7-65 控制器参数配置寄存器 1

位域	位域名称	位宽	复位值	访问	描述
31	tx post split en	1	0x0	R/W	使能 tx post ID 窗口命中时的写拆包功能(所有跨越 32 字节边界的写请求会被拆为两个连续的写请求 (byte write))
30	tx wr passPW pc	1	0x0	R/W	将所有发出的 Post 通道写请求的 passPW 位设置为 1
29	tx wr passPW npc	1	0x0	R/W	将所有发出的 Nonpost 通道写请求的 passPW 位设置为 1
28	tx rd passPW	1	0x0	R/W	将所有发出的读请求的 passPW 位设置为 1
27	stop same id wr	1	0x0	R/W	发送端遇到相同 AXI ID 的写请求时，停止发送直至前一个同 ID 请求返回

26	stop same id rd	1	0x0	R/W	发送端遇到相同 AXI ID 的请求时，停止发送直至前一个同 ID 请求返回
25:24	Reserved	2	0x0	R/W	保留
23	Wait idle wr int	1	0x1	R/W	发送 int 请求时是否等待所有写请求已经响应完成，为 1 表示等待
22	Tx wr passPW pc int	1	0x0	R/W	配置发送端 Int 命令的 passPW 位
21	act as slave	1	0x1	R/W	设置 SLAVE 模式
20	Host hide	1	0x0	R/W	禁止接收端对配置寄存器空间的访问
19:16	Rrequest delay	4	0x3	R/W	用于在一致性模式下，控制 Rrequest 传输的随机延迟范围
					000: 0 延迟
					001: 随机延迟 0-8
					010: 随机延迟 8-15
					011: 随机延迟 16-31
					100: 随机延迟 32-63
					101: 随机延迟 64-127
					110: 随机延迟 128-255
111: 0 延迟					
15	Crc Int en	1	0x0	R/W	使能 CRC 错时的中断发送
14:12	Crc Int route	3	0x0	R/W	CRC 中断时的中断引脚选择
11	p wr seqid rel en	1	0x1	R/W	post 写完成是否需要等待一个固定时间，置 1 表示需要等待，等待时间由 25C 寄存器设置，为 0 表示认为 post 写立即完成
10	ht int 8 bit	1	0x0	R/W	使用 8 根中断线
9:8	ht_int_stripe	2	0x0	R/W	对应于 3 种中断路由方式，具体描述见中断向量寄存器
					0x0: ht_int_stripe_1
					0x1: ht_int_stripe_2
0x2: ht_int_stripe_4					
4:0	Interrupt Index	5	0x0	R/W	将除了标准中断之外的其它中断重定向到哪个中断向量中（包括 SMI, NMI, INIT, INTA, INTB, INTC, INTD）
					总共 256 个中断向量，本寄存器表示的是中断向量的高 5 位，内部中断向量如下：
					000: SMI
					001: NMI
					010: INIT
					011: Reserved
					100: INTA
					101: INTB
110: INTC					
111: INTD					



## 7.2.17 接收诊断寄存器

偏移量： 0x1C8

复位值： 0x00000000

名称： 接收诊断寄存器

表 7-66 接收诊断寄存器

位域	位域名称	位宽	复位值	访问	描述
31:16	rx_cad_phase_0	24	0x0	R/W	保存采样得到的输入 CAD[15:0] 的值
15:8	rx_ctl_catch	24	0x0	R/W	保存采样得到的输入 ctl (0、2、4、6) 对应 CTL0 采样的四个相位 (1、3、5、7) 对应 CTL1 采样的四个相位
7:0					

## 7.2.18 PHY 状态寄存器

用于观测 PHY 相关的状态，调试使用

偏移量： 0x1CC

复位值： 0x00000000

名称： PHY 状态寄存器

表 7-67 PHY 状态寄存器

位域	位域名称	位宽	复位值	访问	描述
31:29	Reserved	3	0x0	R	保留
28	dll locked hi	1	0x0	R	
27	dll locked lo	1	0x0	R	
26	cdr locked hi	1	0x0	R	
25	cdr locked lo	1	0x0	R	
24	phase locked	1	0x0	R	
23:20	phy state	4	0x0	R	
19:17	tx training status	3	0x0	R	
16:14	rx training status	3	0x0	R	
13:8	Init done	6	0x0	R	
7:0	Reserved	8		R	

## 7.2.19 软件频率配置寄存器

用于实现控制器在工作过程中切换到任意协议和 PLL 支持的链路频率及控制器频率；

具体切换方法为：在使能软件配置模式的前提下，置位软件频率配置寄存器第 1 位，并写入新的时钟相关的参数，包括决定 PLL 输出频率的 div\_refc 和 div\_loop，链路上的



分频系数 phy\_hi\_div 和 phy\_lo\_div，以及控制器的分频系数 core\_div。之后进入 warm reset 或 LDT disconnect，控制器将会自动复位 PLL，配置新的时钟参数。

PHY\_LINK\_CLK 为 HT 总线频率。

时钟频率的计算公式为：

HyperTransport 1.0:

$$\text{PHY\_LINK\_CLK} = 50\text{MHz} \times \text{div\_loop} / \text{div\_refc} / \text{phy\_div}$$

HyperTransport 3.0:

$$\text{PHY\_LINK\_CLK} = 100\text{MHz} \times \text{div\_loop} / \text{div\_refc} / \text{phy\_div}$$

偏移量： 0x1F4

复位值： 0x00000000

名称： 软件频率配置寄存器

表 7-68 软件频率配置寄存器

位域	位域名称	位宽	复位值	访问	描述
31:27	PLL relock counter	5	0x0	R/W	计数器上限配置寄存器，当置位 counter select 时，计数器计数上限为 {PLL_relock_counter, 5' h1f}，否则计数上限为 10' 3ff
26	Counter select	1	0x0	R/W	锁定计时器自定义使能： 1' b0 使用默认计数上限； 1' b1 由 PLL_relock_counter 计算得出
25:22	Soft_phy_lo_div	4	0x0	R/W	低位 PHY 分频系数
21:18	Soft_phy_hi_div	4	0x0	R/W	高位 PHY 分频系数
17:16	Soft_div_refc	2	0x0	R/W	PLL 内分频系数
15: 9	Soft_div_loop	7	0x0	R/W	PLL 内倍频系数
8: 5	Soft_core_div	4	0x0	R/W	控制器时钟分频系数
4	Reserved	1	0x0	R	保留
3	Pll lock	1	0x0	R	PHY 内部 PLL 锁定标志
2	Bypass ht core	1	0x0	R/W	使能旁路参考时钟作为控制器时钟
1	Soft cofig enable	1	0x0	R/W	软件配置频率使能位 1' b0 禁用软件频率配置 1' b1 使能软件频率配置
0	Reserved	1	0x0	R	保留

## 7.2.20 PHY 阻抗匹配控制寄存器

用于控制 PHY 的阻抗匹配使能，发送端和接收端阻抗匹配参数设置

偏移量： 0x1F8

复位值： 0xC87F0000

名称： PHY 阻抗匹配控制寄存器

表 7-69 阻抗匹配控制寄存器

位域	位域名称	位宽	复位值	访问	描述
31	Tx_scanin_en	1	0x0	R/W	TX 阻抗匹配使能
30	Rx_scanin_en	1	0x0	R/W	RX 阻抗匹配使能
27:24	Tx_scanin_ncode	4	0x0	R/W	TX 阻抗匹配扫描输入 ncode
23:20	Tx_scanin_pcode	4	0x0	R/W	TX 阻抗匹配扫描输入 pcode
19:12	Rx_scanin_code	8	0x0	R/W	RX 阻抗匹配扫描输入

### 7.2.21 PHY 配置寄存器

用于配置 PHY 相关的物理参数，当控制器做为两个独立的 8bit 控制器时，高位的 PHY 和低位的 PHY 分别由两个控制器独立控制；当控制器作为 1 个 16bit 的控制器时，高位和低位的 PHY 的配置参数由低位控制器统一控制；

偏移量： 0x1FC

复位值： 0x85308000

名称： PHY 配置寄存器

表 7-70 PHY 配置寄存器

位域	位域名称	位宽	复位值	访问	描述
31	Tx_ckpll_term	1	0x1	R/W	PLL 到 TX 端片上传输线终端阻抗
30	Rx_ckpll_term	1	0x0	R/W	PLL 到 RX 端片上传输线终端阻抗
29	Rx_clk_in_sel	1	0x0	R/W	时钟 PAD 供给数据 PAD 的时钟选择，HT1 模式下自动选择为 CLKPAD： 1' b0 外来时钟源 1' b1 PLL 时钟
28	Rx_ck_dll_sell	1	0x0	R/W	用来锁定 DLL 的时钟选择： 1' b0 PLL 时钟 1' b1 外来时钟源
27:26	Rx_ctle_bitc	2	0x0	R/W	PAD EQD 高频增益
25:24	Rx_ctle_bitr	2	0x3	R/W	PAD EQD 低频增益
23:22	Rx_ctle_bitlim	2	0x0	R/W	PAD EQD 补偿限制
21	Rx_en_ldo	1	0x1	R/W	LDO 控制 1' b0 LDO 禁用 1' b1 LDO 使能
20	Rx_en_by	1	0x1	R/W	BandGap 控制 1' b0 BandGap 禁用 1' b1 BandGap 使能
19:17	Reserved	3	0x0	R	保留
16:12	Tx_preenmp	5	0x08	R/W	PAD 预加重控制信号
11: 0	Reserved	12	0x0	R	保留

## 7.2.22 链路初始化调试寄存器

用于配置在 HyperTransport 3.0 模式下，链路初始化过程中是否使用 PHY 提供的 CDR Lock 信号做为链路 CDR 完成的标志；如果忽略该锁定信号，则需要控制器计数等待一段时间后默认 CDR 完成。

偏移量： 0x240

复位值： 0x00000000

名称： 链路初始化调试寄存器

表 7-71 链路初始化调试寄存器

位域	位域名称	位宽	复位值	访问	描述
15	Cdr_ignore_enable	1	0x0	R/W	链路初始化时是否忽略 CRC lock，通过计数器计数完成等待： 1' b0 等待 CDR lock 1' b1 忽略 CDR lock 信号，通过计数器累加等待
14: 0	Cdr_wait_counter	15	0x0	R/W	等待计数器计数上限，基于控制器时钟完成计数

## 7.2.23 LDT 调试寄存器

软件改变控制器频率后，会导致对 LDT reconnect 阶段计时不准确，需配置该计数器，作为软件配置频率后，LDT 信号无效到控制器开始链路初始化之间的时间，该计时基于控制器时钟。

偏移量： 0x244

复位值： 0x00000000

名称： LDT 调试寄存器 1

表 7-72 LDT 调试寄存器 1

位域	位域名称	位宽	复位值	访问	描述
31:16	Rx_wait_time	16	0x0	R/W	RX 端等待计数器的初值
15:0	Tx_wait_time	16	0x0	R/W	TX 端等待计数器的初值

偏移量： 0x248

复位值： 0x00000000

名称： LDT 调试寄存器 2

表 7-73 LDT 调试寄存器 2

位域	位域名称	位宽	复位值	访问	描述
31:30	Reserved	16	0x0	R/W	
29:0	rx_lane_ts 0	16	0x0	R/W	Gen3 模式下 lane0-5 链路训练状态

偏移量： 0x24C

复位值： 0x00000000  
名 称： LDT 调试寄存器 3

表 7-74 LDT 调试寄存器 3

位域	位域名称	位宽	复位值	访问	描述
31:30	Reserved	16	0x0	R/W	
29:0	rx lane ts 1	16	0x0	R/W	Gen3 模式下 lane6-11 链路训练状态

偏移量： 0x250  
复位值： 0x00000000  
名 称： LDT 调试寄存器 4

表 7-75 LDT 调试寄存器 4

位域	位域名称	位宽	复位值	访问	描述
31:30	Reserved	16	0x0	R/W	
29:0	rx lane ts 2	16	0x0	R/W	Gen3 模式下 lane12-17 链路训练状态

偏移量： 0x254  
复位值： 0x00000000  
名 称： LDT 调试寄存器 5

表 7-76 LDT 调试寄存器 5

位域	位域名称	位宽	复位值	访问	描述
31:22	Reserved	10	0x0	R/W	
21:18	wait ctl	4	0x0	R/W	为对齐 ctl 所在 lane 相位，接收端同频 fifo 阻塞拍数，每 2bit 对应一个 lane
17:0	phase lock	18	0x0	R/W	训练阶段每个 lane 数据相位是否锁定

偏移量： 0x258  
复位值： 0x00000000  
名 称： LDT 调试寄存器 5

表 7-77 LDT 调试寄存器 5

位域	位域名称	位宽	复位值	访问	描述
31:0	wait cad	32	0x0	R/W	为对齐 cad 所在 lane 相位，接收端同频 fifo 阻塞拍数，每 2bit 对应一个 lane

偏移量： 0x25C  
复位值： 0x000000ff  
名 称： post 请求响应完成计数寄存器

表 7-78 post 请求响应完成计数寄存器

位域	位域名称	位宽	复位值	访问	描述
31:0	p_wr_seqid_rel_cnt	32	0xff	R/W	发送端 post 请求响应完成等待时间

### 7.2.24 HT TX POST ID 窗口配置寄存器

该窗口通过将内部写请求的 ID 与预设的窗口相比较，将命中的请求通过 HT POST 通道向外发出。

偏移量： 0x260

复位值： 0x00000000

名称： HT TX POST ID WIN0

表 7-79 HT TX POST ID WIN0

位域	位域名称	位宽	复位值	访问	描述
31:16	HT TX POST ID0 MASK	16	0x0	R/W	AXI ID 命中的请求使用 POST 窗口进行传输，ID 的 MASK 位
15:0	HT TX POST ID0 BASE	16	0x0	R/W	AXI ID 命中的请求使用 POST 窗口进行传输，ID 的 BASE 位

偏移量： 0x264

复位值： 0x00000000

名称： HT TX POST ID WIN1

表 7-80 HT TX POST ID WIN1

位域	位域名称	位宽	复位值	访问	描述
31:16	HT TX POST ID1 MASK	16	0x0	R/W	AXI ID 命中的请求使用 POST 窗口进行传输，ID 的 MASK 位
15:0	HT TX POST ID1 BASE	16	0x0	R/W	AXI ID 命中的请求使用 POST 窗口进行传输，ID 的 BASE 位

偏移量： 0x268

复位值： 0x00000000

名称： HT TX POST ID WIN2

表 7-81 HT TX POST ID WIN2

位域	位域名称	位宽	复位值	访问	描述
31:16	HT TX POST ID2 MASK	16	0x0	R/W	AXI ID 命中的请求使用 POST 窗口进行传输，ID 的 MASK 位
15:0	HT TX POST ID2 BASE	16	0x0	R/W	AXI ID 命中的请求使用 POST 窗口进行传输，ID 的 BASE 位

偏移量： 0x26C

复位值： 0x00000000

名称： HT TX POST ID WIN3

表 7-82 HT TX POST ID WIN3

位域	位域名称	位宽	复位值	访问	描述
31:16	HT TX POST ID3 MASK	16	0x0	R/W	AXI ID 命中的请求使用 POST 窗口进行传输, ID 的 MASK 位
15:0	HT TX POST ID3 BASE	16	0x0	R/W	AXI ID 命中的请求使用 POST 窗口进行传输, ID 的 BASE 位

## 7.2.25 外部中断转换配置

该设置将 HT 收到的中断转换为对某个特定地址的写操作, 直接写入芯片内部的扩展 IO 中断向量, 而不是在 HT 控制器内部产生中断。采用这种方式, 可以直接 IO 中断的直接跨片分发等高级功能。

偏移量: 0x270

复位值: 0x00000000

名称: HT RX INT TRANS Lo

表 7-83 HT RX INT TRANS LO

位域	位域名称	位宽	复位值	访问	描述
31:4	INT_trans_addr[31:4]	28	0x0	R/W	中断转换地址低位
3:0	Reserved	4	0x0	R	保留

偏移量: 0x274

复位值: 0x00000000

名称: HT RX INT TRANS Hi

表 7-84 HT RX INT TRANS Hi

位域	位域名称	位宽	复位值	访问	描述
31	INT_trans_en	1	0x0	R/W	中断转换使能
30	INT_trans_allow	1	0x0	R/W	中断转换使能允许 设置该位后, INT_trans_en 或者芯片的 EXT_INT_en 才可生效。
29:26	INT_trans_cache	4	0x0	R/W	中断转换 Cache 域
25:0	INT_trans_addr[57:32]	27	0x0	R/W	中断转换地址高位

## 7.2.26 扩展地址寄存器

该寄存器配置 HT 工作在扩展地址模式, 复位生效。

偏移量: 0x354

复位值: 0x00000000

名称: 扩展地址寄存器

表 7-85 扩展地址寄存器

位域	位域名称	位宽	复位值	访问	描述
31:5	Reserved	27	0x0	R/W	保留
4	seqid_ext_en	1	0x0	R/W	seqid扩展使能,使能后将使用unitid的第0位作为seqid的最高位
3	rx_ext_addr_high_pfm	1	0x0	R/W	接收端高性能模式使能,使能后将自动恢复省略的高位地址
2	rx_ext_addr	1	0x0	R/W	接收端扩展地址模式使能
1	tx_ext_addr_high_pfm	1	0x0	R/W	发送端高性能模式使能,使能后将自动省略相同的高位地址
0	tx_ext_addr	1	0x0	R/W	发送端扩展地址模式使能

## 7.2.27 Unitid 控制寄存器

该设置根据设备 id 将 HT 发送的请求配置为固定 unitid。

偏移量: 0x360

复位值: 0x00000000

名称: unitid 控制寄存器 0

表 7-86 unitid 控制寄存器 0

位域	位域名称	位宽	复位值	访问	描述
31:21	Reserved	11	0x0	R/W	保留
23:16	unit_id0	5	0x0	R/W	0 号窗口固定 unitid
15:8	axi2unitid0_mask	8	0x0	R/W	0 号窗口设备 id 掩码
7:0	axi2unitid0_base	8	0x0	R/W	0 号窗口设备 id 基址

偏移量: 0x364

复位值: 0x00000000

名称: unitid 控制寄存器 1

表 7-87 unitid 控制寄存器 1

位域	位域名称	位宽	复位值	访问	描述
31:21	Reserved	11	0x0	R/W	保留
23:16	unit_id1	5	0x0	R/W	1 号窗口固定 unitid
15:8	axi2unitid1_mask	8	0x0	R/W	1 号窗口设备 id 掩码
7:0	axi2unitid1_base	8	0x0	R/W	1 号窗口设备 id 基址

偏移量: 0x368

复位值: 0x00000000

名称: unitid 控制寄存器 2

表 7-88 unitid 控制寄存器 2

位域	位域名称	位宽	复位值	访问	描述
31:21	Reserved	11	0x0	R/W	保留
23:16	unit_id2	5	0x0	R/W	2号窗口固定 unitid
15:8	axi2unitid2_mask	8	0x0	R/W	2号窗口设备 id 掩码
7:0	axi2unitid2_base	8	0x0	R/W	2号窗口设备 id 基址

偏移量: 0x36C

复位值: 0x00000000

名称: unitid 控制寄存器 3

表 7-89 unitid 控制寄存器 3

位域	位域名称	位宽	复位值	访问	描述
31:21	Reserved	11	0x0	R/W	保留
23:16	unit_id3	5	0x0	R/W	3号窗口固定 unitid
15:8	axi2unitid3_mask	8	0x0	R/W	3号窗口设备 id 掩码
7:0	axi2unitid3_base	8	0x0	R/W	3号窗口设备 id 基址



## 8 MISC 低速设备 (D2:F0)

### 8.1 MISC 低速设备配置寄存器 (MISC-D2:F0)

MISC 低速设备块包含多个低速设备。该总线控制器作为一个设备具有相应的配置头空间。配置头的默认值见表 8-1。

表 8-1 MISC 低速设备块的 PCI 配置头 (MISC-D2:F0)

地址偏移	简称	描述	默认值	访问类型
00h-01h	VID	Vendor ID	0014h	RO
02h-03h	DID	Device ID	7A22h	RO
04h-05h	PCICMD	PCI Command	0000h	R/W, RO
08h	RID	Revision ID	00h	RO
09h	PI	Programming Interface	00h	RO
0Ah	SCC	Sub Class Code	80h	RO
0Bh	BCC	Base Class Code	08h	RO
0Ch	CLS	Cache Line Size	10h	RO
0Eh	HEADTYP	Header Type	00h	RO
10h-17h	CNL_BAR	Control Block Base Address Register	0000000000000004h	R/W,RO
2Ch-2Dh	SVID	Subsystem Vendor ID	0000h	RO
2Eh-2Fh	SID	Subsystem Identification	0000h	RO
3Ch	INT_LN	Interrupt Line	FFh	R/W
3Dh	INT_PN	Interrupt Pin	00h	RO

注：表中未列出的地址空间表示保留。

下面列出与 PCI 配置头规范稍有不同的寄存器及其描述。

#### PCICMD-PCI 命令寄存器

地址偏移：04-05h

属性：R/W, RO

默认值：0000h

大小：16 位

位域	名称	访问	描述
15:2	Reserved	RO	保留
1	Memory Space Enable	R/W	该位用来控制是否使能对 MISC 低速设备块内部设备的访问。 0：禁止访问； 1：使能对 MISC 低速设备块内部设备的访问。在将该位配置为 1 之前，必须先配置 PCNL_BAR 寄存器。
0	Reserved	RO	保留

## 8.2 内部设备地址路由

MISC 低速设备块下挂载了多个低速设备，包括：UART/CAN、I2C、PWM、HPET、ACPI、RTC 和 GPIO。这些低速设备根据地址位的 bit[18:16] 进行区分，对应关系为：

表 8-2 MISC 低速设备地址路由

bit[18:16]	0	1	2	4	5	6	7
设备	UART/CAN	I2C	PWM	HPET	ACPI/RTC	GPIO	-
访问类型	B	B	W	W	W	BHW	保留

对于 UART、I2C、PWM、ACPI/RTC 来说，由于包含多个控制器，它们需要进一步的路由。这些设备块的内部路由见表 8-2。不同的设备块需要的路由地址位数是不同的。

表 8-3 MISC 低速设备地址路由（续）

	0	1	2	3	4	5	6	7
UART (bit[10:8])	UART0	UART1	UART2	UART3	CAN0	CAN1	CAN2	CAN3
I2C (bit[10:8])	I2C0	I2C1	I2C2	I2C3	I2C4	I2C5	-	-
PWM (bit[9:8])	PWM0	PWM1	PWM2	PWM3	-	-	-	-
ACPI/RTC (bit[8])	ACPI	RTC	-	-	-	-	-	-

后续章节将分别对这些低速设备进行介绍。

## 9 UART 控制器

桥片集成的 UART 控制器遵守 RS232 标准，控制器在设计上兼容 16550A。UART 控制器的内部时钟频率为 50MHz。UART 总线支持的最高波特率为 460800。

桥片集成了 4 个 UART 控制器，统一安排在 UART 模块内。其中，UART1、UART2、UART3 只能工作在两线 UART 模式，UART0 可以工作在全功能 UART 模式，也可以工作在两线 UART 模式。UART 引脚的工作模式见表 9-1。此外，UART 还可复用为 GPIO 功能，部分 UART 引脚可以复用为 I2C 功能。与 UART 相关的引脚复用配置寄存器见第 4.6 节。

表 9-1 UART 功能复用

UART_TXD/RXD	UART_RTS/CTS	UART_DTR/DSR	UART_RI/DCD
UART0			
UART0	UART1	UART2	UART3

### 9.1 访问地址

UART 控制器的访问基地址为 MISC 低速设备块的基地址加偏移 0x0。

**注意：**UART 模块仅支持按字节访问。

4 个 UART 控制器由 bit[9:8]区分，UART 模块内部物理地址划分见表 9-2。

表 9-2 UART 模块物理地址构成

地址位	构成	备注
[15:10]	0	保留
[09:08]	UART 号	0x0 - 0x3，分别表示各个 UART 控制器
[07:00]	REG	内部寄存器地址

### 9.2 寄存器描述

#### 数据寄存器 (DAT)

偏移量：0x00

复位值：0x00

位域	位域名称	位宽	访问	描述
7:0	Tx FIFO	8	W	数据传输寄存器

#### 中断使能寄存器 (IER)

偏移量：0x01

复位值：0x00

位域	位域名称	位宽	访问	描述
7:4	Reserved	4	R/W	保留
3	IME	1	R/W	Modem 状态中断使能 0: 关闭 1: 打开
2	ILE	1	R/W	接收器线路状态中断使能 0: 关闭 1: 打开
1	ITxE	1	R/W	传输保存寄存器为空中断使能 0: 关闭 1: 打开
0	IRxE	1	R/W	接收有效数据中断使能 0: 关闭 1: 打开

### 中断标识寄存器 (IIR)

偏移量: 0x02

复位值: 0xc1

位域	位域名称	位宽	访问	描述
7:4	Reserved	4	R	保留
3:1	II	3	R	中断源表示位, 详见下表
0	INTp	1	R	中断表示位

### 中断控制功能表:

Bit 3	Bit 2	Bit 1	优先级	中断类型	中断源	中断复位控制
0	1	1	1st	接收线路状态	奇偶、溢出或帧错误, 或打断中断	读 LSR
0	1	0	2nd	接收到有效数据	FIFO 的字符个数达到 trigger 的水平	FIFO 的字符个数低于 trigger 的值
1	1	0	2nd	接收超时	在 FIFO 至少有一个字符, 但在 4 个字符时间内没有任何操作, 包括读和写操作	读接收 FIFO
0	0	1	3rd	传输保存寄存器为空	传输保存寄存器为空	写数据到 THR 或者多 IIR
0	0	0	4th	Modem 状态	CTS, DSR, RI or DCD.	读 MSR

### FIFO 控制寄存器 (FCR)

偏移量: 0x02

复位值: 0xc0

位域	位域名称	位宽	访问	描述
7:6	TL	2	W	接收 FIFO 提出中断申请的 trigger 值 00b: 1 字节 01b: 4 字节 10b: 8 字节 11b: 14 字节
5:3	Reserved	3	W	保留
2	Txset	1	W	清除发送 FIFO 的内容, 复位其逻辑

1	Rxset	1	W	清除接收 FIFO 的内容，复位其逻辑
0	Reserved	1	W	保留

### 线路控制寄存器 (LCR)

偏移量: 0x03

复位值: 0x03

位域	位域名称	位宽	访问	描述
7	dlab	1	R/W	分频锁存器访问位 1: 访问操作分频锁存器 0: 访问操作正常寄存器
6	bcb	1	R/W	打断控制位 1: 此时串口的输出被置为 0(打断状态). 0: 正常操作
5	spb	1	R/W	指定奇偶校验位 0: 不用指定奇偶校验位 1: 如果 LCR[4]位是 1 则传输和检查奇偶校验位为 0。如果 LCR[4]位是 0 则传输和检查奇偶校验位为 1。
4	eps	1	R/W	奇偶校验位选择 0: 在每个字符中有奇数个 1 (包括数据和奇偶校验位) 1: 在每个字符中有偶数个 1
3	pe	1	R/W	奇偶校验位使能 0: 没有奇偶校验位 1: 在输出时生成奇偶校验位, 输入则判断奇偶校验位
2	sb	1	R/W	定义生成停止位的位数 0: 1 个停止位 1: 在 5 位字符长度时是 1.5 个停止位, 其他长度是 2 个停止位
1:0	bec	2	R/W	设定每个字符的位数 00b: 5 位 01b: 6 位 10b: 7 位 11b: 8 位

### MODEM 控制寄存器 (MCR)

偏移量: 0x04

复位值: 0x00

位域	位域名称	位宽	访问	描述
7:5	Reserved	3	W	保留
4	Loop	1	W	回环模式控制位 0: 正常操作 1: 回环模式。在在回环模式中, TXD 输出一直为 1, 输出移位寄存器直接连到输入移位寄存器中。其他连接如下。 DTR     DSR RTS     CTS Out1    RI Out2    DCD
3	OUT2	1	W	在回环模式中连到 DCD 输入
2	OUT1	1	W	在回环模式中连到 RI 输入
1	RTSC	1	W	RTS 信号控制位
0	DTRC	1	W	DTR 信号控制位

### 线路状态寄存器 (LSR)

偏移量: 0x05

复位值: 0x00

位域	位域名称	位宽	访问	描述
7	ERROR	1	R	错误表示位 1: 至少有奇偶校验位错误, 帧错误或打断中断的一个。 0: 没有错误
6	TE	1	R	传输为空表示位 1: 传输 FIFO 和传输移位寄存器都为空。给传输 FIFO 写数据时清零 0: 有数据
5	TFE	1	R	传输 FIFO 位空表示位 1: 当前传输 FIFO 为空, 给传输 FIFO 写数据时清零 0: 有数据
4	BI	1	R	打断中断表示位 1: 接收到起始位+数据+奇偶位+停止位都是 0, 即有打断中断 0: 没有打断
3	FE	1	R	帧错误表示位 1: 接收的数据没有停止位 0: 没有错误
2	PE	1	R	奇偶校验位错误表示位 1: 当前接收数据有奇偶错误 0: 没有奇偶错误
1	OE	1	R	数据溢出表示位 1: 有数据溢出 0: 无溢出
0	DR	1	R	接收数据有效表示位 0: 在 FIFO 中无数据 1: 在 FIFO 中有数据

对这个寄存器进行读操作时, LSR[4:1]和 LSR[7]被清零, LSR[6:5]在给传输 FIFO 写数据时清零, LSR[0]则对接收 FIFO 进行判断。

### MODEM 状态寄存器 (MSR)

偏移量: 0x06

复位值: 0x00

位域	位域名称	位宽	访问	描述
7	CDCD	1	R	DCD 输入值的反, 或者在回环模式中连到 Out2
6	CRI	1	R	RI 输入值的反, 或者在回环模式中连到 OUT1
5	CDSR	1	R	DSR 输入值的反, 或者在回环模式中连到 DTR
4	CCTS	1	R	CTS 输入值的反, 或者在回环模式中连到 RTS
3	DDCD	1	R	DDCD 指示位
2	TERI	1	R	RI 边沿检测。RI 状态从低到高变化
1	DDSR	1	R	DDSR 指示位
0	DCTS	1	R	DCTS 指示位

**分频锁存器**

偏移量: 0x00

复位值: 0x00

位域	位域名称	位宽	访问	描述
7:0	LSB	8	R/W	存放分频锁存器的低8位

偏移量: 0x01

复位值: 0x00

位域	位域名称	位宽	访问	描述
7:0	MSB	8	R/W	存放分频锁存器的高8位

## 10 CAN 控制器

龙芯 7A2000 集成了四路 CAN 接口控制器。CAN 总线是由发送数据线 TX 和接收数据线 RX 构成的串行总线，可发送和接收数据。器件与器件之间进行双向传送，最高传送速率 1Mbps。

### 10.1 访问地址及引脚复用

表 10-1 CAN 功能复用

UART_TXD/RXD	UART_RTS/CTS	UART_DTR/DSR	UART_RI/DCD
CAN0	CAN1	CAN2	CAN3

CAN 控制器的访问基地址为 MISC 低速设备块的基地址加偏移 0x0。4 个 CAN 控制器由 bit[10:8]区分，CAN 模块内部物理地址划分见表 10-2。

表 10-2 CAN 模块物理地址构成

地址位	构成	备注
[15:11]	0	保留
[10:08]	CAN 号	0x4 - 0x7，分别表示各个 CAN 号
[07:00]	REG	内部寄存器地址

与 CAN 相关的引脚设置寄存器见 4.6 节。

### 10.2 标准模式

地址区包括控制段和信息缓冲区，控制段在初始化载入是可被编程来配置通讯参数的，应发送的信息会被写入发送缓冲器，成功接收信息后，微控制器从接收缓冲器中读取接收的信息，然后释放空间以做下一步应用。

初始载入后，寄存器的验收代码，验收屏蔽，总线定时寄存器 0 和 1 以及输出控制就不能改变了。只有控制寄存器的复位位被置高时，才可以访问这些寄存器。在复位模式和工作模式两种不同的模式中，访问寄存器是不同的。当硬件复位或控制器掉线，状态寄存器的总线状态位时会自动进入复位模式。工作模式是通过置位控制寄存器的复位请求位激活的。

在标准模式下，CAN 控制器将 ID[10:3]的值和验收代码的值相同的消息包存入接收缓冲区。如果验收屏蔽码的某位为 1，则验收码对应的位不参与对 ID 的检查。



表 10-3 CAN 控制器标准模式下寄存器定义

CAN 地址	段	工作模式		复位模式	
		读	写	读	写
0	控制	控制	控制	控制	控制
1		FF	命令	FF	命令
2		状态	—	状态	—
3		FF	—	中断	—
4		FF	—	验收代码	验收代码
5		FF	—	验收屏蔽	验收屏蔽
6		FF	—	总线定时 0	总线定时 0
7		FF	—	总线定时 1	总线定时 1
8		保留	保留	保留	保留
9		保留	保留	保留	保留
10	发送缓冲器	ID(10-3)	ID(10-3)	FF	—
11		ID(2-0), RTR, DLC	ID(2-0), RTR, DLC	FF	—
12		数据字节 1	数据字节 1	FF	—
13		数据字节 2	数据字节 2	FF	—
14		数据字节 3	数据字节 3	FF	—
15		数据字节 4	数据字节 4	FF	—
16		数据字节 5	数据字节 5	FF	—
17		数据字节 6	数据字节 6	FF	—
18		数据字节 7	数据字节 7	FF	—
19		数据字节 8	数据字节 8	FF	—
20	接收缓冲器	ID(10-3)	ID(10-3)	FF	—
21		ID(2-0), RTR, DLC	ID(2-0), RTR, DLC	FF	—
22		数据字节 1	数据字节 1	FF	—
23		数据字节 2	数据字节 2	FF	—
24		数据字节 3	数据字节 3	FF	—
25		数据字节 4	数据字节 4	FF	—
26		数据字节 5	数据字节 5	FF	—
27		数据字节 6	数据字节 6	FF	—
28		数据字节 7	数据字节 7	FF	—
29		数据字节 8	数据字节 8	FF	—

### 10.2.1 控制寄存器 (CR)

中文名：控制寄存器

寄存器位宽： [7: 0]

偏移量： 0x00

复位值： 0x01

读此位的值总是逻辑 1。在硬启动或总线状态位设置为 1（总线关闭）时，复位请求位被置为 1。如果这些位被软件访问，其值将发生变化而且会影响内部时钟的下一个上升沿，在外部复位期间微控制器不能把复位请求位置为 0。如果把复位请求位设为 0，微控制器就必须检查这一位以保证外部复位引脚不保持为低。复位请求位的变化是同内部分频时钟同步的。读复位请求位能够反映出这种同步状态。

复位请求位被设为 0 后控制器将会等待

a) 一个总线空闲信号（11 个弱势位），如果前一次复位请求是硬件复位或 CPU 初始复位。

b) 128 个总线空闲，如果前一次复位请求是 CAN 控制器在重新进入总线开启模式前初始化总线造成的。

表 10-4 CAN 控制器标准模式下的控制寄存器格式

位域	位域名称	位宽	访问	描述
7: 5	Reserve	3	—	保留
4	OIE	1	RW	溢出中断使能
3	EIE	1	RW	错误中断使能
2	TIE	1	RW	发送中断使能
1	RIE	1	RW	接收中断使能
0	RR	1	RW	复位请求

## 10.2.2 命令寄存器（CMR）

中文名：命令寄存器

寄存器位宽： [7: 0]

偏移量： 0x01

复位值： 0x00

命令寄存器对微控制器来说是只写存储器如果去读这个地址返回值是 0xff

表 10-5 CAN 控制器标准模式下命令寄存器格式

位域	位域名称	位宽	访问	描述
7	EFF	1	W	扩展模式
6: 5	Reserve	2	—	保留
4	GTS	1	W	睡眠
3	CDO	1	W	清除数据溢出
2	RRB	1	W	释放接收缓冲器
1	AT	1	W	中止发送
0	TR	1	W	发送请求

### 10.2.3 状态寄存器（SR）

中文名：状态寄存器

寄存器位宽： [7: 0]

偏移量： 0x02

复位值： 0x00

表 10-6 CAN 控制器标准模式下状态寄存器格式

位域	位域名称	位宽	访问	描述
7	BS	1	R	总线状态
6	ES	1	R	出错状态
5	TS	1	R	发送状态
4	RS	1	R	接收状态
3	TCS	1	R	发送完毕状态
2	TBS	1	R	发送缓存器状态
1	DOS	1	R	数据溢出状态
0	RBS	1	R	接收缓存器状态

### 10.2.4 中断寄存器（IR）

中文名：中断寄存器

寄存器位宽： [7: 0]

偏移量： 0x03

复位值： 0x00

表 10-7 CAN 控制器标准模式下中断寄存器格式

位域	位域名称	位宽	访问	描述
7: 5	Reserved	1	R	保留
4	WUI	1	R	唤醒中断
3	DOI	1	R	数据溢出中断
2	EI	1	R	错误中断
1	TI	1	R	发送中断
0	RI	1	R	接收中断

### 10.2.5 验收代码寄存器（ACR）

中文名：验收代码寄存器

寄存器位宽： [7: 0]

偏移量： 0x04

复位值： 0x00

在复位情况下，该寄存器是可以读写的。

表 10-8 CAN 验收代码寄存器

位域	位域名称	位宽	访问	描述
7:0	AC	8	RW	ID 验收代码

## 10.2.6 验收屏蔽寄存器（AMR）

中文名：验收屏蔽寄存器

寄存器位宽： [7: 0]

偏移量： 0x05

复位值： 0x00

验收代码位 AC 和信息识别码的高 8 位 ID. 10-ID. 3 相等且与验收屏蔽位 AM 的相应位相或为 1 时数据可以接收。在复位情况下，该寄存器是可以读写的。

表 10-9 CAN 验收屏蔽寄存器

位域	位域名称	位宽	访问	描述
7:0	AM	8	RW	ID 屏蔽位

## 10.2.7 发送缓冲区列表

缓冲器是用来存储微控制器要 CAN 控制器发送的信息，它被分为描述符区和数据区。发送缓冲器的读/写只能由微控制器在工作模式下完成，在复位模式下读出的值总是 0xff。

表 10-10 CAN 控制器标准模式下发送缓冲区格式

地址	区	名称	数据位
10	发送缓冲器	识别码字节 1	ID(10-3)
11		识别码字节 2	ID(2-0), RTR, DLC
12		TX 数据 1	TX 数据 1
13		TX 数据 2	TX 数据 2
14		TX 数据 3	TX 数据 3
15		TX 数据 4	TX 数据 4
16		TX 数据 5	TX 数据 5
17		TX 数据 6	TX 数据 6
18		TX 数据 7	TX 数据 7
19		TX 数据 8	TX 数据 8

## 10.2.8 接收缓冲区列表

接收缓冲区的配置和发送缓冲区的一样，只是地址变为 20—29。

## 10.3 扩展模式

在扩展模式下，允许使用 11 位 ID 的标准帧和 29 位 ID 的扩展帧（是标准帧还是扩展帧由 TX 帧信息的最高位 IDE 位确定）。

在扩展模式下，CAN 控制器可以接收 11 位 ID 的标准帧也可以接收 29 位 ID 的扩展帧。在接收不同格式的帧的时候，验收代码 0~验收代码 3（code0 ~ code3）所检查的内容有所不同。验收屏蔽码 0~验收屏蔽码 3（mask0 ~ mask3）对应验收代码 0~验收代码 3。如果验收屏蔽码的某 1 位为 1，则对应的验收代码的那一位就不参与对接收包的 ID 的检查。

在扩展模式下，CAN 控制器可以对接收到的消息包进行单滤波也可以进行双滤波。在进行单滤波时，验收代码 0~验收代码 3 仅对单一 ID 进行过滤。在进行双滤波时，验收代码 0~验收代码 3 可以对两个不同的 ID 进行。CAN 控制器只将 ID 符合滤波条件的消息包存入接收缓冲区。

对 11 位 ID 包的单滤波：

允许将 rdata0 和 rdata1 用来扩展对 ID 的验收长度

$(rx\_id[10:3] == code0) \&\& (rtr == code1[4]) \&\& (rx\_id[2:0] == code1[7:5]) \&\& (rx\_data0 == code2) \&\& (rx\_data1 == code3)$

对 11 位 ID 包的双滤波：

$(rx\_id[10:3] == code0) \&\& (rtr == code1[4]) \&\& (rx\_id[2:0] == code1[7:5]) \&\& (rx\_data0 == \{code1[3:0], code3[3:0]\})$

或者

$(rx\_id[10:3] == code2) \&\& (rtr == code3[4]) \&\& (rx\_id[2:0] == code3[7:5])$

对 29 位 ID 包的单滤波：

$(rx\_id[28:21] == code0) \&\& (rx\_id[20:13] == code1) \&\& (rx\_id[12:5] == code2) \&\& (rtr == code3[2]) \&\& (rx\_id[4:0] == code3[7:3])$

对 29 位 ID 的双滤波：

$(rx\_id[28:21] == code0) \&\& (rx\_id[20:13] == code1)$

或者

$(rx\_id[28:21] == code2) \&\& (rx\_id[20:13] == code3)$

表 10-11 扩展模式下 CAN 控制器的地址列表

CAN 地址	工作模式		复位模式	
	读	写	读	写
0	控制	控制	控制	控制
1	0	命令	0	命令

CAN 地址	工作模式		复位模式	
2	状态	—	状态	—
3	中断	—	中断	—
4	中断使能	中断使能	中断使能	中断使能
5	—	—	验收屏蔽	验收屏蔽
6	总线定时 0	—	总线定时 0	总线定时 0
7	总线定时 1	—	总线定时 1	总线定时 1
8	保留	保留	保留	保留
9	保留	保留	保留	保留
10	保留	保留	保留	保留
11	仲裁丢失捕捉	—	仲裁丢失捕捉	—
12	错误代码捕捉	—	错误代码捕捉	—
13	错误警报限制	—	错误警报限制	—
14	RX 错误计数器	—	RX 错误计数器	—
15	TX 错误计数器	—	TX 错误计数器	—
16	RX 帧信息 {IDE, RTR, 2' h0, DLC[3:0]}	TX 帧信息 {IDE, RTR, 2' h0, DLC[3:0]}	验收代码 0 Id[28:21] (扩展帧) Id[10:3] (非扩展帧)	验收代码 0
17	RX_Id[28:21] (扩展帧) RX_Id[10:3] (非扩展帧)	TX_Id[28:21] (扩展帧) TX_Id[10:3] (非扩展帧)	验收代码 1 Id[20:13] (扩展帧) {Id[2:0], RTR, 4' h0} (非扩展帧, 单滤波) {Id[2:0], RTR, data0[7:4]} (非扩展帧, 双滤波) {Id[2:0], RTR, 4' h0} (非扩展帧, 单滤波)	验收代码 1
18	RX_Id[20:13] (扩展帧) {RX_Id[2:0], 5' h0} (非扩展帧)	TX_Id[20:13] (扩展帧) {TX_Id[2:0], 5' h0} (非扩展帧)	验收代码 2 Id2[28:21] (扩展帧, 双滤波) Id[12:5] (扩展帧, 单滤波) Id2[10:3] (非扩展帧, 双滤波) Data0 (非扩展帧, 单滤波)	验收代码 2
19	RX Id[12:5] (扩展帧) RX data0 (非扩展帧)	TX Id[12:5] (扩展帧) TX data0 (非扩展帧)	验收代码 3 Id2[20:13] (扩展帧, 双滤波) {id[4:0], RTR, 2' h0} (扩展帧, 单滤波) {id2[2:0], RTR, data0[3:0]} (非扩展帧, 双滤波) Data1 (非扩展帧, 单滤波)	验收代码 3
20	{RX_id[4:0], 3' h0} (扩展帧) RX data1 (非扩展帧)	{TX_id[4:0], 3' h0} (扩展帧) TX data1 (非扩展帧)	验收屏蔽 0 (不判断为 1 的那位的 id 值)	验收屏蔽 0

CAN 地址	工作模式		复位模式	
	21	RX data0(扩展帧) RX data2(非扩展帧)	TX data0(扩展帧) TX data2(非扩展帧)	验收屏蔽 1
22	RX data1(扩展帧) RX data3(非扩展帧)	TX data1(扩展帧) TX data3(非扩展帧)	验收屏蔽 2	验收屏蔽 2
23	RX data2(扩展帧) RX Data4(非扩展帧)	TX data2(扩展帧) TX Data4(非扩展帧)	验收屏蔽 3	验收屏蔽 3
24	RX data3(扩展帧) RX data5(非扩展帧)	TX data3(扩展帧) TX data5(非扩展帧)	—	—
25	RX data4(扩展帧) RX data6(非扩展帧)	TX data4(扩展帧) TX data6(非扩展帧)	—	—
26	RX data5(扩展帧) RX data7(非扩展帧)	TX data5(扩展帧) TX data7(非扩展帧)	—	—
27	RX data6(扩展帧)	TX data6(扩展帧)	—	—
28	RX data7(扩展帧)	TX data7(扩展帧)	—	—
29	RX 信息计数器	—	RX 信息计数器	—

### 10.3.1 模式寄存器 (MOD)

中文名：模式寄存器

寄存器位宽： [7: 0]

偏移量： 0x00

复位值： 0x01

读此位的值总是逻辑 1。在硬启动或总线状态位设置为 1（总线关闭）时，复位请求位被置为 1。如果这些位被软件访问，其值将发生变化而且会影响内部时钟的下一个上升沿，在外部复位期间微控制器不能把复位请求位置为 0。如果把复位请求位设为 0，微控制器就必须检查这一位以保证外部复位引脚不保持为低。复位请求位的变化是同内部分频时钟同步的。读复位请求位能够反映出这种同步状态。

复位请求位被设为 0 后控制器将会等待

a) 一个总线空闲信号（11 个弱势位），如果前一次复位请求是硬件复位或 CPU 初始复位。

b) 128 个总线空闲，如果前一次复位请求是 CAN 控制器在重新进入总线开启模式前初始化总线造成的。

表 10-12 CAN 控制器扩展模式下的模式寄存器格式

位域	位域名称	位宽	访问	描述
7: 5	Reserve	3	—	保留
4	SM	1	RW	睡眠模式
3	AFM	1	RW	单/双滤波模式(0 为双滤波, 仅复位模式可写)
2	STM	1	RW	正常工作模式(1 为自测试模式, 仅复位模式可写)
1	LOM	1	RW	只听模式(仅复位模式可写)
0	RM	1	RW	复位模式

### 10.3.2 命令寄存器 (CMR)

中文名: 命令寄存器

寄存器位宽: [7: 0]

偏移量: 0x01

复位值: 0x00

命令寄存器对微控制器来说是只写存储器如果去读这个地址返回值是 0xff

表 10-13 CAN 控制器扩展模式下命令寄存器格式

位域	位域名称	位宽	访问	描述
7	EFF	1	W	扩展模式(仅在复位模式下可写)
6: 5	Reserve	2	—	保留
4	SRR	1	W	自接收请求(和 TR 不能同时为 1)
3	CDO	1	W	清除数据溢出
2	RRB	1	W	释放接收缓冲器
1	AT	1	W	中止发送
0	TR	1	W	发送请求(和 SRR 不能同时为 1)

### 10.3.3 状态寄存器 (SR)

中文名: 状态寄存器

寄存器位宽: [7: 0]

偏移量: 0x02

复位值: 0x00

表 10-14 CAN 控制器扩展模式下状态寄存器格式

位域	位域名称	位宽	访问	描述
7	BS	1	R	总线状态
6	ES	1	R	出错状态
5	TS	1	R	发送状态



4	RS	1	R	接收状态
3	TCS	1	R	发送完毕状态
2	TBS	1	R	发送缓存器状态
1	DOS	1	R	数据溢出状态
0	RBS	1	R	接收缓存器状态

### 10.3.4 中断寄存器（IR）

中文名：中断寄存器

寄存器位宽： [7: 0]

偏移量： 0x03

复位值： 0x00

表 10-15 CAN 控制器扩展模式下中断寄存器格式

位域	位域名称	位宽	访问	描述
7	BEI	1	R	总线错误中断
6	ALI	1	R	仲裁丢失中断
5	EPI	1	R	错误消极中断
4	WUI	1	R	唤醒中断
3	DOI	1	R	数据溢出中断
2	EI	1	R	错误中断
1	TI	1	R	发送中断
0	RI	1	R	接收中断

### 10.3.5 中断使能寄存器（IER）

中文名：中断使能寄存器

寄存器位宽： [7: 0]

偏移量： 0x04

复位值： 0x00

表 10-16 CAN 控制器扩展模式下中断使能寄存器格式

位域	位域名称	位宽	访问	描述
7	BEIE	1	RW	总线错误中断使能
6	ALIE	1	RW	仲裁丢失中断使能
5	EPIE	1	RW	错误消极中断使能
4	WUIE	1	RW	唤醒中断使能
3	DOIE	1	RW	数据溢出中断使能
2	EIE	1	RW	错误中断使能
1	TIE	1	RW	发送中断使能

0	RIE	1	RW	接收中断使能
---	-----	---	----	--------

### 10.3.6 仲裁丢失捕捉寄存器

中文名：仲裁丢失捕捉寄存器

寄存器位宽： [7: 0]

偏移量： 0xB

复位值： 0x00

表 10-17 CAN 控制器扩展模式下仲裁丢失捕捉寄存器格式

位域	位域名称	位宽	访问	描述
7: 5	—	3	R	保留
4	BITN04	1	R	第四位
3	BITN03	1	R	第三位
2	BITN02	1	R	第二位
1	BITN01	1	R	第一位
0	BITN00	1	R	第零位

位					十进制值	功能
ALC. 4	ALC. 3	ALC. 2	ALC. 1	ALC. 0		
0	0	0	0	0	0	仲裁丢失在识别码的 bit1
0	0	0	0	1	1	仲裁丢失在识别码的 bit2
0	0	0	1	0	2	仲裁丢失在识别码的 bit3
0	0	0	1	1	3	仲裁丢失在识别码的 bit4
0	0	1	0	0	4	仲裁丢失在识别码的 bit5
0	0	1	0	1	5	仲裁丢失在识别码的 bit6
0	0	1	1	0	6	仲裁丢失在识别码的 bit7
0	0	1	1	1	7	仲裁丢失在识别码的 bit8
0	1	0	0	0	8	仲裁丢失在识别码的 bit9
0	1	0	0	1	9	仲裁丢失在识别码的 bit10
0	1	0	1	0	10	仲裁丢失在识别码的 bit11
0	1	0	1	1	11	仲裁丢失在 SRTR 位
0	1	1	0	0	12	仲裁丢失在 IDE 位
0	1	1	0	1	13	仲裁丢失在识别码的 bit12
0	1	1	1	0	14	仲裁丢失在识别码的 bit13
0	1	1	1	1	15	仲裁丢失在识别码的 bit14
1	0	0	0	0	16	仲裁丢失在识别码的 bit15
1	0	0	0	1	17	仲裁丢失在识别码的 bit16
1	0	0	1	0	18	仲裁丢失在识别码的 bit17
1	0	0	1	1	19	仲裁丢失在识别码的 bit18

1	0	1	0	0	20	仲裁丢失在识别码的 bit19
1	0	1	0	1	21	仲裁丢失在识别码的 bit20
1	0	1	1	0	22	仲裁丢失在识别码的 bit21
1	0	1	1	1	23	仲裁丢失在识别码的 bit22
1	1	0	0	0	24	仲裁丢失在识别码的 bit23
1	1	0	0	1	25	仲裁丢失在识别码的 bit24
1	1	0	1	0	26	仲裁丢失在识别码的 bit25
1	1	0	1	1	27	仲裁丢失在识别码的 bit26
1	1	1	0	0	28	仲裁丢失在识别码的 bit27
1	1	1	0	1	29	仲裁丢失在识别码的 bit28
1	1	1	1	0	30	仲裁丢失在识别码的 bit29
1	1	1	1	1	31	仲裁丢失在识别码的 RTR 位

### 10.3.7 错误警报限制寄存器 (EMLR)

中文名：错误警报限制寄存器

寄存器位宽： [7: 0]

偏移量： 0xD

复位值： 0x60

表 10-18 CAN 错误劲爆限制寄存器

位域	位域名称	位宽	访问	描述
7: 0	EML	8	RW	错误警报阈值

### 10.3.8 RX 错误计数寄存器 (RXERR)

中文名： RX 错误计数寄存器

寄存器位宽： [7: 0]

偏移量： 0xE

复位值： 0x60

表 10-19 CAN 的 RX 错误计数寄存器

位域	位域名称	位宽	访问	描述
7: 0	RXERR	8	R	接收错误计数

### 10.3.9 TX 错误计数寄存器 (TXERR)

中文名： TX 错误计数寄存器

寄存器位宽： [7: 0]

偏移量： 0xF

复位值： 0x60

表 10-20 CAN 的 TX 错误计数寄存器

位域	位域名称	位宽	访问	描述
7: 0	TXERR	8	R	发送错误计数

### 10.3.10 验收滤波器

在验收滤波器的帮助下，只有当接收信息中的识别位和验收滤波器预定义的值相等时，CAN 控制器才允许将已接收信息存入 RXFIFO。验收滤波器由验收代码寄存器和验收屏蔽寄存器定义。在模式寄存器中选择单滤波器模式或者双滤波器模式。具体的配置可以参考 SJA1000 的数据手册。

### 10.3.11 RX 信息计数寄存器 (RMCR)

中文名： RX 信息计数寄存器

寄存器位宽： [7: 0]

偏移量： 0x1D

复位值： 0x00

表 10-21 CAN 的 RX 错信息计数寄存器

位域	位域名称	位宽	访问	描述
7: 0	RMCR	8	R	接收的数据帧计数器

## 10.4 公共寄存器

$$1\text{bit time} = \text{internal\_clock\_time} * ((\text{BRP} + 1) * 2) * (1 + (\text{TESG2} + 1) + (\text{TESG1} + 1))$$

### 10.4.1 总线定时寄存器 0 (BTR0)

中文名： 总线定时寄存器

寄存器位宽： [7: 0]

偏移量： 0x06

复位值： 0x00

**注：**在复位模式是可以读写的，工作模式是只读的。

表 10-22 CAN 总线定时寄存器 0

位域	位域名称	位宽	访问	描述
7: 6	SJW	8	RW	同步跳转宽度
5: 0	BRP	8	RW	波特率分频系数

## 10.4.2 总线定时寄存器 1 (BTR1)

中文名：总线定时寄存器 1

寄存器位宽： [7: 0]

偏移量： 0x07

复位值： 0x00

表 10-23 CAN 总线定时寄存器 1

位域	位域名称	位宽	访问	描述
7	SAM	1	RW	为 1 时三次采样，否则是一次采用
6: 4	TESG2	3	RW	一个 bit 中的时间段 2 的计数值
3: 0	TSEG1	4	RW	一个 bit 中的时间段 1 的计数值

## 10.4.3 输出控制寄存器 (OCR)

中文名：输出控制寄存器

寄存器位宽： [7: 0]

偏移量： 0x08

复位值： 0x00

表 10-24 CAN 输出控制寄存器

位域	位域名称	位宽	访问	描述
7: 0	OCR	8	RW	保留 (未使用)

## 11 I2C 控制器

桥片集成了 6 个 I2C 控制器，I2C 接口的最高传送速率 400kbps。

### 11.1 访问地址及引脚复用

I2C 控制器的访问基地址为 MISC 低速设备块的基地址加偏移 0x10000。

I2C 模块内部寄存器的物理地址构成如下：

表 11-1 I2C 控制器访问地址及引脚复用

地址位	构成	备注
[15:11]	0	保留
[10:8]	I2C 控制器编号	0x0-0x5 分别代表 I2C0-I2C5
[7:3]	0	保留
[2:0]	REG	内部寄存器地址

对于 I2C 模块，使用时要注意将对应的引脚设置为相应的功能。

与 I2C 相关的引脚设置见 4.6 节。

### 11.2 I2C 控制器寄存器说明

#### 分频锁存器低字节寄存器（PRERlo）

偏移量：0x00

复位值：0xff

表 11-2 I2C 分频锁存器低字节寄存器

位域	位域名称	位宽	访问	描述
7:0	PRERlo	8	R/W	存放分频锁存器的低 8 位

#### 分频锁存器高字节寄存器（PRERhi）

偏移量：0x01

复位值：0xff

表 11-3 I2C 分频锁存器高字节寄存器

位域	位域名称	位宽	访问	描述
7:0	PRERhi	8	R/W	存放分频锁存器的高 8 位

假设分频锁存器的值为 prescale，从 LPB 总线 PCLK 时钟输入的频率为 clock\_a，SCL 总线的输出频率为 clock\_s，则应满足如下关系：

$$\text{Prcescale} = \text{clock\_a}/(5*\text{clock\_s})-1$$

### 控制寄存器（CTR）

偏移量：0x02

复位值：0x00

表 11-4 I2C 控制寄存器

位域	位域名称	位宽	访问	描述
7	EN	1	R/W	模块工作使能位 为 1 正常工作模式, 0 对分频寄存器进行操作
6	IEN	1	R/W	中断使能位为 1 则打开中断
5	Slave_mode	1	R/W	从模式设置(仅对 I2C0 有效): 1 为从模式, 0 为主模式
4:0	Reserved	5	R/W	保留

### 发送数据寄存器（TXR）

偏移量：0x03

复位值：0x00

表 11-5 I2C 发送数据寄存器

位域	位域名称	位宽	访问	描述
7:1	DATA	7	W	存放下个将要发送的字节
0	DR/W	1	W	当数据传送时, 该位保存的是数据的最低位; 当地址传送时, 该位指示读写状态

### 接收数据寄存器（RXR）

偏移量：0x03

复位值：0x00

表 11-6 I2C 接收数据寄存器

位域	位域名称	位宽	访问	描述
7:0	RXR	8	R	存放最后一个接收到的字节

### 命令控制寄存器（CR）

偏移量：0x04

复位值：0x00

表 11-7 I2C 命令控制寄存器

位域	位域名称	位宽	访问	描述
7	STA	1	W	产生 START 信号

6	STO	1	W	产生 STOP 信号
5	RD	1	W	产生读信号
4	WR	1	W	产生写信号
3	ACK	1	W	产生应答信号
2:1	Reserved	2	W	保留
0	IACK	1	W	产生中断应答信号

这些位在 I2C 发送数据后硬件自动清零。对这些位读操作时候总是读回 0'。bit 3 为 1 时表示此次传输结束时控制器不发送 ack，反之结束时发送 ack。

### 状态寄存器 (SR)

偏移量: 0x04

复位值: 0x00

表 11-8 I2C 状态寄存器

位域	位域名称	位宽	访问	描述
7	RxACK	1	R	收到应答位 1 没收到应答位 0 收到应答位
6	Busy	1	R	I2c 总线忙标志位 1 总线在忙 0 总线空闲
5	AL	1	R	当 I2C 核失去 I2C 总线控制权时，该位置 1
4:2	Reserved	3	R	保留
1	TIP	1	R	指示传输的过程 1 表示正在传输数据 0 表示数据传输完毕
0	IF	1	R	中断标志位，一个数据传输完，或另外一个器件发起数据传输，该位置 1

### 从模式控制寄存器 (SLV\_CTRL)

偏移量: 0x07

复位值: 0x00

仅 I2C0 有该组寄存器。

表 11-9 I2C 从模式控制寄存器

位域	位域名称	位宽	访问	描述
7	slv_en	1	RW	I2C0 从模式使能，高有效
6:0	slv_addr	7	RW	I2C0 从模式地址



## 12 PWM 控制器

桥片里实现了四路脉冲宽度调节/计数控制器，以下简称 PWM。四路 PWM 的工作和控制方式完全相同。每路 PWM 有一路脉冲宽度输出信号和一路待测脉冲输入信号。时钟频率为 50MHz，计数寄存器和参考寄存器均为 32 位数据宽度。

### 12.1 访问地址及引脚复用

PWM 控制器的访问基地址为 MISC 低速设备块的基地址加偏移 0x20000。PWM 控制器内部寄存器的物理地址构成如下：

表 12-1 PWM 控制器访问地址及引脚复用

地址位	构成	备注
[15:10]	0	保留
[9:8]	PWM 编号	取值 0x0-0x3 分别代表 PWM0-PWM3
[7:4]	0	保留
[3:0]	REG	内部寄存器地址

对于 PWM 模块，使用时要注意将对应的引脚设置为相应的功能。与 PWM 相关的引脚设置见 4.6 节。

### 12.2 寄存器描述

每路控制器共有五个寄存器，具体描述如下：

表 12-2 PWM 寄存器列表

名称	地址	宽度	访问	说明
Low_buffer	Base + 0x4	32	R/W	低脉冲缓冲寄存器
Full_buffer	Base + 0x8	32	R/W	脉冲周期缓冲寄存器
CTRL	Base + 0xC	11	R/W	控制寄存器

### PWM 控制寄存器设置

表 12-3 PWM 控制寄存器设置

位域	名称	访问	复位值	说明
0	EN	R/W	0	计数器使能位 置 1 时：CNTR 用来计数 置 0 时：CNTR 停止计数（输出保持）
2: 1	Reserve d	R/W	0	预留
3	OE	R/W	0	脉冲输出使能控制位，低有效 置 0 时：脉冲输出使能 置 1 时：脉冲输出屏蔽

4	SINGLE	R/W	0	单脉冲控制位 置 1 时：脉冲仅产生一次 置 0 时：脉冲持续产生
5	INTE	R/W	0	中断使能位 置 1 时：当 full_pulse 到 1 时送中断 置 0 时：不产生中断
6	INT	R/W	0	中断位 读操作：1 表示有中断产生, 0 表示没有中断 写入 1：清中断
7	RST	R/W	0	使得 Low_level 和 full_pulse 计数器重置 置 1 时：计数器重置（从 buffer 读，输出低电平） 置 0 时：计数器正常工作
8	CAPTE	R/W	0	测量脉冲使能 置 1 时：测量脉冲模式 置 0 时：非测量脉冲模式（一般而言则是脉冲输出模式）
9	INVERT	R/W	0	输出翻转使能 置 1 时：使脉冲在输出出去发生信号翻转（周期以高电平开始） 置 0 时：使脉冲保持原始输出（周期以低电平开始）
10	DZONE	R/W	0	防死区功能使能 置 1 时：该计数模块需要启用防死区功能 置 0 时：该模块无需防死区功能

## 12.3 功能说明

### 12.3.1 脉宽调制功能

Low\_buffer 和 Full\_buffer 寄存器可以由系统编程写入获得初始值。系统编程写入完毕后，模块内部的 low\_level 和 full\_pulse 寄存器分别从 Low\_buffer 和 Full\_buffer 缓冲寄存器中读取初值，之后在系统时钟驱动下不断自减（初始输出低电平）。当 low\_level 寄存器到达 1 之后，输出变为高电平，此时 full\_pulse 仍在自减。当 full\_pulse 寄存器到达 1 之后，输出变为低电平，low\_level 和 full\_pulse 又分别从 Low\_buffer 和 Full\_buffer 缓冲寄存器中读取初值，然后重新开始不断自减，控制器就产生连续不断的脉冲宽度输出。当 full\_pulse 寄存器的值等于 1 的时候，可以配置产生一个中断，从而作为定时器使用。

例：如果要产生宽度为系统时钟周期 50 倍的高脉宽和 90 倍的低脉宽，在 low\_buffer 中应该配置初始值 90，在 full\_buffer 寄存器中配置初始值  $(50+90)=140$ 。

值得说明的是，由于两个缓冲寄存器的写入有先后之分，在某些特殊的情况下（比如写入时刻刚好是旧脉冲结束时）会使得输出脉冲有异于预期。推荐的做法是在向缓冲寄存器写入新数前，将控制寄存器 EN 位写 0，在写入新数之后再写 1。值得说明的是，即使没有重写 EN 位，紊乱的脉冲输出最多只会维持一个周期。

如果对两个缓冲寄存器都写 0，则输出为低电平；如果对 low\_buffer 写 0，对 full\_buffer 写 1，则输出高电平；如果写入 Low\_buffer 的值不小于 full\_buffer，则输出低电平。但这三类数值都是不推荐的。

此外，缓冲寄存器的数值写入应当先于 CTRL 控制寄存器。

### 12.3.2 脉冲测量功能

待测脉冲信号连在 PWM 输入信号接口上，在设置完 CTRL 控制寄存器后，在系统时钟的驱动下，Low\_level 和 full\_pulse 寄存器开始不断自增。当检测到输入脉冲信号上跳变时，将 Low\_level 寄存器的值传送到 low\_buffer 寄存器中；当检测到输入脉冲信号下跳变时，将 full\_pulse 寄存器的值传送到 full\_buffer 寄存器中，并将 Low\_level 和 full\_pulse 寄存器置 1，重新开始计数。

例：如果要输入脉冲为系统时钟 50 倍的高脉宽和 90 倍的低脉宽，在 low\_buffer 中最终读出的值为 90，在 full\_buffer 寄存器中读出的值为 (50+90)=140。

待测脉冲应当是周期信号，且脉冲周期不应超出 32 位计数器能计量的范围。

每次测量均是从下跳变开始，到下一个下跳变结束。由于测量及缓冲的需要，在连续测量两个脉冲周期后，low\_buffer 和 full\_buffer 寄存器中存储的才是正确的脉冲参数。

若出现持续的周期超过 0xFFFF\_FFF9 的脉冲，控制寄存器 INT 位会被置 1，表示待测脉冲超出了计量范围。

### 12.3.3 防死区功能

四路 PWM 都配备了防死区功能，可以防止四路脉冲输出同时发生跳变。

将四路模块分别标记为 PWM\_0、PWM\_1、PWM\_2、PWM\_3，它们的优先级为 0>1>2>3，即若要同时产生跳变，在 PWM\_0 跳变之后 PWM\_1 才能跳变（低优先级的信号被“抹去”一个或多个系统时钟），依此类推。该优先级是固化的，不可配置。

一个典型的防死区示例如下（PWM\_\*为未开防死区的输出，PWM\_\*'为打开防死区后的输出）：

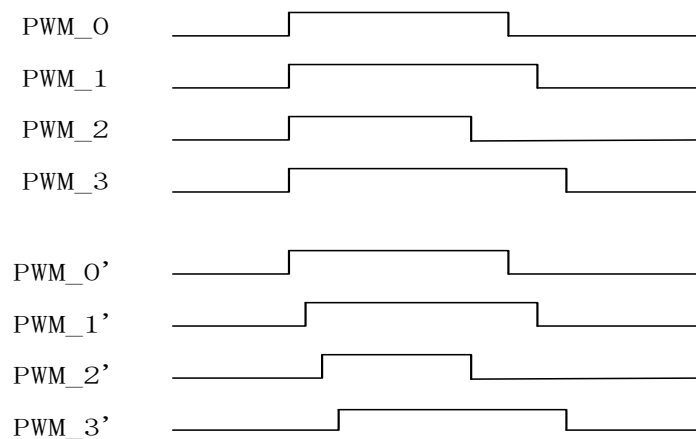


图 12-1 防死区功能

## 13 HPET 控制器

HPET 控制器兼容标准规范。内部包括一个 64 位的主计数器（main count）以及三个 32 位的比较器（comparator）。在这三个比较器中，比较器 0 支持周期性中断（periodic-capable）和非周期性中断，其他两个比较器支持非周期性中断。

### 13.1 访问地址

HPET 模块的访问基地址为 MISC 低速设备块的基地址加偏移 0x40000。HPET 控制器内部寄存器的物理地址构成如下：

表 13-1 HPET 控制器访问地址

地址位	构成	备注
[15:08]	0	保留
[07:00]	REG	内部寄存器地址

### 13.2 寄存器描述

下表列出了 HPET 的寄存器：

表 13-2 HPET 寄存器列表

寄存器偏移地址	寄存器	类型
000-007h	General Capabilities and ID Register	只读
008-00Fh	Reserved	
010-017h	General Configuration Register	读/写
018-01Fh	Reserved	R/WC
020-027h	General Interrupt Status Register	R/W
028-0EFh	Reserved	
0F0-0F7h	Main Counter Value Register	R/W
100-107h	Timer 0 Configuration and Capability Register	R/W
108-10Fh	Timer 0 Comparator Value Register	R/W
110-11Fh	Reserved	
120-127h	Timer 1 Configuration and Capability Register	R/W
128-12Fh	Timer 1 Comparator Value Register	R/W
130-13Fh	Reserved	
140-147h	Timer 2 Configuration and Capability Register	R/W
148-14Fh	Timer 2 Comparator Value Register	R/W
150-15Fh	Reserved	

若系统在状态转换过程中需要保存这些寄存器的值以便随后恢复，那么操作系统负责保存这些寄存器的值，硬件无需保存这些寄存器的值。因此当系统处于 S3, S4, S5 状态时，这些寄存器无需维持。

### General Capabilities and ID Register

表 13-3 General Capabilities and ID Register

位域	名称	描述	读写特性
63: 32	COUNTER_CLK_PERIOD	Main Counter Tick Period: 这个域标示了主计时器的计时频率，以 fps ( $10^{-15}$ s) 为单位。这个值必须大于 0，且小于或等于 05F5E100 (100ns, 即 10MHz)	RO
31: 16	VENDOR_ID		RO
15: 14	Reserved		
13	COUNT_SIZE_CAP	Counter Size: 主计时器的宽度; 0: 32 bits 1: 64 bits	RO
12:8	NUM_TIM_CAP	Num of Timer: 定时器的个数; 这个域的值指示最后一个定时器的编号, GS 南桥桥片的 HPET 有三个定时器, 因此这个域的值是 2。	RO
7:0	REV_ID	版本号; 不可为 0	RO

### General Configuration Register

表 13-4 General Configuration Register

位域	名称	描述	读写特性
63: 1	Reserved		
0	ENABLE_CNF	Overall Enable; 用来使能所有定时器产生中断。如果为 0, 主计时器停止计时且所有的定时器都不再产生中断。 0: 主计时器停止计时且所有的定时器都不再产生中断; 1: 主计时器计时且允许定时器产生中断;	R/W

### General Interrupt Status Register

表 13-5 General Interrupt Status Register

位域	名称	描述	读写特性
63: 3	Reserved		
2	T2_INT_STS	Timer 2 Interrupt Active: 功能同 T0_INT_STS	R/WC
1	T1_INT_STS	Timer 1 Interrupt Active: 功能同 T0_INT_STS	R/WC
0	T0_INT_STS	Timer 0 Interrupt Active: 功能依赖于这个定时器的中断触发模式是电平触发还是边沿触发: 如果是电平触发模式: 这位默认是 0。当对应的定时器发生中断, 那么有硬件将其置 1。一旦被置位, 软件往这位写 1 将会清空这位。往这位写 0, 则无意义。 如果边沿触发模式: 软件将忽略这位。软件通常往这位写 0。	R/WC

各个定时器的中断触发模式由各自 Configuration and Capability 寄存器的 Tn\_TYPE\_CNF 位确定。

## Main Counter Value Register

表 13-6 Main Counter Value Register

位域	名称	描述	读写特性
63: 32	Reserved		
31: 0	Main_Counter_Val	主计时器的值；只有当主计时器停止计时时，才允许修改这个寄存器的值。	R/W

## Timer N Configuration and Capabilities Register

表 13-7 Timer N Configuration and Capabilities Register

位域	名称	描述	读写特性
63: 9	Reserved		
8	Tn_32MODE_CNF	Timer n 32-bit 模式 (N 为 0-2)。当定时器为 32 位时，这位为 0，且只读	RO
7	Reserved		RO
6	Tn_VAL_SET_CNF	Timer N Value Set (N 为 0-2)：只有能产生周期性中断的定时器才会使用这个域。通过对这位写 1，软件能直接修改周期性定时器的累加器。软件无需对这位清 0 GS 南桥桥片中只有 Timer 0 能产生周期性中断，因此对 Timer0 来讲，这位是可读可写。而对于 Timer1，Timer2，这位默认为 0，且为只读。	R/W
5	Tn_SIZE_CAP	Timer N Size；Timer N 的宽度 (N 为 0-2)。0：32 位宽。	RO
4	Tn_PER_INT_CAP	Timer N Periodic Interrupt Capable (N 为 0-2)： 1：定时器能产生周期性中断； 0：定时器不能产生周期性中断；	RO
3	Tn_TYPE_CNF	Timer N type (N 为 0-2)： 如果对应的 Tn_PER_INT_CAP 位为 0，那么这位为只读，且默认为 0。 若对应的 Tn_PER_INT_CAP 位为 1，那么这位可读可写。用作使能相应的定时器产生周期性中断。 1：使能定时器产生周期性中断 0：使能定时器产生非周期性中断	R/W
2	Tn_INT_ENB_CNF	Timer N interrupt Enable (N 为 0-2)：使能定时器产生中断	R/W
1	Tn_INT_TYPE_CNF	Timer N Interrupt Type (N 为 0-2)： 0：定时器的中断触发模式为边沿触发；这意味着对应的定时器将产生边沿触发中断。若另外的的中断产生，那么将产生另外的边沿。 1：定时器的中断触发模式为电平触发；这意味着对应的定时器将产生电平触发中断。这个中断将一直有效直到被软件清掉 (General Interrupt Status Register)。	
0	Reserved		

## Timer N Comparator Value Register

表 13-8 Timer N Comparator Value Register

位域	名称	描述	读写特性
63: 32	Reserved	保留	
31: 0	Tn_Com_VAL	<p>Tn_Comparator value (N 为 0-2)：定时器比较器的值；</p> <p>当对应的定时器配置为非周期性模式时： 这个寄存器的值将与主计时器寄存器的值做比较； 若主计时器的值与比较器的值相等时，则产生定时中断（如果； 对应的中断使能打开）。</p> <p>比较器的值不会因为中断的产生而发生变化</p> <p>若对应的定时器配置为周期性模式时： 当主计时器的值域比较器的值相等时，产生中断（如果对应的中 断使能被打开）；</p> <p>如果产生中断，那么比较器的值将累加最后一次软件写入比较器 的值。比如当比较器的值被写入 0x0123h， 那么当主计时器的值为 0x123h 时，产生中断； 比较器的值被硬件修改为 0x246h； 当主计时器的值达到 0x246h 时，产生另外一个中断； 比较器的值被硬件修改为 0x369h。</p> <p>只要产生中断，那么比较器的值都会累加；直到比较器的值达到 最大 (0xffffffff)，那么累加器的值将会继续累加。比如当比 较器的值是 FFFF0000h，而最后一次由软件写入比较器的值是 20000。当中断发生后，比较器的值变为 00010000h。</p>	R/W

## 14 电源管理模块（ACPI）

桥片电源管理模块提供系统功耗管理功能。支持 Advanced Configuration and Power Interface, Version 4.0a(ACPI)，提供相应的功耗管理功能。

- 系统休眠与唤醒，支持ACPI S3（待机到内存），ACPI S4（待机到硬盘），ACPI S5（软关机），并且支持电源失效检测和自动系统恢复。支持多种唤醒方式（USB，GMAC，电源开关等）。
- 系统时钟控制，模块时钟门控，多种方式调节频率。
- 集成一个看门狗。最大定时时间约82s。

### 14.1 访问地址

电源管理模块的访问基地址为 MISC 低速设备块的基地址加偏移 0x50000。

**注意：**ACPI 模块支持按 4/1 字节访问。

ACPI 模块的内部寄存器物理地址构成如下：

表 14-1 ACPI 访问地址

地址位	构成	备注
[15:8]	0	保留
[7:0]	REG	内部寄存器地址

### 14.2 电源级别

表 14-2 ACPI 状态说明

状态	描述
G0/S0	全部工作，该模式下系统全部工作。
G1/S1	暂不支持
G1/S3	Suspend to RAM(STR)，上下文保存到内存
G1/S4	Suspend to Disk(STD)，保存到硬盘，除唤醒电路全部掉电
G2/S5	Soft off，只有唤醒电路上电
G3	Mechanical off，所有供电失效

### 14.3 寄存器描述

本节介绍电源管理相关寄存器。寄存器电压域表示寄存器的该位所属电压域。



**PMCON\_SOC : SOC General PM Configuration Register**

地址偏移	电压域	属性
0x00	SOC	R/W, RO
位域	描述	
25	<b>PWRBTN_LVL - RO</b> 该位指示当前 PWRBTNn 信号状态。	
24:0	保留	

**PMCON\_RESUME : RESUME General PM Configuration Register**

地址偏移	电压域	属性
0x04	ACPI	R/W, RO, R/WC
位域	描述	
31:14	保留	
13	<b>VSB_GATE<sub>n</sub>_EN - R/W</b> 是否使能 VSB_GATE <sub>n</sub> 功能。0: 关闭; 1: 使能。 如果 RSMRST <sub>n</sub> 有效过, 该位为 1。重新上电后由系统配置该位。如果主板使用 VSB_GATE <sub>n</sub> 引脚作为电源管理控制信号, 系统软件必须将该位写 1。	
12:11	<b>VSB_GATE<sub>n</sub>_SLEEP_DLY - R/W</b> 用来控制 S0 到 S3 时, VSB_GATE <sub>n</sub> 信号下降沿相对 S3 <sub>n</sub> 下降沿的提前时间 2' b00: 休眠时提前 31.25ms; 2' b01: 休眠时提前 62.5ms; 2' b10: 休眠时提前 125ms; 2' b11: 休眠时提前 250ms; 如果 RSMRST <sub>n</sub> 有效过, 该字段为 2' b0。重新上电后由系统配置该字段。	
10:9	<b>VSB_GATE<sub>n</sub>_WAKE_DLY - R/W</b> 用来控制 S3 到 S0 时, VSB_GATE <sub>n</sub> 信号上升沿相对 S3 <sub>n</sub> 上升沿的延后时间 2' b00: 唤醒时延后 125ms; 2' b01: 唤醒时延后 250ms; 2' b10: 唤醒时延后 500ms; 2' b11: 唤醒时延后 1s; 如果 RSMRST <sub>n</sub> 有效过, 该字段为 2' b0。重新上电后由系统配置该字段。	
8	保留	
7	<b>USB_GMAC_OK - R/W</b> 如果 RSMRST <sub>n</sub> 有效过, 该位为 0, 表示 USB 和 GMAC 没有配置, 不能唤醒系统。重新上电后由系统配置该位。	
6	<b>CTT_STS - R/WC</b> 系统在 S0 状态时发生 temperature trip, 系统进入 G2/S5 状态, 该位为重新上电后系统检测记录事件状态	
5	<b>CTT_EN - R/W</b> 使能 temperature trip 保护机制	
4:3	保留	
2	<b>SRS (System Reset Status) - R/WC</b> 0: SYS_RESE <sub>Tn</sub> 没有被按下 1: SYS_RESE <sub>Tn</sub> 被按下过, 系统重新复位后需检查此位并作出相应清除操作。	
1	<b>PWROK_FLR (PWROK Failure) - R/WC</b> 当系统在 S0 状态时, PWROK 信号变无效该位置 1, 软件通过写 1 将该位清除。	
0	<b>DRAM_INIT - R/W</b> 该位不影响硬件功能, PMON 在进行 DRAM 初始化前将该位置 1, 结束 DRAM 初始化后将该位写 0, 软件可通过此位检查 DRAM 初始化是否被打断过。	

**PMCON\_RTC : RTC General PM Configuration Register**

地址偏移	电压域	属性
0x08	RTC	R/W, R/WC
位域	描述	
31:13	保留	
12	RTC_DLY_BYPASS - R/W 是否将 RTC 域输入的关键信号进行延迟控制，默认值为 1	
11	RESUME_2POWER - R/W RESUME 域与 ACPI 域是否为单独电压域，默认值为 1	
10	RTC_REG_LP - R/W 控制 RTC 模块中寄存器是否进入低功耗模式 0: 非低功耗 1: 低功耗	
9	CMOS_CLKDISABLE - R/W 控制 CMOS 模块的时钟是否关闭 0: 打开 1: 关闭	
8:7	保留	
6:5	S3_ASSERTION_WIDTH - R/W 这 2 bit 值代表 S3n 信号从有效到重新无效的最小时间间隔。 11: 1s 10: 125ms 01: 1ms 00: 60us	
4:3	S4_ASSERTION_WIDTH - R/W 这 2 bit 值代表 S4n 信号从有效到重新无效的最小时间间隔。 11: 4 s 10: 2 s 01: 1 s 00: 125 us	
2	S4_ASSERTION_EN - R/W 0: S4n 信号从有效到重新无效的间隔为几个 RTC 周期。 1: S4n 信号从有效到重新无效的间隔为 S4_ASSERTION_WIDTH 决定。	
1	PWR_FLR (Power Failure) - R/WC. 该位在 RTC 域，只能被 RTC_RSTn 复位。 如果置 1 表示系统发生过电源失效（进入 G3 状态），即除 RTC 以外所有供电失效过（RSMRSTn 有效过），软件通过写 1 将该位清除。	
0	AFTERG3_EN - R/W 该位决定系统进入 G3 状态后重新供电后的动作。 0: 系统在供电恢复后将自动回复到 S0 状态。 1: 系统将恢复到 S5 状态，如果发生电源失效时系统在 S4 状态，重新供电后系统恢复到 S4 状态。 该位会被 power button override 和 thermal trip 事件置 1。	

**PM1\_STS : Power Management 1 Status Register**

地址偏移	电压域	属性
0x0C	ACPI/RTC/SOC	R/WC
位域	描述	电压域
31:16	Reserved	
15	WAK_STS (Wake Status) - R/WC 0: 软件写 1 将该位清除。 1: 如果系统从任何一个休眠状态被唤醒事件唤醒，硬件将该位置 1。	ACPI

14	<b>PCIEXP_WAKE_STS - R/WC.</b> 1: PCIE 唤醒事件发生 0: 软件写 1 将该位清除	ACPI
13:12	Reserved	
11	<b>PRBTNOR_STS (Power Button Override Status) - R/WC</b> 0: 软件写 1 将该位清除 1: 当 power button override 发生时, 该位置 1, 系统无条件进入 G2/S5 状态, 同时将 AFTERG3_EN 位置 1。	RTC
10	<b>RTC_STS (RTC Status) - R/WC</b> 0: 软件写 1 将该位清除 1: 当 RTC 产生 alarm 时该位置 1。此外当 RTC_EN 有效时, 该位产生唤醒事件。	ACPI
9	Reserved	
8	<b>PWRBTN_STS (Power Button Status) - R/WC</b> 0: 软件写 1 将该位清除。Thermal trip 会清除该位。 1: 当按下 PWRBTNn 保持 16ms 以上 (4s 以下) 时, 该位会置 1。 在 S0 状态时, 当 PWRBTN_EN 和 PWRBTN_STS 同时有效时, 将产生中断。 在 S3-S5 任何休眠状态时, 如果 PWRBTN_STS 置位, 系统将恢复。	ACPI
7:1	保留	
0	<b>TMROF_STS (PM Timer Overflow Status) - R/WC</b> 0: 软件写 1 将该位清除 1: 当 24bit 计数器 (一个时钟周期 20ns) 的最高位翻转时, 该位置 1。记时功能推荐使用 HPET 完成。	SOC

### PM1\_EN : Power Management 1 Enable Register

地址偏移	电压域	属性
0x10	ACPI/RTC/SOC	R/W
位域	描述	电压域
31:15	保留	
14	<b>PCIEXP_WAKE_DIS - R/W</b> 当置位时, 不产生 PCIE 唤醒事件, 但是该位的值不影响 PCIEXP_WAKE_STS 的值。	ACPI
13:11	保留	
10	<b>RTC_EN (RTC Event Enable) - R/W</b> RTC 唤醒和中断使能	rtc
9	保留	
8	<b>PWRBTN_EN (Power Button Enable) - R/W.</b> PWRBTN 中断事件产生使能, 该位不影响 PWRBTN 唤醒事件产生。	ACPI
7:1	保留	
0	<b>TMROF_EN (PM Timer Overflow Enable) - R/W.</b> 如果该位置位, TMROF_STS 将产生中断。	SOC

### PM1\_CNT : Power Management 1 Control Register

地址偏移	电压域	属性
0x14	ACPI/RTC/SOC	R/W
位域	描述	电压域
31:14	保留	
13	<b>SLP_EN (Sleep Enable) - R/W.</b> 该位写 1 将会使系统进入 SLP_TYP 声明的休眠状态, 进入相关休眠状态后该	ACPI

	位自动恢复为 0。	
12:10	<b>SLP_TYP (Sleep Type) - R/W.</b> 该 3bit 表示系统的休眠状态。 000: 表示 S0 状态。 001: Reserved. 010: Reserved. 011: Reserved. 100: Reserved. 101: Suspend-to-RAM. S3n 信号有效, 进入 S3 状态。 110: Suspend-to-Disk. S3n, S4n 信号有效, 进入 S4 状态。 111: Soft Off. S3n, S4n, S5n 信号有效, 进入 S5 状态。	rtc
9:1	Reserved	
0	<b>INT_EN - R/W</b> 中断使能开关, 使能电源管理控制器中断信号的产生。	SOC

### PM1\_TMR : Power Management 1 Timer

地址偏移	电压域	属性
0x18	SOC	RO
位域	描述	
31:24	保留	
23:0	<b>TMR_VAL (Timer Value) - RO.</b> 计数器计数, 周期 8ns。当 23 位翻转时, 置位 TNROF_STS 位。 推荐使用 HPET。	

### GPE0\_STS : General Purpose Event0 Status Register

地址偏移	电压域	属性
0x28	ACPI	R/WC
位域	描述	
31:16	<b>GPI_STS - R/WC</b> 16 位寄存器, 由高到低分别对应引脚 ACPI_GPIO15-0 0: 软件写 1 将该位清除。 1: 当 ACPI_GPIO 发生中断事件时这些位被置位, 当 GPI_EN 位有效时, 产生唤醒事件或中断。	
15:10	<b>USB[5:0]_STS - R/WC.</b> 只有第 10 位有意义, 15:11 位暂无意义。 0: 软件写 1 将该位清除。 1: 当 USB 发生 wake 事件时这些位被置位, 当 USBx_EN 位有效时, 产生唤醒事件或中断。	
9:7	保留	
6	<b>GMAC1_STS - R/WC.</b> 0: 软件写 1 将该位清除。 1: 当 GMAC1 发生 wake 事件时这些位被置位, 当 GMAC1_EN 位有效时, 产生唤醒事件或中断。	
5	<b>GMACO_STS - R/WC.</b> 0: 软件写 1 将该位清除。 1: 当 GMACO 发生 wake 事件时这些位被置位, 当 GMACO_EN 位有效时, 产生唤醒事件或中断。	
4	保留	
3	<b>CTW_STS - R/WC.</b> thermal warning 发生	
2	<b>CTA_STS - R/WC.</b> thermal alert 发生	
1:0	保留	

**GPE0\_EN : General Purpose Event0 Enable Register**

地址偏移	电压域	属性
0x2C	ACPI/RTC	R/W
位域	描述	电压域
31:16	GPI_EN - R/W 16 位寄存器，由高到低分别对应引脚 ACPI_GPIO15-0 0: 无效 1: 使能 GPIO 中断唤醒事件，GPIO 中断类型可配置	
15:10	USB[5:0]_EN - R/W. 0: 无效 1: 使能 USBx_STS 产生唤醒事件，当回到 S0 将产生中断信号。	
9:7	保留	
6	GMAC1_EN - R/W. 0: 无效 1: 使能 GMAC1_STS 产生唤醒事件，当回到 S0 将产生中断信号。	RTC
5	GMACO_EN - R/W. 0: 无效 1: 使能 GMACO_STS 产生唤醒事件，当回到 S0 将产生中断信号。	
4	保留	
3	CTW_EN - R/W 使能 THERMAL WARNING 产生中断。	
2	CTA_EN - R/W 使能 THERMAL ALERT 产生中断。	
1:0	保留	

**RST\_CNT : Reset Control Register**

地址偏移	电压域	属性
0x30	SOC	R/W
位域	描述	
31:2	保留	
1	WD_EN - R/W Watch dog 功能使能	
0	OS_RST - R/W 软件写该位使系统复位。	

**WD\_SET : Watch Dog Set Register**

地址偏移	电压域	属性
0x34	SOC	WO
位域	描述	
31:1	保留	
0	当 WD_EN 为 1 时，写该位将重填内部 watch dog 计数器，充填的值为 WD_Timer。 注意，watch dog 计数器的工作频率为 50MHz。	

### WD\_Timer : Watch Dog Timer Register

地址偏移	电压域	属性
0x38	SOC	R/W
位域	描述	
31:0	该寄存器的值为 watch dog 重填的值，复位值为全 1。	

### GEN\_RTC\_1 : General RTC Register 1

地址偏移	电压域	属性
0x50	RTC	R/W
位域	描述	
31:0	RTC 通用寄存器	

### GEN\_RTC\_2 : General RTC Register 2

地址偏移	电压域	属性
0x54	RTC	R/W
位域	描述	
31:0	RTC 通用寄存器	

### ACPI\_GPIO\_0 : ACPI 域 GPIO 输出

地址偏移	电压域	属性
0x80	ACPI	R/W
位域	描述	
31:16	保留	
15:0	ACPI 域 GPIO 输出值	

### ACPI\_GPIO\_OEN : ACPI 域 GPIO 输出使能

地址偏移	电压域	属性
0x84	ACPI	R/W
位域	描述	
31:16	保留	
15:0	ACPI 域 GPIO 输出使能控制，低有效	

### ACPI\_GPIO\_I : ACPI 域 GPIO 输入

地址偏移	电压域	属性
0x88	ACPI	R/W
位域	描述	

31:16	保留
15:0	ACPI 域 GPIO 输入值

**ACPI\_GPIO\_POL : ACPI 域 GPIO 中断极性**

地址偏移	电压域	属性
0x90	ACPI	R/W
位域	描述	
31:16	保留	
15:0	ACPI 域 GPIO 中断极性控制, 1 代表高电平/上升沿, 0 代表低电平/下降沿	

**ACPI\_GPIO\_EDGE : ACPI 域 GPIO 中断边沿设置**

地址偏移	电压域	属性
0x94	ACPI	R/W
位域	描述	
31:16	保留	
15:0	ACPI 域 GPIO 中断边沿设置, 1 代表边沿类型中断, 0 代表电平类型中断	

**ACPI\_GPIO\_DUALEDGE : ACPI 域 GPIO 中断双沿设置**

地址偏移	电压域	属性
0x98	ACPI	R/W
位域	描述	
31:16	保留	
15:0	ACPI 域 GPIO 中断是否为双沿触发, 1 代表双沿触发, 0 代表单沿触发	

## 15 RTC

实时时钟（RTC）单元可以在主板上电后进行配置，当主板断电后，该单元仍然运作，可以仅靠板上的电池供电就正常运行。RTC 单元运行时电流仅几个微安。

RTC 包含振荡器，结合外部 32.768KHZ 晶体产生工作时钟。该时钟用于时间信息的维护以及产生各种定时和计数中断。

RTC 模块中包含两个计数器，分别为 TOY（Time of Year）计数器和 RTC 计数器。其中 TOY 计数器按年月日时分秒计数，精度为以 0.1 秒；RTC 计数器以 32.768KHz 时钟计数，宽度为 32 位。

### 15.1 访问地址

RTC 模块的访问基地址为 MISC 低速设备块的基地址加偏移 0x50100。RTC 模块的内部寄存器物理地址构成如下：

表 15-1 RTC 访问地址

地址位	构成	备注
[15:9]	0	保留
[8]	1	保留
[7:0]	REG	内部寄存器地址

名称	偏移地址	位宽	R/W	描述
sys_toytrim	0x20	32	R/W	软件必须初始化为 0
sys_toywrite0	0x24	32	W	TOY 低 32 位数值写入
sys_toywrite1	0x28	32	W	TOY 高 32 位数值写入
sys_toyread0	0x2C	32	R	TOY 低 32 位数值读出
sys_toyread1	0x30	32	R	TOY 高 32 位数值读出
sys_toymatch0	0x34	32	R/W	TOY 定时中断 0
sys_toymatch1	0x38	32	R/W	TOY 定时中断 1
sys_toymatch2	0x3C	32	R/W	TOY 定时中断 2
sys_rtctrl	0x40	32	R/W	TOY 和 RTC 控制寄存器 软件必须初始化
sys_rtctrim	0x60	32	R/W	软件必须初始化为 0
sys_rtctime0	0x64	32	W	RTC 定时计数写入
sys_rtctime0	0x68	32	R	RTC 定时计数读出
sys_rtcmatch0	0x6C	32	R/W	RTC 时钟定时中断 0
sys_rtcmatch1	0x70	32	R/W	RTC 时钟定时中断 1
sys_rtcmatch2	0x74	32	R/W	RTC 时钟定时中断 2



## 15.2 寄存器地址列表

## 15.3 寄存器描述

### 15.3.1 SYS\_TOYWRITE0

寄存器位宽： [31: 0]

偏移量： 0x24

复位值： 0x00000000

位域	位域名称	访问	缺省	描述
31:26	TOY_MONTH	W		月，范围 1~12
25:21	TOY_DAY	W		日，范围 1~31
20:16	TOY_HOUR	W		小时，范围 0~23
15:10	TOY_MIN	W		分，范围 0~59
9:4	TOY_SEC	W		秒，范围 0~59
3:0	TOY_MILLISEC	W		0.1 秒，范围 0~9

### 15.3.2 SYS\_TOYWRITE1

寄存器位宽： [31: 0]

偏移量： 0x28

复位值： 0x00000000

位域	位域名称	访问	缺省	描述
31:0	TOY_YEAR	W		年，范围 0~16383

### 15.3.3 SYS\_TOYREAD0

寄存器位宽： [31: 0]

偏移量： 0x2C

复位值： 0x00000000

位域	位域名称	访问	缺省	描述
31:26	TOY_MONTH	R		月，范围 1~12
25:21	TOY_DAY	R		日，范围 1~31
20:16	TOY_HOUR	R		小时，范围 0~23
15:10	TOY_MIN	R		分，范围 0~59
9:4	TOY_SEC	R		秒，范围 0~59
3:0	TOY_MILLISEC	R		0.1 秒，范围 0~9

### 15.3.4 SYS\_TOYREAD1

寄存器位宽： [31: 0]

偏移量： 0x30

复位值： 0x00000000

位域	位域名称	访问	缺省	描述
31:0	TOY_YEAR	R		年, 范围 0~16383

### 15.3.5 SYS\_TOYMATCH0/1/2

寄存器位宽： [31: 0]

偏移量： 0x34/38/3C

复位值： 0x00000000

位域	位域名称	访问	缺省	描述
31:26	YEAR	R/W		年, 范围 0~16383
25:22	MONTH	R/W		月, 范围 1~12
21:17	DAY	R/W		日, 范围 1~31
16:12	HOUR	R/W		小时, 范围 0~23
11:6	MIN	R/W		分, 范围 0~59
5:0	SEC	R/W		秒, 范围 0~59

### 15.3.6 SYS\_RTCCTRL

寄存器位宽： [31: 0]

偏移量： 0x40

复位值： 无

位域	位域名称	访问	缺省	描述
31:24	保留	R	0	保留, 置 0
23	ERS	R	0	REN(bit13) 写状态
22:21	保留	R	0	保留, 置 0
20	RTS	R	0	Sys_rtctrim 写状态
19	RM2	R	0	Sys_rtcmatch2 写状态
18	RM1	R	0	Sys_rtcmatch1 写状态
17	RM0	R	0	Sys_rtcmatch0 写状态
16	RS	R	0	Sys_rtcwrite 写状态
15	保留	R	0	保留, 置 0
14	保留	R	0	保留, 置 0
13	REN	R/W	0	RTC 使能, 高有效。需要初始化为 1

12	保留	R	0	保留, 置 0
11	TEN	R/W	0	TOY 使能, 高有效。需要初始化为 1
10	保留	R	0	保留, 置 0
9	保留	R	0	保留, 置 0
8	EO	R/W	0	0: 32.768k 晶振禁止; 1: 32.768k 晶振使能
7	保留	R	0	保留, 置 0
6	保留	R	0	保留, 置 0
5	32S	R	0	0: 32.768k 晶振不工作; 1: 32.768k 晶振正常工作。
4	保留	R	0	保留, 置 0
3	TM2	R	0	Sys_toymatch2 写状态
2	TM1	R	0	Sys_toymatch1 写状态
1	TM0	R	0	Sys_toymatch0 写状态
0	TS	R	0	Sys_toywrite 写状态

### 15.3.7 SYS\_RTCWRITE

寄存器位宽: [31: 0]

偏移量: 0x64

复位值: 0x00000000

位域	位域名称	访问	缺省	描述
31:0	RTC	W		

### 15.3.8 SYS\_RTCREAD

寄存器位宽: [31: 0]

偏移量: 0x68

复位值: 0x00000000

位域	位域名称	访问	缺省	描述
31:0	RTC	R		

### 15.3.9 SYS\_RTCMATCH0/1/2

寄存器位宽: [31: 0]

偏移量: 0x6C/70/74

复位值: 0x00000000

位域	位域名称	访问	缺省	描述
31:26	RTC	R/W		

## 16 GPIO

桥片共有 58 个 GPIO 引脚，2 个为专用 GPIO，其余 56 个与其他功能复用。GPIO 引脚由一组寄存器控制，包括：GPIO 方向控制（GPIO\_OEN）、GPIO 输出值（GPIO\_O）、GPIO 输入值（GPIO\_I）、GPIO 输入中断使能控制（GPIO\_INT\_EN）、GPIO 输入中断极性控制（GPIO\_INT\_POL）、GPIO 输入中断边沿性控制（GPIO\_INT\_EDGE）、GPIO 输入中断清除（GPIO\_INT\_CLR）、GPIO 输入中断状态（GPIO\_INT\_STS）。

表 16-1 GPIO 控制寄存器

寄存器	大小（位）	描述															
GPIO_OEN	1	GPIO 输出使能，低有效。															
GPIO_O	1	GPIO 输出值。															
GPIO_I	1	GPIO 输入值。															
GPIO_INT_EN	1	GPIO 中断使能。															
GPIO_INT_POL	1	GPIO 中断极性。															
GPIO_INT_EDGE	1	GPIO 中断边沿性。与 GPIO 中断极性配合控制 GPIO 中断状态的产生。															
		<table border="1"> <thead> <tr> <th>POL</th> <th>EDGE</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>低电平触发中断</td> </tr> <tr> <td>1</td> <td>0</td> <td>高电平触发中断</td> </tr> <tr> <td>0</td> <td>1</td> <td>下降沿触发中断</td> </tr> <tr> <td>1</td> <td>1</td> <td>上升沿触发中断</td> </tr> </tbody> </table>	POL	EDGE	描述	0	0	低电平触发中断	1	0	高电平触发中断	0	1	下降沿触发中断	1	1	上升沿触发中断
		POL	EDGE	描述													
		0	0	低电平触发中断													
		1	0	高电平触发中断													
0	1	下降沿触发中断															
1	1	上升沿触发中断															
GPIO_INT_CLR	1	GPIO 中断状态清除															
GPIO_INT_STS	1	GPIO 中断状态															

### 16.1 访问地址

GPIO 的访问基地址等于 MISC 低速设备块的基地址加偏移 0x60000。

桥片提供了两种方式来控制 GPIO 引脚。一种是按位控制每个 GPIO 引脚，一种是按字节控制每个 GPIO 引脚。桥片是通过提供两个地址空间来映射 GPIO 控制寄存器实现该功能的。一种是按位映射，一种是按字节来索引控制寄存器的每个比特位。对应的，GPIO 内部的地址空间也分为两部分。

推荐使用后一种方式来控制器 GPIO 引脚。

GPIO 模块的内部寄存器物理地址构成如下：

表 16-2 GPIO 模块内部寄存器物理地址

地址空间	说明
0x800-0xF00	按字节控制寄存器地址
0x0 - 0x70	按位控制寄存器地址

表 16-3 按位控制 GPIO 配置寄存器地址

地址偏移	寄存器	大小 (位)	描述
0x00	GPIO_OEN	58	GPIO 输出使能, 低有效。每位控制一个 GPIO 引脚。
0x10	GPIO_0	58	GPIO 输出值。每位控制一个 GPIO 引脚。
0x20	GPIO_I	58	GPIO 输入值。每位控制一个 GPIO 引脚。
0x30	GPIO_INT_EN	58	GPIO 中断使能。每位控制一个 GPIO 引脚。
0x40	GPIO_INT_POL	58	GPIO 中断极性。每位控制一个 GPIO 引脚。
0x50	GPIO_INT_EDGE	58	GPIO 中断边沿性。每位控制一个 GPIO 引脚。
0x60	GPIO_INT_CLR	58	GPIO 中断清除。每位控制一个 GPIO 引脚。
0x70	GPIO_INT_STS	58	GPIO 中断状态。每位控制一个 GPIO 引脚。

表 16-4 按字节控制 GPIO 配置寄存器地址

地址偏移	寄存器	大小 (字节)	描述
0x800	GPIO_OEN	58	GPIO 输出使能, 低有效。每个字节控制一个 GPIO 引脚。
0x900	GPIO_0	58	GPIO 输出值。每个字节控制一个 GPIO 引脚。
0xA00	GPIO_I	58	GPIO 输入值。每个字节控制一个 GPIO 引脚。
0xB00	GPIO_INT_EN	58	GPIO 中断使能。每个字节控制一个 GPIO 引脚。
0xC00	GPIO_INT_POL	58	GPIO 中断极性。每个字节控制一个 GPIO 引脚。
0xD00	GPIO_INT_EDGE	58	GPIO 中断边沿性。每个字节控制一个 GPIO 引脚。
0xE00	GPIO_INT_CLR	58	GPIO 中断清除。每个字节控制一个 GPIO 引脚。
0xF00	GPIO_INT_STS	58	GPIO 中断状态。每个字节控制一个 GPIO 引脚。

## 16.2 控制寄存器

### GPIO 方向控制

地址偏移: 00-03h 属性: R/W

默认值: FFFFFFF0h 大小: 4

位域	名称	访问	描述
31:0	GPIO_OEN	R/W	对应于 GPIO[31:0]的方向控制。 0: 输出 1: 输入

地址偏移: 04-07h 属性: R/W

默认值: FFFFFFFFh 大小: 4

位域	名称	访问	描述
31:26	Reserved	R/W	保留

25:0	GPIO_OEN	R/W	对应于 GPIO[57:32]的方向控制。 0: 输出 1: 输入
------	----------	-----	---

### GPIO 输出值

地址偏移: 10-13h                      属性: R/W

默认值: 0000000Fh                    大小: 4

位域	名称	访问	描述
31:0	GPIO_O	R/W	对应于 GPIO[31:0]的输出值。

地址偏移: 14-17h                      属性: R/W

默认值: 00000000h                    大小: 4

位域	名称	访问	描述
31:26	Reserved	R/W	保留
25:0	GPIO_O	R/W	对应于 GPIO[57:32]的输出值。

### GPIO 输入值

地址偏移: 20-23h                      属性: RO

默认值: N/A                            大小: 4

位域	名称	访问	描述
31:0	GPIO_I	RO	对应于 GPIO[31:0]的输入值。

地址偏移: 24-27h                      属性: RO

默认值: 00000000h                    大小: 4

位域	名称	访问	描述
31:26	Reserved	R/W	保留
25:0	GPIO_I	RO	对应于 GPIO[57:32]的输入值。

### GPIO 中断使能

地址偏移: 30-33h                      属性: R/W

默认值: 00000000h                    大小: 4

位域	名称	访问	描述
31:0	GPIO_INT_EN	R/W	对应于 GPIO[31:0]的输入中断使能。 0: 关闭中断 1: 使能中断

地址偏移: 34-37h                      属性: R/W

默认值: 00000000h                    大小: 4

位域	名称	访问	描述
31:26	Reserved	R/W	保留
25:0	GPIO_INT_EN	R/W	对应于 GPIO[57:32] 的输入中断使能。 0: 关闭中断 1: 使能中断

### GPIO 中断极性

地址偏移: 40-43h 属性: R/W

默认值: 00000000h 大小: 4

位域	名称	访问	描述
31:0	GPIO_INT_POL	R/W	对应于 GPIO[31:0] 的中断极性。 与中断边沿性配合, 组成四种中断触发方式, 见下文 GPIO 中断边沿。

地址偏移: 44-47h 属性: R/W

默认值: 00000000h 大小: 4

位域	名称	访问	描述
31:26	Reserved	R/W	保留
25:0	GPIO_INT_POL	R/W	对应于 GPIO[57:32] 的中断极性。 与中断边沿性配合, 组成四种中断触发方式, 见下文 GPIO 中断边沿。

### GPIO 中断边沿

地址偏移: 50-53h 属性: R/W

默认值: 00000000h 大小: 4

位域	名称	访问	描述		
31:0	GPIO_INT_EDGE	R/W	对应于 GPIO[31:0] 的中断边沿。 与中断极性配合, 组成四种中断触发方式。		
			POL	EDGE	描述
			0	0	低电平触发中断
			1	0	高电平触发中断
			0	1	下降沿触发中断
1	1	上升沿触发中断			

地址偏移: 54-57h 属性: R/W

默认值: 00000000h 大小: 4

位域	名称	访问	描述
31:26	Reserved	R/W	保留

25:0	GPIO_INT_EDGE	R/W	对应于 GPIO[57:32] 的中断边沿。 与中断极性配合，组成四种中断触发方式。		
			POL	EDGE	描述
			0	0	低电平触发中断
			1	0	高电平触发中断
			0	1	下降沿触发中断
1	1	上升沿触发中断			

### GPIO 中断清除

地址偏移：60-63h 属性：R/W

默认值：00000000h 大小：4

位域	名称	访问	描述
31:0	GPIO_INT_CLR	R/W	对应于 GPIO[31:0] 的中断清除。 写 1 清除相应 GPIO 位上的中断，随后硬件会自动置 0 中断清除寄存器相应位，不需软件再作处理。

地址偏移：64-67h 属性：R/W

默认值：00000000h 大小：4

位域	名称	访问	描述
31:26	Reserved	R/W	保留
25:0	GPIO_INT_CLR	R/W	对应于 GPIO[57:32] 的中断极性。 写 1 清除相应 GPIO 位上的中断，随后硬件会自动置 0 中断清除寄存器相应位，不需软件再作处理。

### GPIO 中断状态

地址偏移：70-73h 属性：R/W

默认值：00000000h 大小：4

位域	名称	访问	描述
31:0	GPIO_INT_STS	R/W	对应于 GPIO[31:0] 的中断状态。 1：有中断 0：无中断

地址偏移：74-77h 属性：R/W

默认值：00000000h 大小：4

位域	名称	访问	描述
31:26	Reserved	R/W	保留
25:0	GPIO_INT_STS	R/W	对应于 GPIO[57:32] 的中断状态。 1：有中断 0：无中断



## 17 GMAC 控制器 (D3:F0)

桥片集成了一个 GMAC 控制器，内部集成 PHY，为 Device 3 的功能 0。

### 17.1 GMAC 配置寄存器 (D3:F0)

表 17-1 GMAC 控制器的 PCI 配置头

地址偏移	简称	描述	默认值	访问类型
00h-01h	VID	Vendor ID	0014h	RO
02h-03h	DID	Device ID	7A13h	RO
04h-05h	PCICMD	PCI Command	0000h	R/W, RO
08h	RID	Revision ID	00h	RO
09h	PI	Programming Interface	00h	RO
0Ah	SCC	Sub Class Code	00h	RO
0Bh	BCC	Base Class Code	02h	RO
0Ch	CLS	Cache Line Size	10h	RO
0Eh	HEADTYP	Header Type	00h	RO
10h-17h	CNL_BAR	Control Block Base Address Register	0000000000000004h	R/W,RO
2Ch-2Dh	SVID	Subsystem Vendor ID	0000h	RO
2Eh-2Fh	SID	Subsystem Identification	0000h	RO
3Ch	INT_LN	Interrupt Line	FFh	R/W
3Dh	INT_PN	Interrupt Pin	00h	RO

注：表中未列出的地址空间表示保留。

下面列出与 PCI 配置头规范稍有不同的寄存器及其描述。

#### PCICMD-PCI 命令寄存器 (GMAC-D3:F0)

地址偏移：04-05h

属性：R/W, RO

默认值：0000h

大小：16 位

位域	名称	访问	描述
15:2	Reserved	RO	保留
1	Memory Space Enable	R/W	该位用来控制是否使能对 GMAC 控制寄存器的访问。 0：禁止访问； 1：使能对 GMAC 控制寄存器的访问。在将该位配置为 1 之前，必须先配置 PCNL_BAR 寄存器。
0	Reserved	RO	保留

## 18 USB 控制器（D4-5:F0/1, D25:F0）

桥片的 USB 主机端口特性如下：

- 兼容 USB Rev 1.1、USB Rev 2.0 和 USB Rev 3.0 协议
- 兼容 OHCI Rev 1.0、EHCI Rev 1.0、XHCI Rev1.1 协议
- 支持四个 USB3.0 端口，每个端口都可挂 SS、LS、FS 或 HS 设备
- 支持八个 USB2.0 端口，每个端口都可挂 LS、FS 或 HS 设备

桥片内置 3 个 USB 控制器(编号 USB20、USB21 和 USB3)。其中 USB20 为 EHCI-OHCI 控制器(D4:F0/1)，对应 usb2 的 port4-7；USB21 为 EHCI-OHCI 控制器(D5:F0/1)，对应 usb2 的 port8-11；USB3 为 XHCI 控制器(D23:F0)，对应 usb3 的 port0-3 和 usb2 的 port0-3。

需注意，USB3.0 端口不支持远程唤醒操作系统的功能，如需使用该功能，请将支持远程唤醒的设备接在 USB2.0 的端口上。

### 18.1 EHCI 控制器

#### 18.1.1 EHCI 配置寄存器（D4:F1, D5:F1）

表 18-1 USB-EHCI 控制器的 PCI 配置头

地址偏移	简称	描述	默认值	访问类型
00h-01h	VID	Vendor ID	0014h	RO
02h-03h	DID	Device ID	7A14h	RO
04h-05h	PCICMD	PCI Command	0000h	R/W, RO
08h	RID	Revision ID	02h	RO
09h	PI	Programming Interface	20h	RO
0Ah	SCC	Sub Class Code	03h	RO
0Bh	BCC	Base Class Code	0Ch	RO
0Ch	CLS	Cache Line Size	10h	RO
0Eh	HEADTYP	Header Type	80h	RO
10h-17h	CNL_BAR	Control Block Base Address Register	0000000000000004h	R/W,RO
2Ch-2Dh	SVID	Subsystem Vendor ID	0000h	RO
2Eh-2Fh	SID	Subsystem Identification	0000h	RO
3Ch	INT_LN	Interrupt Line	FFh	R/W
3Dh	INT_PN	Interrupt Pin	00h	RO

注：表中未列出的地址空间表示保留。

下面列出与 PCI 配置头规范稍有不同的寄存器及其描述。

## PCICMD-PCI 命令寄存器 (USB EHCI-D4:F1, D5:F1)

地址偏移: 04-05h

属性: R/W, RO

默认值: 0000h

大小: 16 位

位域	名称	访问	描述
15:2	Reserved	RO	保留
1	Memory Space Enable	R/W	该位用来控制是否使能对 USB EHCI 控制寄存器的访问。 0: 禁止访问; 1: 使能对 USB EHCI 控制寄存器的访问。在将该位配置为 1 之前, 必须先配置 PCNL BAR 寄存器。
0	Reserved	RO	保留

EHCI 的相关寄存器包括 Capability 寄存器、Operational 寄存器和, EHCI 实现相关寄存器。桥片的 USB 主机控制器兼容 EHCI Rev 1.0 协议, Capability 寄存器和 Operational 寄存器的详细信息参照 Enhanced Host Controller Interface Rev 1.0 Specification。

### 18.1.2 Capability 寄存器

表 18-2 EHCI Capability 寄存器列表

名称	偏移地址	宽度	访问	说明
HCCAPBASE	0x00	32	RO	默认值为 32'h01000010
HCSPARAMS	0x04	32	RO	默认值为 32'h00001116
HCCPARAMS	0x08	32	RO	默认值为 32'h0000A010

### 18.1.3 Operational 寄存器

表 18-3 EHCI Operational 寄存器列表

名称	偏移地址	宽度	访问	说明
USBCMD	0x10	32	R/W、RO	USB 主机控制器的命令寄存器
USBSTS	0x14	32	R/W、RO	USB 主机控制器的状态寄存器
USBINTR	0x18	32	R/W	USB 主机控制器的中断设置寄存器
FRINDEX	0x1c	32	R/W	USB 主机控制器的帧索引寄存器
CTRLDSSEGMENT	0x20	32	R/W	存放 EHCI 控制数据结构的地址
PERIODICLISTBASE	0x24	32	R/W	存放周期数据帧表的起始地址
ASYNCLISTADDR	0x28	32	R/W	存放下一个要被执行的异步队列的起始地址
CONFIGFLAG	0x50	32	R/W	配置模式寄存器
PORTSC 1	0x54	32	R/W、RO	端口 1 状态和控制寄存器
PORTSC 2	0x58	32	R/W、RO	端口 2 状态和控制寄存器
PORTSC 3	0x5c	32	R/W、RO	端口 3 状态和控制寄存器
PORTSC 4	0x60	32	R/W、RO	端口 4 状态和控制寄存器
PORTSC 5	0x64	32	R/W、RO	端口 5 状态和控制寄存器

PORTSC 6	0x88	32	R/W、RO	端口 6 状态和控制寄存器
----------	------	----	--------	---------------

注：USBOPBase 固定为 EHCI 的起始地址+ 10h

## 18.2 OHCI 控制器

### 18.2.1 OHCI 配置寄存器 (D4:F0, D5:F0)

表 18-4 USB-OHCI 控制器的 PCI 配置头

地址偏移	简称	描述	默认值	访问类型
00h-01h	VID	Vendor ID	0014h	RO
02h-03h	DID	Device ID	7A24h	RO
04h-05h	PCICMD	PCI Command	0000h	R/W, RO
08h	RID	Revision ID	02h	RO
09h	PI	Programming Interface	10h	RO
0Ah	SCC	Sub Class Code	03h	RO
0Bh	BCC	Base Class Code	0Ch	RO
0Ch	CLS	Cache Line Size	10h	RO
0Eh	HEADTYP	Header Type	80h	RO
10h-17h	CNL_BAR	Control Block Base Address Register	0000000000000004h	R/W,RO
2Ch-2Dh	SVID	Subsystem Vendor ID	0000h	RO
2Eh-2Fh	SID	Subsystem Identification	0000h	RO
3Ch	INT_LN	Interrupt Line	FFh	R/W
3Dh	INT_PN	Interrupt Pin	00h	RO

注：表中未列出的地址空间表示保留。

下面列出与 PCI 配置头规范稍有不同的寄存器及其描述。

#### PCICMD-PCI 命令寄存器 (USB OHCI-D4:F0, D5:F0)

地址偏移：04-05h

属性：R/W, RO

默认值：0000h

大小：16 位

位域	名称	访问	描述
15:2	Reserved	RO	保留
1	Memory Space Enable	R/W	该位用来控制是否使能对 USB OHCI 控制寄存器的访问。 0：禁止访问； 1：使能对 USB OHCI 控制寄存器的访问。在将该位配置为 1 之前，必须先配置 PCNL_BAR 寄存器。
0	Reserved	RO	保留

OHCI 的相关寄存器包括 Operational 寄存器和 OHCI 实现相关寄存器。桥片的 USB 主机控制器兼容 OHCI Rev 1.0 协议，Operational 寄存器的详细信息参照 Open Host Controller Interface Rev 1.0 Specification

## 18.2.2 Operational 寄存器

表 18-5 USB-OHCI Operational 寄存器

名称	偏移地址	宽度	访问	说明
HcRevision	0x00	32	-	控制和状态
HcControl	0x04	32	-	
HcCommonStatus	0x08	32	-	
HcInterruptStatus	0x0C	32	-	
HcInterruptEnable	0x10	32	-	
HcInterruptDisable	0x14	32	--	
HcHCCA	0x18	32	-	内存指针
HcPeriodCurrentED	0x1C	32	-	
HcControlHeadED	0x20	32	-	
HcControlCurrentED	0x24	32	-	
HcBulkHeadED	0x28	32	-	
HcBulkCurrentED	0x2C	32	-	
HcDoneHead	0x30	32	-	帧计数器
HcRmInterval	0x34	32	-	
HcFmRemaining	0x38	32	-	
HcFmNumber	0x3C	32	-	
HcPeriodicStart	0x40	32	-	
HcLSThreshold	0x44	32	-	根集线器
HcRhDescriptorA	0x48	32	-	
HcRhDescriptorB	0x04C	32	-	
HcRhStatus	0x50	32	-	
HcRhPortStatus1	0x54	32	-	
HcRhPortStatus2	0x58	32	-	

## 18.3 XHCI 控制器

### 18.3.1 XHCI 配置寄存器 (D25:F0)

表 18-6 USB-XHCI 控制器的 PCI 配置头

地址偏移	简称	描述	默认值	访问类型
00h-01h	VID	Vendor ID	0014h	RO

02h-03h	DID	Device ID	7A34h	RO
04h-05h	PCICMD	PCI Command	0000h	R/W, RO
08h	RID	Revision ID	00h	RO
09h	PI	Programming Interface	30h	RO
0Ah	SCC	Sub Class Code	03h	RO
0Bh	BCC	Base Class Code	0Ch	RO
0Ch	CLS	Cache Line Size	10h	RO
0Eh	HEADTYP	Header Type	00h	RO
10h-17h	CNL_BAR	Control Block Base Address Register	000000000000004h	R/W,RO
2Ch-2Dh	SVID	Subsystem Vendor ID	0000h	RO
2Eh-2Fh	SID	Subsystem Identification	0000h	RO
3Ch	INT_LN	Interrupt Line	FFh	R/W
3Dh	INT_PN	Interrupt Pin	00h	RO

**注：**表中未列出的地址空间表示保留。

下面列出与 PCI 配置头规范稍有不同的寄存器及其描述。

### PCICMD-PCI 命令寄存器 (USB XHCI-D25:F0)

地址偏移：04-05h

属性：R/W, RO

默认值：0000h

大小：16 位

位域	名称	访问	描述
15:2	Reserved	RO	保留
1	Memory Space Enable	R/W	该位用来控制是否使能对 USB XHCI 控制寄存器的访问。 0：禁止访问； 1：使能对 USB XHCI 控制寄存器的访问。在将该位配置为 1 之前，必须先配置 PCNL_BAR 寄存器。
0	Reserved	RO	保留

## 19 图形处理器 (D6:F0)

### 19.1 GPU 配置寄存器 (D6:F0)

表 19-1 GPU 控制器的 PCI 配置头

地址偏移	简称	描述	默认值	访问类型
00h-01h	VID	Vendor ID	0014h	RO
02h-03h	DID	Device ID	7A25h	RO
04h-05h	PCICMD	PCI Command	0000h	R/W, RO
08h	RID	Revision ID	01h	RO
09h	PI	Programming Interface	00h	RO
0Ah	SCC	Sub Class Code	02h	RO
0Bh	BCC	Base Class Code	03h	RO
0Ch	CLS	Cache Line Size	10h	RO
0Eh	HEADTYP	Header Type	80h	RO
10h-17h	CNL_BAR	Control Block Base Address Register	0000000000000004h	R/W,RO
18h-1Fh	GMEM_BAR	Graphic Memory Base Address Register	0000000000000004h	R/W,RO
2Ch-2Dh	SVID	Subsystem Vendor ID	0000h	RO
2Eh-2Fh	SID	Subsystem Identification	0000h	RO
3Ch	INT_LN	Interrupt Line	FFh	R/W
3Dh	INT_PN	Interrupt Pin	00h	RO

注：表中未列出的地址空间表示保留。

下面列出与 PCI 配置头规范稍有不同的寄存器及其描述。

#### PCICMD-PCI 命令寄存器 (GPU-D6:F0)

地址偏移：04-05h

属性：R/W, RO

默认值：0000h

大小：16 位

位域	名称	访问	描述
15:2	Reserved	RO	保留
1	Memory Space Enable	R/W	该位用来控制是否使能对 GPU 控制寄存器的访问。 0：禁止访问； 1：使能对 GPU 控制寄存器的访问。在将该位配置为 1 之前，必须先配置 PCNL_BAR 寄存器。
0	Reserved	RO	保留

### 19.2 DDR4 显存接口

桥片的显存接口遵守 DDR4 SDRAM 行业标准 (JESD79-4)。

## 访问地址

DDR4 显存控制器包括两个地址空间：显存控制器的寄存器配置空间和显存的存储空间。当配置参数 `mc_disable_reg = 0` 时，发给显存的访问都为寄存器配置访问；当配置参数 `mc_disable_reg = 1` 时，发给显存的访问都为显存的读写访问。

## 19.3 DDR4 SDRAM 控制器功能概述

显存支持的片选个数为 1，一共含有 22 位的地址总线（即：18 位的行列地址总线、2 位逻辑 Bank 总线和 2 位逻辑 Bank Group 总线，其中行列地址总线与 RASn、CASn 和 Wen 复用）。在具体选择使用不同内存芯片类型时，可以调整 DDR4 控制器参数设置进行支持。其中，行地址（ROW）数为 18，列地址（COL）数为 12。

桥片的显存内存控制器具有如下特征：

- 接口上命令、读写数据全流水操作；
- 内存命令合并、排序提高整体带宽；
- 配置寄存器读写端口，可以修改内存设备的基本参数；
- 内建动态延迟补偿电路（DCC），用于数据的可靠发送和接收；
- 支持 DDR4 SDRAM，且参数配置支持 x4、x8、x16 颗粒；
- 控制器与 PHY 频率比 1/2 ；
- 支持数据传输速率范围为 800Mbps-2400Mbps。

## 19.4 DDR4 SDRAM 参数配置格式

内存控制器的参数列表

表 19-2 内存控制器软件可见参数列表

Offset	63:55	55:48	47:40	39:32	31:24	23:16	15:8	7:0
<b>PHY</b>								
0x0000							version(RD)	
0x0008		switch_byte38	x4_mode	ddr3_mode			capability (RD)	
0x0010							dram_init(RD)	init_start
0x0018								
0x0020							preamble2	rdfifo_valid
0x0028		rdfifo_empty(RD)				Overflow(RD)		
0x0030		dll_value(RD)	dll_init_done(RD)	dll_lock_mode	dll_bypass	dll_adj_cnt	dll_increment	dll_start_point
0x0038				dll_dbl_fix			dll_close_disable	dll_ck
0x0040								
0x0048							clken_ckca	
0x0050								
0x0058							clken_ds_0	
0x0060								
0x0068							clken_ds_1	



0x0070								
0x0078							clken_ds_2	
0x0080								
0x0088							clken_ds_3	
0x0090								
0x0098							clken_ds_4	
0x00a0								
0x00a8							clken_ds_5	
0x00b0								
0x00b8							clken_ds_6	
0x00c0								
0x00c8							clken_ds_7	
0x00d0								
0x00d8							clken_ds_8	
0x00e0			vrefclk_inv	vref_sample		vref_num	vref_dly	dll_vref
.....								
0x0100					dll_1xdly_0	dll_1xgen_0	dll_wrdds_0	dll_wrddq_0
0x0108						dll_gate_0	dll_rdds1_0	dll_rdds0_0
0x0110	rdodt_ctrl_0	rdgate_len_0	rdgate_mode_0	rdgate_ctrl_0			dqs_oe_ctrl_0	dq_oe_ctrl_0
0x0118					dly_2x_0		redge_sel_0	rdds_phase_0(RD)
0x0120	w_bdly0_0[31:28]	w_bdly0_0[27:24]	w_bdly0_0[23:20]	w_bdly0_0[19:16]	w_bdly0_0[15:12]	w_bdly0_0[11:8]	w_bdly0_0[7:4]	w_bdly0_0[3:0]
0x0128		w_bdly0_0[59:56]	w_bdly0_0[55:52]	w_bdly0_0[51:48]	w_bdly0_0[47:44]	w_bdly0_0[43:40]	w_bdly0_0[39:36]	w_bdly0_0[35:32]
0x0130	w_bdly1_0[24:21]	w_bdly1_0[20:18]	w_bdly1_0[17:15]	w_bdly1_0[14:12]	w_bdly1_0[11:9]	w_bdly1_0[8:6]	w_bdly1_0[5:3]	w_bdly1_0[2:0]
0x0138								w_bdly1_0[27:26]
0x0140							rg_bdly_0[7:4]	rg_bdly_0[3:0]
0x0148								
0x0150	rdqsp_bdly_0[31:28]	rdqsp_bdly_0[27:24]	rdqsp_bdly_0[23:20]	rdqsp_bdly_0[19:16]	rdqsp_bdly_0[15:12]	rdqsp_bdly_0[11:8]	rdqsp_bdly_0[7:4]	rdqsp_bdly_0[3:0]
0x0158								rdqsp_bdly_0[35:32]
0x0160	rdqsn_bdly_0[31:28]	rdqsn_bdly_0[27:24]	rdqsn_bdly_0[23:20]	rdqsn_bdly_0[19:16]	rdqsn_bdly_0[15:12]	rdqsn_bdly_0[11:8]	rdqsn_bdly_0[7:4]	rdqsn_bdly_0[3:0]
0x0168								rdqsn_bdly_0[35:32]
0x0170	rdq_bdly_0[24:21]	rdq_bdly_0[20:18]	rdq_bdly_0[17:15]	rdq_bdly_0[14:12]	rdq_bdly_0[11:9]	rdq_bdly_0[8:6]	rdq_bdly_0[5:3]	rdq_bdly_0[2:0]
0x0178								rdq_bdly_0[27:26]
0x0180					dll_1xdly_1	dll_1xgen_1	dll_wrdds_1	dll_wrddq_1
0x0188						dll_gate_1	dll_rdds1_1	dll_rdds0_1
0x0190	rdodt_ctrl_1	rdgate_len_1	rdgate_mode_1	rdgate_ctrl_1			dqs_oe_ctrl_1	dq_oe_ctrl_1
0x0198					dly_2x_1		redge_sel_1	rdds_phase_1(RD)
0x01a0	w_bdly0_1[31:28]	w_bdly0_1[27:24]	w_bdly0_1[23:20]	w_bdly0_1[19:16]	w_bdly0_1[15:12]	w_bdly0_1[11:8]	w_bdly0_1[7:4]	w_bdly0_1[3:0]
0x01a8		w_bdly0_1[59:56]	w_bdly0_1[55:52]	w_bdly0_1[51:48]	w_bdly0_1[47:44]	w_bdly0_1[43:40]	w_bdly0_1[39:36]	w_bdly0_1[35:32]
0x01b0	w_bdly1_1[24:21]	w_bdly1_1[20:18]	w_bdly1_1[17:15]	w_bdly1_1[14:12]	w_bdly1_1[11:9]	w_bdly1_1[8:6]	w_bdly1_1[5:3]	w_bdly1_1[2:0]
0x01b8								w_bdly1_1[27:26]
0x01c0							rg_bdly_1[7:4]	rg_bdly_1[3:0]
0x01c8								
0x01d0	rdqsp_bdly_1[31:28]	rdqsp_bdly_1[27:24]	rdqsp_bdly_1[23:20]	rdqsp_bdly_1[19:16]	rdqsp_bdly_1[15:12]	rdqsp_bdly_1[11:8]	rdqsp_bdly_1[7:4]	rdqsp_bdly_1[3:0]
0x01d8								rdqsp_bdly_1[35:32]
0x01e0	rdqsn_bdly_1[31:28]	rdqsn_bdly_1[27:24]	rdqsn_bdly_1[23:20]	rdqsn_bdly_1[19:16]	rdqsn_bdly_1[15:12]	rdqsn_bdly_1[11:8]	rdqsn_bdly_1[7:4]	rdqsn_bdly_1[3:0]
0x01e8								rdqsn_bdly_1[35:32]
0x01f0	rdq_bdly_1[24:21]	rdq_bdly_1[20:18]	rdq_bdly_1[17:15]	rdq_bdly_1[14:12]	rdq_bdly_1[11:9]	rdq_bdly_1[8:6]	rdq_bdly_1[5:3]	rdq_bdly_1[2:0]

0x01f8								rdq_bdly_1[27:26]
0x0200					dll_1xdly_2	dll_1xgen_2	dll_wrdqs_2	dll_wrdq_2
0x0208						dll_gate_2	dll_rddqs1_2	dll_rddqs0_2
0x0210	rdodt_ctrl_2	rdgate_len_2	rdgate_mode_2	rdgate_ctrl_2			dqs_oe_ctrl_2	dq_oe_ctrl_2
0x0218					dly_2x_2		redge_sel_2	rddqs_phase_2(RD)
0x0220	w_bdly0_2[31:28]	w_bdly0_2[27:24]	w_bdly0_2[23:20]	w_bdly0_2[19:16]	w_bdly0_2[15:12]	w_bdly0_2[11:8]	w_bdly0_2[7:4]	w_bdly0_2[3:0]
0x0228		w_bdly0_2[59:56]	w_bdly0_2[55:52]	w_bdly0_2[51:48]	w_bdly0_2[47:44]	w_bdly0_2[43:40]	w_bdly0_2[39:36]	w_bdly0_2[35:32]
0x0230	w_bdly1_2[24:21]	w_bdly1_2[20:18]	w_bdly1_2[17:15]	w_bdly1_2[14:12]	w_bdly1_2[11:9]	w_bdly1_2[8:6]	w_bdly1_2[5:3]	w_bdly1_2[2:0]
0x0238								w_bdly1_2[27:26]
0x0240							rg_bdly_2[7:4]	rg_bdly_2[3:0]
0x0248								
0x0250	rdqsp_bdly_2[31:28]	rdqsp_bdly_2[27:24]	rdqsp_bdly_2[23:20]	rdqsp_bdly_2[19:16]	rdqsp_bdly_2[15:12]	rdqsp_bdly_2[11:8]	rdqsp_bdly_2[7:4]	rdqsp_bdly_2[3:0]
0x0258								rdqsp_bdly_2[35:32]
0x0260	rdqsn_bdly_2[31:28]	rdqsn_bdly_2[27:24]	rdqsn_bdly_2[23:20]	rdqsn_bdly_2[19:16]	rdqsn_bdly_2[15:12]	rdqsn_bdly_2[11:8]	rdqsn_bdly_2[7:4]	rdqsn_bdly_2[3:0]
0x0268								rdqsn_bdly_2[35:32]
0x0270	rdq_bdly_2[24:21]	rdq_bdly_2[20:18]	rdq_bdly_2[17:15]	rdq_bdly_2[14:12]	rdq_bdly_2[11:9]	rdq_bdly_2[8:6]	rdq_bdly_2[5:3]	rdq_bdly_2[2:0]
0x0278								rdq_bdly_2[27:26]
0x0280					dll_1xdly_3	dll_1xgen_3	dll_wrdqs_3	dll_wrdq_3
0x0288						dll_gate_3	dll_rddqs1_3	dll_rddqs0_3
0x0290	rdodt_ctrl_3	rdgate_len_3	rdgate_mode_3	rdgate_ctrl_3			dqs_oe_ctrl_3	dq_oe_ctrl_3
0x0298					dly_2x_3		redge_sel_3	rddqs_phase_3(RD)
0x02a0	w_bdly0_3[31:28]	w_bdly0_3[27:24]	w_bdly0_3[23:20]	w_bdly0_3[19:16]	w_bdly0_3[15:12]	w_bdly0_3[11:8]	w_bdly0_3[7:4]	w_bdly0_3[3:0]
0x02a8		w_bdly0_3[59:56]	w_bdly0_3[55:52]	w_bdly0_3[51:48]	w_bdly0_3[47:44]	w_bdly0_3[43:40]	w_bdly0_3[39:36]	w_bdly0_3[35:32]
0x02b0	w_bdly1_3[24:21]	w_bdly1_3[20:18]	w_bdly1_3[17:15]	w_bdly1_3[14:12]	w_bdly1_3[11:9]	w_bdly1_3[8:6]	w_bdly1_3[5:3]	w_bdly1_3[2:0]
0x02b8								w_bdly1_3[27:26]
0x02c0							rg_bdly_3[7:4]	rg_bdly_3[3:0]
0x02c8								
0x02d0	rdqsp_bdly_3[31:28]	rdqsp_bdly_3[27:24]	rdqsp_bdly_3[23:20]	rdqsp_bdly_3[19:16]	rdqsp_bdly_3[15:12]	rdqsp_bdly_3[11:8]	rdqsp_bdly_3[7:4]	rdqsp_bdly_3[3:0]
0x02d8								rdqsp_bdly_3[35:32]
0x02e0	rdqsn_bdly_3[31:28]	rdqsn_bdly_3[27:24]	rdqsn_bdly_3[23:20]	rdqsn_bdly_3[19:16]	rdqsn_bdly_3[15:12]	rdqsn_bdly_3[11:8]	rdqsn_bdly_3[7:4]	rdqsn_bdly_3[3:0]
0x02e8								rdqsn_bdly_3[35:32]
0x02f0	rdq_bdly_3[24:21]	rdq_bdly_3[20:18]	rdq_bdly_3[17:15]	rdq_bdly_3[14:12]	rdq_bdly_3[11:9]	rdq_bdly_3[8:6]	rdq_bdly_3[5:3]	rdq_bdly_3[2:0]
0x02f8								rdq_bdly_3[27:26]
0x0300					dll_1xdly_4	dll_1xgen_4	dll_wrdqs_4	dll_wrdq_4
0x0308						dll_gate_4	dll_rddqs1_4	dll_rddqs0_4
0x0310	rdodt_ctrl_4	rdgate_len_4	rdgate_mode_4	rdgate_ctrl_4			dqs_oe_ctrl_4	dq_oe_ctrl_4
0x0318					dly_2x_4		redge_sel_4	rddqs_phase_4(RD)
0x0320	w_bdly0_4[31:28]	w_bdly0_4[27:24]	w_bdly0_4[23:20]	w_bdly0_4[19:16]	w_bdly0_4[15:12]	w_bdly0_4[11:8]	w_bdly0_4[7:4]	w_bdly0_4[3:0]
0x0328		w_bdly0_4[59:56]	w_bdly0_4[55:52]	w_bdly0_4[51:48]	w_bdly0_4[47:44]	w_bdly0_4[43:40]	w_bdly0_4[39:36]	w_bdly0_4[35:32]
0x0330	w_bdly1_4[24:21]	w_bdly1_4[20:18]	w_bdly1_4[17:15]	w_bdly1_4[14:12]	w_bdly1_4[11:9]	w_bdly1_4[8:6]	w_bdly1_4[5:3]	w_bdly1_4[2:0]
0x0338								w_bdly1_4[27:26]
0x0340							rg_bdly_4[7:4]	rg_bdly_4[3:0]
0x0348								
0x0350	rdqsp_bdly_4[31:28]	rdqsp_bdly_4[27:24]	rdqsp_bdly_4[23:20]	rdqsp_bdly_4[19:16]	rdqsp_bdly_4[15:12]	rdqsp_bdly_4[11:8]	rdqsp_bdly_4[7:4]	rdqsp_bdly_4[3:0]
0x0358								rdqsp_bdly_4[35:32]
0x0360	rdqsn_bdly_4[31:28]	rdqsn_bdly_4[27:24]	rdqsn_bdly_4[23:20]	rdqsn_bdly_4[19:16]	rdqsn_bdly_4[15:12]	rdqsn_bdly_4[11:8]	rdqsn_bdly_4[7:4]	rdqsn_bdly_4[3:0]
0x0368								rdqsn_bdly_4[35:32]

0x0370	rdq_bdly_4[24:21]	rdq_bdly_4[20:18]	rdq_bdly_4[17:15]	rdq_bdly_4[14:12]	rdq_bdly_4[11:9]	rdq_bdly_4[8:6]	rdq_bdly_4[5:3]	rdq_bdly_4[2:0]
0x0378								rdq_bdly_4[27:26]
0x0380					dll_1xdly_5	dll_1xgen_5	dll_wrdds_5	dll_wrdq_5
0x0388						dll_gate_5	dll_rdds1_5	dll_rdds0_5
0x0390	rdodt_ctrl_5	rdgate_len_5	rdgate_mode_5	rdgate_ctrl_5			dqs_oe_ctrl_5	dq_oe_ctrl_5
0x0398					dly_2x_5		redge_sel_5	rdds_phase_5(RD)
0x03a0	w_bdly0_5[31:28]	w_bdly0_5[27:24]	w_bdly0_5[23:20]	w_bdly0_5[19:16]	w_bdly0_5[15:12]	w_bdly0_5[11:8]	w_bdly0_5[7:4]	w_bdly0_5[3:0]
0x03a8		w_bdly0_5[59:56]	w_bdly0_5[55:52]	w_bdly0_5[51:48]	w_bdly0_5[47:44]	w_bdly0_5[43:40]	w_bdly0_5[39:36]	w_bdly0_5[35:32]
0x03b0	w_bdly1_5[24:21]	w_bdly1_5[20:18]	w_bdly1_5[17:15]	w_bdly1_5[14:12]	w_bdly1_5[11:9]	w_bdly1_5[8:6]	w_bdly1_5[5:3]	w_bdly1_5[2:0]
0x03b8								w_bdly1_5[27:26]
0x03c0							rg_bdly_5[7:4]	rg_bdly_5[3:0]
0x03c8								
0x03d0	rdqsp_bdly_5[31:28]	rdqsp_bdly_5[27:24]	rdqsp_bdly_5[23:20]	rdqsp_bdly_5[19:16]	rdqsp_bdly_5[15:12]	rdqsp_bdly_5[11:8]	rdqsp_bdly_5[7:4]	rdqsp_bdly_5[3:0]
0x03d8								rdqsp_bdly_5[35:32]
0x03e0	rdqsn_bdly_5[31:28]	rdqsn_bdly_5[27:24]	rdqsn_bdly_5[23:20]	rdqsn_bdly_5[19:16]	rdqsn_bdly_5[15:12]	rdqsn_bdly_5[11:8]	rdqsn_bdly_5[7:4]	rdqsn_bdly_5[3:0]
0x03e8								rdqsn_bdly_5[35:32]
0x03f0	rdq_bdly_5[24:21]	rdq_bdly_5[20:18]	rdq_bdly_5[17:15]	rdq_bdly_5[14:12]	rdq_bdly_5[11:9]	rdq_bdly_5[8:6]	rdq_bdly_5[5:3]	rdq_bdly_5[2:0]
0x03f8								rdq_bdly_5[27:26]
0x0400					dll_1xdly_6	dll_1xgen_6	dll_wrdds_6	dll_wrdq_6
0x0408						dll_gate_6	dll_rdds1_6	dll_rdds0_6
0x0410	rdodt_ctrl_6	rdgate_len_6	rdgate_mode_6	rdgate_ctrl_6			dqs_oe_ctrl_6	dq_oe_ctrl_6
0x0418					dly_2x_6		redge_sel_6	rdds_phase_6(RD)
0x0420	w_bdly0_6[31:28]	w_bdly0_6[27:24]	w_bdly0_6[23:20]	w_bdly0_6[19:16]	w_bdly0_6[15:12]	w_bdly0_6[11:8]	w_bdly0_6[7:4]	w_bdly0_6[3:0]
0x0428		w_bdly0_6[59:56]	w_bdly0_6[55:52]	w_bdly0_6[51:48]	w_bdly0_6[47:44]	w_bdly0_6[43:40]	w_bdly0_6[39:36]	w_bdly0_6[35:32]
0x0430	w_bdly1_6[24:21]	w_bdly1_6[20:18]	w_bdly1_6[17:15]	w_bdly1_6[14:12]	w_bdly1_6[11:9]	w_bdly1_6[8:6]	w_bdly1_6[5:3]	w_bdly1_6[2:0]
0x0438								w_bdly1_6[27:26]
0x0440							rg_bdly_6[7:4]	rg_bdly_6[3:0]
0x0448								
0x0450	rdqsp_bdly_6[31:28]	rdqsp_bdly_6[27:24]	rdqsp_bdly_6[23:20]	rdqsp_bdly_6[19:16]	rdqsp_bdly_6[15:12]	rdqsp_bdly_6[11:8]	rdqsp_bdly_6[7:4]	rdqsp_bdly_6[3:0]
0x0458								rdqsp_bdly_6[35:32]
0x0460	rdqsn_bdly_6[31:28]	rdqsn_bdly_6[27:24]	rdqsn_bdly_6[23:20]	rdqsn_bdly_6[19:16]	rdqsn_bdly_6[15:12]	rdqsn_bdly_6[11:8]	rdqsn_bdly_6[7:4]	rdqsn_bdly_6[3:0]
0x0468								rdqsn_bdly_6[35:32]
0x0470	rdq_bdly_6[24:21]	rdq_bdly_6[20:18]	rdq_bdly_6[17:15]	rdq_bdly_6[14:12]	rdq_bdly_6[11:9]	rdq_bdly_6[8:6]	rdq_bdly_6[5:3]	rdq_bdly_6[2:0]
0x0478								rdq_bdly_6[27:26]
0x0480					dll_1xdly_7	dll_1xgen_7	dll_wrdds_7	dll_wrdq_7
0x0488						dll_gate_7	dll_rdds1_7	dll_rdds0_7
0x0490	rdodt_ctrl_7	rdgate_len_7	rdgate_mode_7	rdgate_ctrl_7			dqs_oe_ctrl_7	dq_oe_ctrl_7
0x0498					dly_2x_7		redge_sel_7	rdds_phase_7(RD)
0x04a0	w_bdly0_7[31:28]	w_bdly0_7[27:24]	w_bdly0_7[23:20]	w_bdly0_7[19:16]	w_bdly0_7[15:12]	w_bdly0_7[11:8]	w_bdly0_7[7:4]	w_bdly0_7[3:0]
0x04a8		w_bdly0_7[59:56]	w_bdly0_7[55:52]	w_bdly0_7[51:48]	w_bdly0_7[47:44]	w_bdly0_7[43:40]	w_bdly0_7[39:36]	w_bdly0_7[35:32]
0x04b0	w_bdly1_7[24:21]	w_bdly1_7[20:18]	w_bdly1_7[17:15]	w_bdly1_7[14:12]	w_bdly1_7[11:9]	w_bdly1_7[8:6]	w_bdly1_7[5:3]	w_bdly1_7[2:0]
0x04b8								w_bdly1_7[27:26]
0x04c0							rg_bdly_7[7:4]	rg_bdly_7[3:0]
0x04c8								
0x04d0	rdqsp_bdly_7[31:28]	rdqsp_bdly_7[27:24]	rdqsp_bdly_7[23:20]	rdqsp_bdly_7[19:16]	rdqsp_bdly_7[15:12]	rdqsp_bdly_7[11:8]	rdqsp_bdly_7[7:4]	rdqsp_bdly_7[3:0]
0x04d8								rdqsp_bdly_7[35:32]
0x04e0	rdqsn_bdly_7[31:28]	rdqsn_bdly_7[27:24]	rdqsn_bdly_7[23:20]	rdqsn_bdly_7[19:16]	rdqsn_bdly_7[15:12]	rdqsn_bdly_7[11:8]	rdqsn_bdly_7[7:4]	rdqsn_bdly_7[3:0]

0x04e8								rdqsn_bdy_7[35:32]
0x04f0	rdq_bdy_7[24:21]	rdq_bdy_7[20:18]	rdq_bdy_7[17:15]	rdq_bdy_7[14:12]	rdq_bdy_7[11:9]	rdq_bdy_7[8:6]	rdq_bdy_7[5:3]	rdq_bdy_7[2:0]
0x04f8								rdq_bdy_7[27:26]
0x0500					dll_1xdly_8	dll_1xgen_8	dll_wrdqs_8	dll_wrdq_8
0x0508						dll_gate_8	dll_rddqs1_8	dll_rddqs0_8
0x0510	rdodt_ctrl_8	rdgate_len_8	rdgate_mode_8	rdgate_ctrl_8			dqs_oe_ctrl_8	dq_oe_ctrl_8
0x0518					dly_2x_8		redge_sel_8	rddqs_phase_8(RD)
0x0520	w_bdy0_8[31:28]	w_bdy0_8[27:24]	w_bdy0_8[23:20]	w_bdy0_8[19:16]	w_bdy0_8[15:12]	w_bdy0_8[11:8]	w_bdy0_8[7:4]	w_bdy0_8[3:0]
0x0528		w_bdy0_8[59:56]	w_bdy0_8[55:52]	w_bdy0_8[51:48]	w_bdy0_8[47:44]	w_bdy0_8[43:40]	w_bdy0_8[39:36]	w_bdy0_8[35:32]
0x0530	w_bdy1_8[24:21]	w_bdy1_8[20:18]	w_bdy1_8[17:15]	w_bdy1_8[14:12]	w_bdy1_8[11:9]	w_bdy1_8[8:6]	w_bdy1_8[5:3]	w_bdy1_8[2:0]
0x0538								w_bdy1_8[27:26]
0x0540							rg_bdy_8[7:4]	rg_bdy_8[3:0]
0x0548								
0x0550	rdqsp_bdy_8[31:28]	rdqsp_bdy_8[27:24]	rdqsp_bdy_8[23:20]	rdqsp_bdy_8[19:16]	rdqsp_bdy_8[15:12]	rdqsp_bdy_8[11:8]	rdqsp_bdy_8[7:4]	rdqsp_bdy_8[3:0]
0x0558								rdqsp_bdy_8[35:32]
0x0560	rdqsn_bdy_8[31:28]	rdqsn_bdy_8[27:24]	rdqsn_bdy_8[23:20]	rdqsn_bdy_8[19:16]	rdqsn_bdy_8[15:12]	rdqsn_bdy_8[11:8]	rdqsn_bdy_8[7:4]	rdqsn_bdy_8[3:0]
0x0568								rdqsn_bdy_8[35:32]
0x0570	rdq_bdy_8[24:21]	rdq_bdy_8[20:18]	rdq_bdy_8[17:15]	rdq_bdy_8[14:12]	rdq_bdy_8[11:9]	rdq_bdy_8[8:6]	rdq_bdy_8[5:3]	rdq_bdy_8[2:0]
0x0578								rdq_bdy_8[27:26]
.....								
0x0700					leveling_cs	tLVL_DELAY	leveling_req(WR)	leveling_mode
0x0708							leveling_done(RD)	leveling_ready(RD)
0x0710	leveling_resp_7	leveling_resp_6	leveling_resp_5	leveling_resp_4	leveling_resp_3	leveling_resp_2	leveling_resp_1	leveling_resp_0
0x0718								leveling_resp_8
0x0720								
.....								
0x0800	dfe_ctrl_ds	pad_ctrl_ds				pad_ctrl_ck		
0x0808		pad_reset_po	pad_oplen_ca	pad_opdly_ca		pad_ctrl_ca		
0x0810	vref_ctrl_ds_3		vref_ctrl_ds_2		vref_ctrl_ds_1		vref_ctrl_ds_0	
0x0818	vref_ctrl_ds_7		vref_ctrl_ds_6		vref_ctrl_ds_5		vref_ctrl_ds_4	
0x0820							vref_ctrl_ds_8	
0x0828								
0x0830			pad_comp_o(RD)				pad_comp_i	
0x0838								
0x0840	pad_ctrl_ds_3		pad_ctrl_ds_2		pad_ctrl_ds_1		pad_ctrl_ds_0	
0x0848	pad_ctrl_ds_7		pad_ctrl_ds_6		pad_ctrl_ds_5		pad_ctrl_ds_4	
0x0850							pad_ctrl_ds_8	
.....								
0x0900		rdedge_soft		rd_phase(RD)			clk_inv	
0x0908							rdedge_inv	
<b>CTL</b>								
0x1000		tRP	tWLDQSEN	tMOD	tXPR		tCKE	tRESET
0x1008								tODTL
0x1010	tREFretention				tRFC		tREF	
0x1018	tCKESR	tXSRD	tXS	tRFC_dlr			tREF_IDLE	
0x1020				tRDPDEN	tCPDED	tXPDLL	tXP	
0x1028				tZQperiod	tZQCL	tZQCS	tZQ_CMD	

.....								
0x1040	tRCD	tRRD_S_slr	tRRD_L_slr	tRRD_dlr				tRAS_min
0x1048				tRTP	tWR_CRC_DM	tWR	tFAW_slr	tFAW
0x1050	tWTR_S_CRC_DM	tWTR_L_CRC_DM	tWTR_S	tWTR		tCCD_dlr	tCCD_S_slr	tCCD_L_slr
0x1058								
0x1060			tPHY_WRLAT	tWL		tRDDATA	tPHY_RDLAT	tRL
0x1068				tCAL				tPL
0x1070			tW2P_sameba	tW2W_sameba	tW2R_sameba	tR2P_sameba	tR2W_sameba	tR2R_sameba
0x1078			tW2P_samebg	tW2W_samebg	tW2R_samebg	tR2P_samebg	tR2W_samebg	tR2R_samebg
0x1080			tW2P_samec	tW2W_samec	tW2R_samec	tR2P_samec	tR2W_samec	tR2R_samec
0x1088								
0x1090			tW2P_samecs	tW2W_samecs	tW2R_samecs	tR2P_samecs	tR2W_samecs	tR2R_samecs
0x1098				tW2W_diffcs	tW2R_diffcs		tR2W_diffcs	tR2R_diffcs
.....								
0x1100			cs_ref	cs_resync	cs_zqcl	cs_zq	cs_mrs	cs_enable
0x1108	cke_map				cs_map			
0x1110				cs2cid				cid_map
0x1118								
0x1120	mrs_done(RD)	mrs_req(WR)	pre_all_done(RD)	pre_all_req(WR)	cmd_cmd	status_cmd(RD)	cmd_req(WR)	command_mode
0x1128	cmd_cke	cmd_a			cmd_ba	cmd_bg	cmd_c	cmd_cs
0x1130								cmd_pda
0x1138						cmd_dq0		
0x1140	mr_3_cs_0		mr_2_cs_0		mr_1_cs_0		mr_0_cs_0	
0x1148	mr_3_cs_1		mr_2_cs_1		mr_1_cs_1		mr_0_cs_1	
0x1150	mr_3_cs_2		mr_2_cs_2		mr_1_cs_2		mr_0_cs_2	
0x1158	mr_3_cs_3		mr_2_cs_3		mr_1_cs_3		mr_0_cs_3	
0x1160	mr_3_cs_4		mr_2_cs_4		mr_1_cs_4		mr_0_cs_4	
0x1168	mr_3_cs_5		mr_2_cs_5		mr_1_cs_5		mr_0_cs_5	
0x1170	mr_3_cs_6		mr_2_cs_6		mr_1_cs_6		mr_0_cs_6	
0x1178	mr_3_cs_7		mr_2_cs_7		mr_1_cs_7		mr_0_cs_7	
0x1180	mr_3_cs_0_ddr4		mr_2_cs_0_ddr4		mr_1_cs_0_ddr4		mr_0_cs_0_ddr4	
0x1188			mr_6_cs_0_ddr4		mr_5_cs_0_ddr4		mr_4_cs_0_ddr4	
0x1190	mr_3_cs_1_ddr4		mr_2_cs_1_ddr4		mr_1_cs_1_ddr4		mr_0_cs_1_ddr4	
0x1198			mr_6_cs_1_ddr4		mr_5_cs_1_ddr4		mr_4_cs_1_ddr4	
0x11a0	mr_3_cs_2_ddr4		mr_2_cs_2_ddr4		mr_1_cs_2_ddr4		mr_0_cs_2_ddr4	
0x11a8			mr_6_cs_2_ddr4		mr_5_cs_2_ddr4		mr_4_cs_2_ddr4	
0x11b0	mr_3_cs_3_ddr4		mr_2_cs_3_ddr4		mr_1_cs_3_ddr4		mr_0_cs_3_ddr4	
0x11b8			mr_6_cs_3_ddr4		mr_5_cs_3_ddr4		mr_4_cs_3_ddr4	
0x11c0	mr_3_cs_4_ddr4		mr_2_cs_4_ddr4		mr_1_cs_4_ddr4		mr_0_cs_4_ddr4	
0x11c8			mr_6_cs_4_ddr4		mr_5_cs_4_ddr4		mr_4_cs_4_ddr4	
0x11d0	mr_3_cs_5_ddr4		mr_2_cs_5_ddr4		mr_1_cs_5_ddr4		mr_0_cs_5_ddr4	
0x11d8			mr_6_cs_5_ddr4		mr_5_cs_5_ddr4		mr_4_cs_5_ddr4	
0x11e0	mr_3_cs_6_ddr4		mr_2_cs_6_ddr4		mr_1_cs_6_ddr4		mr_0_cs_6_ddr4	
0x11e8			mr_6_cs_6_ddr4		mr_5_cs_6_ddr4		mr_4_cs_6_ddr4	
0x11f0	mr_3_cs_7_ddr4		mr_2_cs_7_ddr4		mr_1_cs_7_ddr4		mr_0_cs_7_ddr4	
0x11f8			mr_6_cs_7_ddr4		mr_5_cs_7_ddr4		mr_4_cs_7_ddr4	
0x1200			nc16_map	nc	channel_width	ba_xor_row_offset	addr_new	cs_place

0x1208						bg_xor_row_offset		addr_mirror
0x1210	addr_base_1			addr_base_0				
0x1218								
0x1220	addr_mask_1			addr_mask_0				
0x1228								
0x1230			cs_diff	c_diff	bg_diff	ba_diff	row_diff	col_diff
0x1238				CF_confbus_timeout				
0x1240	WRQthreshold	tRDQidle	wr_pkc_num	rwq_arb	retry	no_dead_inorder	placement_en	stb_en/pbuf
0x1248								tRWGNTidle
0x1250							rfifo_age	
0x1258	prior_age3		prior_age2		prior_age1		prior_age0	
0x1260	retry_cnt(RD)					rbuffer_max(RD)	rdfifo_depth	stat_en
0x1268								
.....								
0x1280	aw_512_align		rd_before_wr	ecc_enable		int_vector(RD)	int_trigger(RD)	int_enable
0x1288								
0x1290						int_cnt_fatal(RD)	int_cnt_err(RD)	int_cnt
0x1298	ecc_cnt_cs_7(RD)	ecc_cnt_cs_6(RD)	ecc_cnt_cs_5(RD)	ecc_cnt_cs_4(RD)	ecc_cnt_cs_3(RD)	ecc_cnt_cs_2(RD)	ecc_cnt_cs_1(RD)	ecc_cnt_cs_0(RD)
0x12a0	ecc_data_dir(RD)	ecc_code_dir(RD)	ecc_code_256(RD)					ecc_code_64(RD)
0x12a8	ecc_addr(RD)							
0x12b0	ecc_data[63:0](RD)							
0x12b8	ecc_data[127:64] (RD)							
0x12c0	ecc_data[191:128] (RD)							
0x12c8	ecc_data[255:192] (RD)							
.....								
0x1300							ref_num	ref_sch_en
0x1308							Status_sref(RD)	srefresh_req
.....								
0x1340	hardware_pd_7	hardware_pd_6	hardware_pd_5	hardware_pd_4	hardware_pd_3	hardware_pd_2	hardware_pd_1	hardware_pd_0
0x1348	power_sta_7(RD)	power_sta_6(RD)	power_sta_5(RD)	power_sta_4(RD)	power_sta_3(RD)	power_sta_2(RD)	power_sta_1(RD)	power_sta_0(RD)
0x1350	selfref_age		slowpd_age		fastpd_age		active_age	
0x1358				power_up				Age_step
0x1360	tCONF_IDLE				tLPMC_IDLE			
.....								
0x1380								zq_overlap
0x1388								zq_stat_en
0x1390	zq_cnt_1(RD)				zq_cnt_0(RD)			
0x1398	zq_cnt_3(RD)				zq_cnt_2(RD)			
0x13a0	zq_cnt_5(RD)				zq_cnt_4(RD)			
0x13a8	zq_cnt_6(RD)				zq_cnt_6(RD)			
.....								
0x13c0					odt_wr_cs_map			
0x13c8							odt_wr_length	odt_wr_delay
0x13d0					odt_rd_cs_map			
0x13d8							odt_rd_length	odt_rd_delay
.....								
0x1400				tRESYNC_length	tRESYNC_delay	tRESYNC_shift	tRESYNC_max	tRESYNC_min

.....								
0x1440					pre_predict		tm_cmdq_num	burst_length
0x1448								ca_timing
0x1450						wr/rd_dbi_en	ca_par_en	crc_en
0x1458							tCA_PAR	tWR_CRC
0x1460	bit_map_7	bit_map_6	bit_map_5	bit_map_6	bit_map_3	bit_map_2	bit_map_1	bit_map_0
0x1468	bit_map_15	bit_map_14	bit_map_13	bit_map_12	bit_map_11	bit_map_10	bit_map_9	bit_map_8
0x1470							bit_map_17	bit_map_16
0x1478								bitmap_mirror
0x1480				alertn_misc(RD)			alertn_cnt	alertn_clr
0x1488	alertn_addr(RD)							
.....								
0x1500	win0_base							
0x1508	win1_base							
0x1510	win2_base							
0x1518	win3_base							
0x1520	win4_base							
0x1528	win5_base							
0x1530	win6_base							
0x1538	win7_base							
.....								
0x1580	win0_mask							
0x1588	win1_mask							
0x1590	win2_mask							
0x1598	win3_mask							
0x15a0	win4_mask							
0x15a8	win5_mask							
0x15b0	win6_mask							
0x15b8	win7_mask							
.....								
0x1600	win0_mmap							
0x1608	win1_mmap							
0x1610	win2_mmap							
0x1618	win3_mmap							
0x1620	win4_mmap							
0x1628	win5_mmap							
0x1630	win6_mmap							
0x1638	win7_mmap							
.....								
0x1700							acc_hp	acc_en
0x1708	acc_fake_b				acc_fake_a			
0x1710								
0x1718								
0x1720	addr_base_acc_1				addr_base_acc_0			
0x1728								
0x1730	addr_mask_acc_1				addr_mask_acc_0			
0x1738								

MON							
0x2000							cmd_monitor
0x2008							
0x2010	cmd_fbck[63:0](RD)						
0x2018	cmd_fbck[127:64] (RD)						
0x2020					rw_switch_cnt(RD)		
.....							
0x2100							scheduler_mon
0x2108							
0x2110	sch_cmd_num(RD)						
0x2118	ba_conflict_all(RD)						
0x2120	ba_conflict_last1(RD)						
0x2128	ba_conflict_last2(RD)						
0x2130	ba_conflict_last3(RD)						
0x2138	ba_conflict_last4(RD)						
0x2140	ba_conflict_last5(RD)						
0x2148	ba_conflict_last6(RD)						
0x2150	ba_conflict_last7(RD)						
0x2158	ba_conflict_last8(RD)						
0x2160	rd_conflict(RD)						
0x2168	wr_conflict(RD)						
0x2170	rtw_conflict(RD)						
0x2178	wtr_conflict(RD)						
0x2180	rd_conflict_last1(RD)						
0x2188	wr_conflict_last1(RD)						
0x2190	rtw_conflict_last1(RD)						
0x2198	wtr_conflict_last1(RD)						
0x21a0	wr_rd_turnaround(RD)						
0x21a8	cs_turnaround(RD)						
0x21b0	bg_conflict(RD)						
.....							
0x2300					sm_leveling		sm_init
0x2308							
0x2310		sm_rank_03		sm_rank_02		sm_rank_01	sm_rank_00
0x2318		sm_rank_07		sm_rank_06		sm_rank_05	sm_rank_04
0x2320		sm_rank_11		sm_rank_10		sm_rank_09	sm_rank_08
0x2328		sm_rank_15		sm_rank_14		sm_rank_13	sm_rank_12
0x2330		sm_rank_19		sm_rank_18		sm_rank_17	sm_rank_16
0x2338		sm_rank_23		sm_rank_22		sm_rank_21	sm_rank_20
0x2340		sm_rank_27		sm_rank_26		sm_rank_25	sm_rank_24
0x2348		sm_rank_31		sm_rank_30		sm_rank_29	sm_rank_28
.....							
TST							
0x3000					lpbk_mode	lpbk_start	lpbk_en
0x3008	lpbk_correct(RD)			lpbk_counter(RD)			lpbk_error(RD)
0x3010	lpbk_data_en[63:0]						
0x3018							lpbk_data_en[71:64]



0x3020							lpbk_data_mask_en	
0x3028								
0x3030	Lpbk_dat_w0[63:0]							
0x3038	Lpbk_dat_w0[127:64]							
0x3040	Lpbk_dat_w1[63:0]							
0x3048	Lpbk_dat_w1[127:64]							
0x3050		lpbk_ecc_mask_w0	lpbk_dat_mask_w0				lpbk_ecc_w0	
0x3058		lpbk_ecc_mask_w1	lpbk_dat_mask_w1				lpbk_ecc_w1	
0x3060							prbs_23	
0x3068						prbs_init		
.....								
0x3100					fix_data_pattern_index	bus_width	page_size	test_engine_en
0x3108			cs_diff_tst	c_diff_tst	bg_diff_tst	ba_diff_tst	row_diff_tst	col_diff_tst
0x3120	addr_base_tst							
0x3128								
0x3130	user_data_pattern							
0x3138								
0x3140	valid_bits[63:0]							
0x3148							valid_bits[71:64]	
0x3150	ctrl[63:0]							
0x3158	ctrl[127:64]							
0x3160	obs[63:0] (RD)							
0x3168	obs[127:64] (RD)							
0x3170	obs[191:128] (RD)							
0x3178	obs[255:192] (RD)							
0x3180	obs[319:256] (RD)							
0x3188	obs[383:320] (RD)							
0x3190	obs[447:384] (RD)							
0x3198	obs[511:448] (RD)							
0x31a0	obs[575:512] (RD)							
0x31a8	obs[639:576] (RD)							
0x31b0					obs[671:640](RD)			
.....								
0x3200								
0x3208								
0x3220	tud_i0							
0x3228	tud_i1							
0x3230	tud_o(RD)							
.....								
0x3300	tst_300							
0x3308	tst_308							
0x3310	tst_310							
0x3318	tst_318							
0x3320	tst_320							
0x3328	tst_328							
0x3330	tst_330							
0x3338	tst_338							

0x3340	tst_340
0x3348	tst_348
0x3350	tst_350
0x3358	tst_358
0x3360	tst_360
0x3368	tst_368
0x3370	tst_370
0x3378	tst_378

## 20 显示控制器（D6:F1）

桥片的显示控制器支持的特性包括：

- 双路 HDMI 接口显示，其中 HDMI0 可复用为 VGA
- 每路显示最大支持至 1920 x 1080@60Hz
- Monochrome、ARGB8888 两种模式硬件光标
- 两路硬件光标，光标像素为 64 x 64 或 32 x 32 可选
- RGB444, RGB555, RGB565, RGB888 四种色深
- 输出抖动和伽马校正
- 可切换的双路线性帧缓冲
- 中断和软复位

### 20.1 DC 配置寄存器（D6:F1）

表 20-1 DC 控制器的 PCI 配置头

地址偏移	简称	描述	默认值	访问类型
00h-01h	VID	Vendor ID	0014h	RO
02h-03h	DID	Device ID	7A36h	RO
04h-05h	PCICMD	PCI Command	0000h	R/W, RO
08h	RID	Revision ID	02h	RO
09h	PI	Programming Interface	00h	RO
0Ah	SCC	Sub Class Code	00h	RO
0Bh	BCC	Base Class Code	03h	RO
0Ch	CLS	Cache Line Size	10h	RO
0Eh	HEADTYP	Header Type	00h	RO
10h-17h	CNL_BAR	Control Block Base Address Register	0000000000000004h	R/W,RO
2Ch-2Dh	SVID	Subsystem Vendor ID	0000h	RO
2Eh-2Fh	SID	Subsystem Identification	0000h	RO
3Ch	INT_LN	Interrupt Line	FFh	R/W
3Dh	INT_PN	Interrupt Pin	00h	RO

注：表中未列出的地址空间表示保留。

下面列出与 PCI 配置头规范稍有不同的寄存器及其描述。

#### PCICMD-PCI 命令寄存器（DC-D6:F1）

地址偏移：04-05h

属性：R/W, RO

默认值：0000h

大小：16 位

位域	名称	访问	描述
15:2	Reserved	R0	保留
1	Memory Space Enable	R/W	该位用来控制是否使能对 DC 控制寄存器的访问。 0: 禁止访问; 1: 使能对 DC 控制寄存器的访问。在将该位配置为 1 之前, 必须先配置 PCNL_BAR 寄存器。
0	Reserved	R0	保留

## 20.2 DC 控制寄存器

### 20.2.1 帧缓冲配置寄存器

寄存器名	偏移地址	R/W	描述	复位值
frameBufferConfig	0x1240 0x1250	R/W	帧缓冲配置寄存器 0x1240: DV00 0x1250: DV01	00000000h
frameBufferConfig	bit	描述		初始值
Reset	20	从值 1 变为 0 时软复位		0
DC_DMA	17:16	DMA 取数字字节数 0: 256 1: 128 2: 64 3: 32		0
Gamma	12	写 1 使能伽马校正		0
FB number	11	指示当前正在使用的缓冲区号, 只读		0
Switch Panel	9	写 1 表示使用另一路显示输出, 两路配合可实现两路显示交换、拷贝		0
Output Enable	8	写 1 使能显示输出		0
FB switch	7	写 1 使能缓冲区切换, 下一帧完成切换		0
Tile8 Enable	5	Tile8 格式使能 1: Tile8 格式 0: Tile4 格式		
TileModeEn	4	0: Linear 模式 1: tile 4x4 模式		0
Format	2:0	色深格式: 0: none 1: RGB444 2: RGB555 3: RGB565 4: ARGB8888 5: RGBA8888_REV		0

### 20.2.2 帧缓冲地址寄存器 0

寄存器名	偏移地址	R/W	描述	复位值
frameBufferAddr0	0x1260 0x1270	R/W	帧缓冲地址寄存器 0。 0x1260: DV00 0x1270: DV01	00000000h

frameBufferAddr0	bit	描述	初始值
Address	31:0	缓冲区 0 在内存中的物理地址	0

### 20.2.3 帧缓冲地址寄存器 1

寄存器名	偏移地址	R/W	描述	复位值
frameBufferAddr1	0x1580 0x1590	R/W	帧缓冲地址寄存器 1。 0x1580: DV00 0x1590: DV01	00000000h
frameBufferAddr1	bit	描述		初始值
Address	31:0	缓冲区 1 在内存中的物理地址		0

### 20.2.4 Meta0 低地址寄存器

寄存器名	偏移地址	R/W	描述	复位值
Meta0BufferAddrLow	0x1b00 0x1b10	R/W	Meta0 低地址寄存器 0x1b00: DV00 0x1b10: DV01	00000000h
Meta0BufferAddrLow	bit	描述		初始值
Address	31:0	Meta0 缓冲区在内存中的物理地址		0

### 20.2.5 Meta0 高地址寄存器

寄存器名	偏移地址	R/W	描述	复位值
Meta0BufferAddrHigh	0x1b20 0x1b30	R/W	Meta0 高地址寄存器 0x1b20: DV00 0x1b30: DV01	00000000h
Meta0BufferAddrHigh	bit	描述		初始值
Address	31:0	Meta0 缓冲区在内存中的物理地址高位		0

### 20.2.6 Meta1 低地址寄存器

寄存器名	偏移地址	R/W	描述	复位值
Meta1BufferAddrLow	0x1b40 0x1b50	R/W	Meta1 低地址寄存器 0x1b40: DV00 0x1b50: DV01	00000000h
Meta1BufferAddrLow	bit	描述		初始值
Address	31:0	Meta1 缓冲区在内存中的物理地址		0

## 20.2.7 Meta1 高地址寄存器

寄存器名	偏移地址	R/W	描述	复位值
MetalBufferAddrHigh	0x1b60 0x1b70	R/W	Meta1 高地址寄存器 0x1b60: DV00 0x1b70: DV01	00000000h
<b>MetalBufferAddrHigh</b>	<b>bit</b>	<b>描述</b>		<b>初始值</b>
Address	31:0	Meta1 缓冲区在内存中的物理地址高位		0

## 20.2.8 帧缓冲跨度寄存器

寄存器名	偏移地址	R/W	描述	复位值
frameBufferStride	0x1280 0x1290	R/W	帧缓冲跨度寄存器: 0x1280: DV00 0x1290: DV01	00000000h
<b>frameBufferStride</b>	<b>bit</b>	<b>描述</b>		<b>初始值</b>
Stride	31:0	显示屏一行的字节数。 需要注意的是根据像素计算出的字节数应按 256 字节向上取证。		0

## 20.2.9 帧缓冲初始字节寄存器

寄存器名	偏移地址	R/W	描述	复位值
frameBufferOrigin	0x1300 0x1310	R/W	帧缓冲初始字节寄存器 0x1300: DV00 0x1310: DV01	00000000h
<b>frameBufferOrigin</b>	<b>bit</b>	<b>描述</b>		<b>初始值</b>
Origin	31:0	显示屏左侧原有字节数，一般配 0 即可。		0

## 20.2.10 颜色抖动配置寄存器

寄存器名	偏移地址	R/W	描述	复位值
ditherConfig	0x1360 0x1370	R/W	颜色抖动配置寄存器 0x1360: DV00 0x1370: DV01	00000000h
<b>ditherConfig</b>	<b>bit</b>	<b>描述</b>		<b>初始值</b>
Enable	31	写 1 使能颜色抖动功能。		0
RedSize	19:16	红色域宽度		0
GreenSize	11:8	绿色域宽度		0
BlueSize	3:0	蓝色域宽度		0

## 20.2.11 颜色抖动查找表低位寄存器

寄存器名	偏移地址	R/W	描述	复位值
ditherTableLow	0x1380 0x1390	R/W	颜色抖动查找表低位寄存器 0x1380: DV00 0x1390: DV01	00000000h
ditherTableLow	bit	描述		初始值
Y1_X3	31:28	坐标 (3, 1) 处的比较值。		0
Y1_X2	27:24	坐标 (2, 1) 处的比较值。		0
Y1_X1	23:20	坐标 (1, 1) 处的比较值。		0
Y1_X0	19:16	坐标 (0, 1) 处的比较值。		0
Y0_X3	15:12	坐标 (3, 0) 处的比较值。		0
Y0_X2	11:8	坐标 (2, 0) 处的比较值。		0
Y0_X1	7:4	坐标 (1, 0) 处的比较值。		0
Y0_X0	3:0	坐标 (0, 0) 处的比较值。		0

### 20.2.12 颜色抖动查找表高位寄存器

寄存器名	偏移地址	R/W	描述	复位值
ditherTableHigh	0x13a0 0x13b0	R/W	颜色抖动查找表高位寄存器 0x13a0: DV00 0x13b0: DV01	00000000h
ditherTableLow	bit	描述		初始值
Y3_X3	31:28	坐标 (3, 3) 处的比较值。		0
Y3_X2	27:24	坐标 (2, 3) 处的比较值。		0
Y3_X1	23:20	坐标 (1, 3) 处的比较值。		0
Y3_X0	19:16	坐标 (0, 3) 处的比较值。		0
Y2_X3	15:12	坐标 (3, 2) 处的比较值。		0
Y2_X2	11:8	坐标 (2, 2) 处的比较值。		0
Y2_X1	7:4	坐标 (1, 2) 处的比较值。		0
Y2_X0	3:0	坐标 (0, 2) 处的比较值。		0

### 20.2.13 液晶面板配置寄存器

寄存器名	偏移地址	R/W	描述	复位值
panelConfig	0x13c0 0x13d0	R/W	液晶面板配置寄存器 0x13c0: DV00 0x13d0: DV01	101h
panelConfig	bit	描述		初始值
ClockPol	9	时钟极性, 写 1 取反		0
ClockEn	8	时钟使能, 写 1 使能		1
DEPol	1	数据使能极性, 写 1 取反		0
DE	0	数据使能, 写 1 使能		1

### 20.2.14 水平显示宽度寄存器

寄存器名	偏移地址	R/W	描述	复位值
HDisplay	0x1400 0x1410	R/W	行水平显示宽度寄存器 0x1400:DV00 0x1410:DV01	00000000h
HDisplay	Bit		描述	初始值
Total	28:16		显示屏一行的总像素数（包括非显示区）	0
Display	12:0		显示屏一行中显示区的像素数	0

### 20.2.15 行同步配置寄存器

寄存器名	偏移地址	R/W	描述	复位值
Hsync	0x1420 0x1430	R/W	行同步配置寄存器 0x1420:DV00 0x1430:DV01	40000000h
Hsync	Bit		描述	初始值
Pol	31		行同步极性，写 1 取反	0
Pulse	30		行同步使能，写 1 使能	1
End	28:16		行同步结束时的像素数	0
Start	12:0		行同步开始时的像素数	0

### 20.2.16 垂直显示高度寄存器

寄存器名	偏移地址	R/W	描述	复位值
VDisplay	0x1480 0x1490	R/W	垂直显示高度寄存器 0x1480:DV00 0x1490:DV01	00000000h
VDisplay	Bit		描述	初始值
Total	27:16		显示屏一列的总像素数（包括非显示区）	0
Display	11:0		显示屏一列中显示区的像素数	0

### 20.2.17 场同步配置寄存器

寄存器名	偏移地址	R/W	描述	复位值
VSync	0x14a0 0x14b0	R/W	场同步配置寄存器 0x14a0:DV00 0x14b0:DV01	40000000h
VSync	Bit		描述	初始值
Pol	31		场同步极性，写 1 取反	0
Pulse	30		场同步使能，写 1 使能	1
End	27:16		场同步结束时的像素数	0
Start	11:0		场同步开始时的像素数	0



## 20.2.18 行场同步偏移配置寄存器

寄存器名	偏移地址	R/W	描述	复位值
SyncDeviation	0x1b80 0x1b90	R/W	场同步配置寄存器 0x1b80:DV00 0x1b90:DV01	00000000h
SyncDeviation	Bit		描述	初始值
SyncDeviation En	31		行场同步偏移使能, 不使能则场同步信号按像素行起落	0
SyncDeviationNum	12:0		行场同步偏移值, 单位为像素点, 数值为行同步起始位置减场同步起始位置	0

## 20.2.19 当前显示位置寄存器

寄存器名	偏移地址	R/W	描述	复位值
DisplayCurrentLocation	0x14c0 0x14d0	R	当前显示位置寄存器 0x14c0:DV00 0x14d0:DV01	00000000h
DisplayCurrentLocation	Bit		描述	初始值
Current Location X	31:16		当前 X 的位置	0
Current Location Y	15:0		当前 Y 的位置	0

## 20.2.20 伽马校正目录寄存器

寄存器名	偏移地址	R/W	描述	复位值
GammaIndex	0x14e0 0x14f0	R/W	伽马校正目录寄存器 0x14e0: DV00 0x14f0: DV01	00000000h
GammaIndex	bit		描述	初始值
Index	7:0		表示从0-255颜色值之间的哪一项开始进行Gamma调整, 一般设0。只需配一次, 此后该值硬件会自增。	0

## 20.2.21 伽马校正寄存器

寄存器名	偏移地址	R/W	描述	复位值
GammaData	0x1500 0x1510	R/W	伽马校正寄存器 0x1500: DV00 0x1510: DV01	00000000h
GammaIndex	bit		描述	初始值
Red	23:16		Gamma调整的红色域, 将Gamma Index指示的值调整为当前域的值	0
Green	15:8		Gamma调整的绿色域, 将Gamma Index指示的值调整为当前域的值	0
Blue	7:0		Gamma调整的蓝色域, 将Gamma Index指示的值调整为当前域的值	0

Gamma调整模块包含三个查找表, 一个负责红色, 一个负责绿色, 一个负责蓝色。

查找表可以通过寄存器改写。查找表只可写。

考虑一个 Gamma 调整如下：（原色，调整色）。当设置 Gamma 颜色查找表时，我们应该按照颜色值大小顺序配置寄存器。

如果我们想要一个调整为（0，0）（1，2）（2，5）（3，6）……，先对寄存器 Gamma Index 配置 0，表示 gamma 调整从 0 开始，然后对 Gamma Data 寄存器配置 256 次完成整个配置过程：0，2，5，6……

### 20.2.22 光标 0 配置寄存器

寄存器名	偏移地址	R/W	描述	复位值
Cursor0Config	0x1520	R/W	光标配置寄存器	00000000h
Cursor0Config	Bit	描述		初始值
HotSpotX	21:16	光标作用点的横坐标(在光标 32*32 或者 64*64 图案中的横坐标)		0
HotSpotY	13:8	光标作用点的纵坐标(在光标 32*32 或者 64*64 图案中的横坐标)		0
Display	4	指示光标存在于哪个显示单元中，0 表示在 0 号显示单元中，1 表示在 1 号显示单元中		0
Mode	2	1: 64*64 光标模式 0: 32*32 光标模式		0
Format	1:0	0:disabled 1:masked 2:ARGB8888		0

### 20.2.23 光标 0 存储地址寄存器

寄存器名	偏移地址	R/W	描述	复位值
Cursor0Address	0x1530	R/W	光标存储地址寄存器	00000000h
Cursor0Address	bit	描述		初始值
Address	31:0	光标数据在内存中的物理地址低 32 位		0
寄存器名	偏移地址	R/W	描述	复位值
Cursor0Address	0x15e0	R/W	光标存储地址寄存器	00000000h
Cursor0Address	bit	描述		初始值
Address	31:0	光标数据在内存中的物理地址高 32 位		0

### 20.2.24 光标 0 显示位置寄存器

寄存器名	偏移地址	R/W	描述	复位值
Cursor0Location	0x1540	R/W	光标显示位置寄存器	00000000h
Cursor0Location	Bit	描述		初始值
Y	27:16	光标作用点在整个屏幕的纵坐标		0
X	11:0	光标作用点在整个屏幕的横坐标		0

### 20.2.25 单色光标 0 背景色寄存器

寄存器名	偏移地址	R/W	描述	复位值
Cursor0Background	0x1550	R/W	单色光标背景色寄存器	00000000h
Cursor0Background	bit	描述		初始值
Red	23:16	光标单色模式下背景色的红色域		0
Green	15:8	光标单色模式下背景色的绿色域		0
Blue	7:0	光标单色模式下背景色的蓝色域		0

### 20.2.26 单色光标 0 前景色寄存器

寄存器名	偏移地址	R/W	描述	复位值
Cursor0Foreground	0x1560	R/W	单色光标前景色寄存器	00000000h
Cursor0Foreground	bit	描述		初始值
Red	23:16	光标单色模式下前景色的红色域		0
Green	15:8	光标单色模式下前景色的绿色域		0
Blue	7:0	光标单色模式下前景色的蓝色域		0

### 20.2.27 光标 1 配置寄存器

寄存器名	偏移地址	R/W	描述	复位值
Cursor1Config	0x1670	R/W	光标 1 配置寄存器	00000000h
Cursor1Config	Bit	描述		初始值
HotSpotX	21:16	光标作用点的横坐标（在光标 32*32 或者 64*64 图案中的横坐标）		0
HotSpotY	13:8	光标作用点的纵坐标（在光标 32*32 或者 64*64 图案中的纵坐标）		0
Display	4	指示光标存在于哪个显示单元中，0 表示在 0 号显示单元中，1 表示在 1 号显示单元中		0
Mode	2	1: 64*64 光标模式 0: 32*32 光标模式		0
Format	1:0	0: disabled 1: masked 2: ARGB8888		0

### 20.2.28 光标 1 存储地址寄存器

寄存器名	偏移地址	R/W	描述	复位值
Cursor1Address	0x1680	R/W	光标 1 存储地址寄存器	00000000h
Cursor1Address	Bit	描述		初始值
Address	31:0	光标数据在内存中的物理地址低 32 位		0
寄存器名	偏移地址	R/W	描述	复位值

Cursor1Address	0x16e0	R/W	光标 1 存储地址寄存器	00000000h
<b>Cursor1Address</b>	<b>Bit</b>	<b>描述</b>		<b>初始值</b>
Address	31:0	光标数据在内存中的物理地址高 32 位		0

### 20.2.29 光标 1 显示位置寄存器

<b>寄存器名</b>	<b>偏移地址</b>	<b>R/W</b>	<b>描述</b>	<b>复位值</b>
Cursor1Location	0x1690	R/W	光标 1 显示位置寄存器	00000000h
<b>Cursor1Location</b>	<b>Bit</b>	<b>描述</b>		<b>初始值</b>
Y	27:16	光标作用点在整个屏幕的纵坐标		0
X	11:0	光标作用点在整个屏幕的横坐标		0

### 20.2.30 单色光标 1 背景色寄存器

<b>寄存器名</b>	<b>偏移地址</b>	<b>R/W</b>	<b>描述</b>	<b>复位值</b>
Cursor1Background	0x16a0	R/W	光标 1 单色光标背景色寄存器	00000000h
<b>Cursor1Background</b>	<b>Bit</b>	<b>描述</b>		<b>初始值</b>
Red	23:16	单色光标模式下背景色的红色域		0
Green	15:8	单色光标模式下背景色的绿色域		0
Blue	7:0	单色光标模式下背景色的蓝色域		0

### 20.2.31 单色光标 1 前景色寄存器

<b>寄存器名</b>	<b>偏移地址</b>	<b>R/W</b>	<b>描述</b>	<b>复位值</b>
Cursor1Foreground	0x16b0	R/W	光标 1 单色光标前景色寄存器	00000000h
<b>Cursor1Foreground</b>	<b>Bit</b>	<b>描述</b>		<b>初始值</b>
Red	23:16	单色光标模式下前景色的红色域		0
Green	15:8	单色光标模式下前景色的绿色域		0
Blue	7:0	单色光标模式下前景色的蓝色域		0

Display Controller 支持硬件光标。当开启硬件光标时，可以下面两种格式：

1. XOR cursor
2. Full color RGB

在 XOR 光标格式情况下，每个像素点使用 2 位。一位为 mask，一位为 XOR。Mask 位产生指针的形状。XOR 位决定该像素显示。

表 20-2 单色光标模式

AND(mask_bit)	XOR(xor_bit)	CURSOR_COLOR
0	0	背景色

0	1	前景色
1	0	透明
1	1	屏幕反色

在 RGB 指针格式下，每个像素包含 8 位 R，8 位 G 和 8 位 B。

Alpha 部分表示插值系数。

指针有两个重要的点：左上点（top-left point）和作用点（hot spot）。

左上点用来作为指针地址的参考点。

作用点用来将鼠标按下动作确定到一个像素上。

### 20.2.32 HDMI 区域配置寄存器

寄存器名	偏移地址	R/W	描述	复位值
HdmiZoneIdle	0x1700 0x1710	R/W	HDMI 区域配置寄存器 0x1700: HDMI0 0x1710: HDMI1	00000000h
HdmiZoneIdle	bit	描述		初始值
HzoneIdle	7:0	Hzone 中预留给可能存在的连续包的 idle 周期数，如果小于 48 则 hblank 区域禁止发包		0
Hwait	15:8	在 hblank 区域等待 Hwait 周期数再允许发包，一般配 0		0
VzoneIdle	23:16	Vzone 中预留给可能存在的连续包的 idle 周期数，如果小于 48 则可能出错		0
Vwait	31:24	在 vblank 区域等待 Vwait 周期数再允许发包，一般配 0		

### 20.2.33 HDMI 控制寄存器

寄存器名	偏移地址	R/W	描述	复位值
HdmiInterfaceCtrl	0x1720 0x1730	R/W	HDMI 控制寄存器 0x1720: HDMI0 0x1730: HDMI1	00000000h
HdmiInterfaceCtrl	bit	描述		初始值
InterfaceEnable	0	HDMI 使能		0
PacketEnable	1	包使能		0
AudioEnable	2	音频使能		0
VideoLGBDisable	3	视频 Leading Guard Band 禁止使能		0
VideoPreambleLength	7:4	Video Preamble 长度		4' d8
I2CPadSel	8	0: 使用 GPIO 通过软件模拟 I2C 功能 1: 使用内置 I2 模块		0
CtlPeriodMode	9	HBlank 和 Vblank 区域的 control 周期发送的值 0: CTL0 - CTL3 都为 0 1: CTL0 为 1, CTL1 - CTL3 为 0, 即等同于 Video Preamble		0

### 20.2.34 Audio BUF 配置寄存器

寄存器名	偏移地址	R/W	描述	复位值
AudioBuf	0x1740 0x1750	R/W	HDMI 模块缓存 HDA 发送的音频数据，推迟 ASP 包的发送，用于音视频同步 0x1740: HDMI0 0x1750: HDMI1	00000000h
AudioBuf	bit	描述		初始值
AudioBufEn	31	写 1 使能推迟 ASP 包的发送功能		0
AudioBufClr	30	写 1 清除 Audio Buf 中缓存的音频数据		0
AudioBufSize	9:0	设置缓存的音频数据的大小，单位为字节；仅低 8 位有效，最多缓存 256 字节音频数据； Base Freq 44.1KHz: 延迟时间= AudioBufSize[7:0]*(1/44.1K) ms Base Freq 48KHz: 延迟时间= AudioBufSize[7:0]*(1/48K) ms		10' h0

### 20.2.35 Audio N 配置寄存器

寄存器名	偏移地址	R/W	描述	复位值
audioNConfig	0x1760 0x1770	R/W	N 配置寄存器 0x1760: HDMI0 0x1770: HDMI1	00000000h
audioNConfig	bit	描述		初始值
NValue	19:0	配置 N 的值		20' h0

### 20.2.36 Audio CTS 配置寄存器

寄存器名	偏移地址	R/W	描述	复位值
audioCTSConfig	0x1780 0x1790	R/W	CTS 配置寄存器 0x1780: HDMI0 0x1790: HDMI1	00000000h
audioCTSConfig	bit	描述		初始值
CTSValue	19:0	配置静态 CTS 的值		20' h0
CTSCntStop	28	CTS 动态测量模式下，是否使能 CTS 动态测量 0: 使能 CTS 动态测量 1: 停止 CTS 动态测量，CTS 动态测量结果设置为 0		0
CTSFix	29	CTS 生成模式： 0: 动态测量 1: 静态配置		0
NCTSUpdate	30	静态配置模式下：写 1 更新 ACR 包中 N 和 CTS 值		0
ACREn	31	使能 ACR 包		0

### 20.2.37 Audio CTS Cal 配置寄存器

寄存器名	偏移地址	R/W	描述	复位值
audioCTSCalConfig	0x17a0 0x17b0	R	CTS Cal 配置寄存器 0x17a0: HDMI0 0x17b0: HDMI1	00000000h
<b>audioCTSCalConfig</b>	<b>bit</b>	<b>描述</b>		<b>初始值</b>
CTSCalValue	19:0	保存动态生成的 CTS 值		0

### 20.2.38 Audio InfoFrame 配置寄存器

寄存器名	偏移地址	R/W	描述	复位值
audioInfoFrameConfig	0x17c0 0x17d0	R/W	音频信息帧包配置寄存器 0x17c0: HDMI0 0x17d0: HDMI1	00000000h
<b>audioInfoFrameConfig</b>	<b>bit</b>	<b>描述</b>		<b>初始值</b>
Enable	0	写 1 使能发送 Audio InfoFrame Packet		0
Frequency	1	Audio InfoFrame Packet 发送频率： 0: 每 1 个视频域发送一次 1: 每 2 个视频域发送一次		0
Update	2	写 1 更新 Audio InfoFrame Packet 寄存器		0
ChannelCount	6:4	音频通道计数		0
ChannelAllocation	15:8	音频通道/播放器分配		0
DownMix	16	仅在 DVD 音频应用： 0: 禁止 down-mix 1: 允许 down-mix		0
LevelShift alue	23:20	水平移位值（仅在 DVD 音频应用）		0
LFEPBL	25:24	LFE 播放等级信息		0

### 20.2.39 Audio Sample 配置寄存器

寄存器名	偏移地址	R/W	描述	复位值
audioSampleConfig	0x17e0 0x17f0	R/W	音频数据包配置寄存器 0x17e0: HDMI0 0x17f0: HDMI1	00000000h
<b>audioSampleConfig</b>	<b>bit</b>	<b>描述</b>		<b>初始值</b>
ASPEn	0	写 1 使能发送 Audio Sample Packet 的功能		0
SampleLayout	1	配置 Audio Sample layout		0
ChannelStatusFiexd	2	配置音频数据报文的 Channel Status 域来源： 0: HDA Digital Converter 命令 1: DC 音频通道状态寄存器		0
LeftChnlValid	3	写 1 强制设置 ASP 中左声道数据有效		0
RightChnlValid	4	写 1 强制设置 ASP 中右声道数据有效		0

### 20.2.40 HDMI PHY 控制寄存器

寄存器名	偏移地址	R/W	描述	复位值
HDMIPhyCtrl	0x1800 0x1810	R/W	HDMI PHY 控制寄存器 0x1800: HDMI0 0x1810: HDMI1	00000000h
HDMIPhyCtrl	bit	描述		初始值
PhyEn	0	除 Sink 的终止检测之外的整个宏功能的使能位。对 IDDQ 模式, PhyEn = 'L' & PhyTermDetEn = 'L'。最小使能脉冲 (PhyEn = 'L') 带宽 $\geq 1\mu\text{s}$		0
PhyResetn	1	全局异步 reset, 低有效		0
PhyRextEn	2	高有效的偏压电阻检测。独立于 PhyEn		0
PhyBiasSel	3	0: 接 1.8V 的 240 欧姆 $\pm 1\%$ 的偏压电阻应当接在 "HDMIBIAS" 引脚上。 1: 45uA $\pm 2\%$ 的对地电流应当接在 "IHDMIbias50u" 引脚上		0
PhyBiasStatus	4	偏压电阻检测状态 PhyBiasStatus = 'H': 偏压电阻已接入。		0
PhyTermEn	9:8	高有效的 Source 的终止控制。 对 $\leq 1.65\text{Gbps}$ : PhyTermEn<1:0> = 'XL' 对 $> 1.65\text{Gbps}$ : PhyTermEn<1:0> = 'XH'		
PhyTermDetEn	10	高有效的 Sink 终止检测使能。与 Phy En 独立		
PhyTermStatus	11	接受者终止检测状态位。 PhyTermStatus = 'H' 表示存在 Sink 终止。		
PhySkewChk	13:12	2Bit 的 Inter Pair Skew 的检测控制位 00: 不检测 01: Inter Pair Skew $\geq 2 \cdot T_{BIT}$ 设错误标识位 10: Inter Pair Skew $\geq 4 \cdot T_{BIT}$ 设错误标识位 11: Inter Pair Skew $\geq 6 \cdot T_{BIT}$ 设错误标识位		
PhySkewErr	14	当 Inter Pair Skew 的值大于所设定值时的错误标识位		

### 20.2.41 HDMI PHY PLL 配置寄存器

寄存器名	偏移地址	R/W	描述	复位值
HDMIPhyPLLConfig	0x1820 0x1830	R/W	HDMI PHY 控制寄存器 0x1820: HDMI0 0x1830: HDMI1	00000000h
HDMIPhyPLLConfig	bit	描述		初始值
HDMIPLL	15:0	0:0 : PLL_ENABLE, 高电平有效的使能信号, 最小使能脉冲 (PLL_ENABLE = 'L') 带宽 $\geq 1\mu\text{s}$ 5:1 : PLL_IDF<4:0>, 输入分频系数 12:6 : PLL_NDIV<6:0>, 循环分频系数 15:13: PLL_ODF<2:0>, 输出分频系数		0
HDMIPLLLocked	16	PHY PLL 锁定指示		0
HDMIPLLBypass	17	HDMI PHY 内 PLL bypass 设置, 高有效, 仅用于调试		0

### 20.2.42 HDMI PHY PEC0 寄存器

寄存器名	偏移地址	R/W	描述	复位值
HDMIPhyPEC0	0x1840 0x1850	R/W	0x1840: HDMI0 0x1850: HDMI1	00000000h



HDMI_PHY_PECO	bit	描述	初始值
TST_PE1C0_ENABLE	0	Tap 号为 1, Channel 号为 0 的预加重控制使能	0
TST_PE2C0_ENABLE	1	Tap 号为 2, Channel 号为 0 的预加重控制使能	0
TST_PE3C0_ENABLE	2	Tap 号为 3, Channel 号为 0 的预加重控制使能	0
TST_PE1C0_STR	7:4	Tap 号为 1, Channel 号为 0 的系数值	0
TST_PE2C0_STR	11:8	Tap 号为 2, Channel 号为 0 的系数值	0
TST_PE3C0_STR	15:12	Tap 号为 3, Channel 号为 0 的系数值	0
TST_PECO_FLOAT_SEL	19:16	Channel 号为 0 的内部扭斜对调整的可编程边沿控制	0
TST_PEBYPASS_ENABLE_N	31	低电平有效的预加重旁路电流使能	0

### 20.2.43 HDMI PHY PEC1 寄存器

寄存器名	偏移地址	R/W	描述	复位值
HDMI_PHY_PEC1	0x1860 0x1870	R/W	0x1860: HDMI0 0x1870: HDMI1	00000000h
HDMI_PHY_PEC1	bit		描述	初始值
TST_PE1C1_ENABLE	0		Tap 号为 1, Channel 号为 1 的预加重控制使能	0
TST_PE2C1_ENABLE	1		Tap 号为 2, Channel 号为 1 的预加重控制使能	0
TST_PE3C1_ENABLE	2		Tap 号为 3, Channel 号为 1 的预加重控制使能	0
TST_PE1C1_STR	7:4		Tap 号为 1, Channel 号为 1 的系数值	0
TST_PE2C1_STR	11:8		Tap 号为 2, Channel 号为 1 的系数值	0
TST_PE3C1_STR	15:12		Tap 号为 3, Channel 号为 1 的系数值	0
TST_PEC1_FLOAT_SEL	19:16		Channel 号为 1 的内部扭斜对调整的可编程边沿控制	0

### 20.2.44 HDMI PHY PEC2 寄存器

寄存器名	偏移地址	R/W	描述	复位值
HDMI_PHY_PEC2	0x1880 0x1890	R/W	0x1880: HDMI0 0x1890: HDMI1	00000000h
HDMI_PHY_PEC2	bit		描述	初始值
TST_PE1C2_ENABLE	0		Tap 号为 1, Channel 号为 2 的预加重控制使能	0
TST_PE2C2_ENABLE	1		Tap 号为 2, Channel 号为 2 的预加重控制使能	0
TST_PE3C2_ENABLE	2		Tap 号为 3, Channel 号为 2 的预加重控制使能	0
TST_PE1C2_STR	7:4		Tap 号为 1, Channel 号为 2 的系数值	0
TST_PE2C2_STR	11:8		Tap 号为 2, Channel 号为 2 的系数值	0
TST_PE3C2_STR	15:12		Tap 号为 3, Channel 号为 2 的系数值	0
TST_PEC2_FLOAT_SEL	19:16		Channel 号为 2 的内部扭斜对调整的可编程边沿控制	0

### 20.2.45 AVI InfoFrame 内容寄存器 0

寄存器名	偏移地址	R/W	描述	复位值
AVI Content0	0x18e0 0x18f0	R/W	AVI InfoFrame 内容配置寄存器 0 0x18e0:HDMI0 0x18f0:HDMI1	00000800h
AVI Content0	Bit	描述		初始值
Scan Information	1:0	相关条目详细定义见 HDMI 1.4b, 8.2.1. Auxiliary Video information(AVI) InfoFrame		0
Bar Info	3:2			0
Active Format Information Present	4			0
RGB or YCbCr	6:5			0
Active Format Aspect Ratio	11:8			4' b1000
Picture Aspect Ratio	13:12			0
Colorimetry	15:14			0
NonUniform Picture Scaling	17:16			0
RGB Quantization Range	19:18			0
Extended Colorimetry	22:20			0
IT content	23			0
Video Format Identification Code	30:24	0		

### 20.2.46 AVI InfoFrame 内容寄存器 1

寄存器名	偏移地址	R/W	描述	复位值
AVI Content1	0x1900 0x1910	R/W	AVI InfoFrame 内容配置寄存器 1 0x1900:HDMI0 0x1910:HDMI1	00000000h
AVI Content1	Bit	描述		初始值
Pixel Repetition	3:0	相关条目详细定义见 HDMI 1.4b, 8.2.1. Auxiliary Video information(AVI) InfoFrame		0
Content Type	5:4			0
YCC Quantization Range	7:6			0

### 20.2.47 AVI InfoFrame 内容寄存器 2

寄存器名	偏移地址	R/W	描述	复位值
AVI Content2	0x1920 0x1930	R/W	AVI InfoFrame 内容配置寄存器 2 0x1920:HDMI0 0x1930:HDMI1	00000000h
AVI Content2	Bit	描述		初始值
Line Number of End of Top Bar(lower 8 bits)	7:0	相关条目详细定义见 HDMI 1.4b, 8.2.1. Auxiliary Video information(AVI) InfoFrame		0
Line Number of End of Top Bar(upper 8 bits)	15:8			0
Line Number of Start of Bottom Bar(lower 8 bits)	23:16			0

Line Number of Start of Bottom Bar (upper 8 bits)	31:24		0
---	-------	--	---

### 20.2.48 AVI InfoFrame 内容寄存器 3

寄存器名	偏移地址	R/W	描述	复位值
AVI Content3	0x1940 0x1950	R/W	AVI InfoFrame 内容配置寄存器 3 0x1940:HDMI0 0x1950:HDMI1	00000000h
AVI Content3	Bit	描述		初始值
Pixel Number of End of Left Bar (lower 8 bits)	7:0	相关条目详细定义见 HDMI 1.4b, 8.2.1. Auxiliary Video information (AVI) InfoFrame		0
Pixel Number of End of Left Bar (upper 8 bits)	15:8			0
Pixel Number of Start of Right Bar (lower 8 bits)	23:16			0
Pixel Number of Start of Right Bar (upper 8 bits)	31:24			0

### 20.2.49 AVI InfoFrame 控制寄存器

寄存器名	偏移地址	R/W	描述	复位值
AVI Control	0x1960 0x1970	R/W	AVI InfoFrame 控制寄存器 0x1960:HDMI0 0x1970:HDMI1	00000000h
AVI Control	Bit	描述		初始值
AVI Enable	0	AVI InfoFrame Packet 使能		0
AVI Frequency	1	AVI InfoFrame Packet 发送频率 1: 每两帧一个 0: 每一帧一个		0
AVI Update	2	写 1 更新 AVI Content0 - AVI Content3 的内容至将要发送的包内, 硬件自动清 0		0

### 20.2.50 Vendor Specific InfoFrame 配置寄存器

寄存器名	偏移地址	R/W	描述	复位值
Vendor Specific InfoFrame Config	0x1980 0x1990	R/W	HDMI Vendor Specific InfoFrame 配置寄存器 0x1980:HDMI0 0x1990:HDMI1	00000000h
Vendor Specific InfoFrame Config	Bit	描述		初始值
VSI Enable	0	Vendor Specific InfoFrame Packet 使能		0
VSI Frequency	1	Vendor Specific InfoFrame Packet 发送频率 1: 每两帧一个 0: 每一帧一个		0
VSI Update	2	写 1 更新该寄存器后续的内容至将要发送的包内, 硬件自动清 0		0
HDMI Video Format	6:4			0

HDMI Video Format Identification Code	15:8	相关条目详细定义见 HDMI 1.4b, 8.2.3. HDMI Vendor Specific InfoFrame	0
3D Video Format Structure	19:16		0
3D Ext Data	23:20		0

### 20.2.51 场同步计数寄存器

寄存器名	偏移地址	R/W	描述	复位值
Vsync Counter	0x1a00 0x1a10	R	Vsync 计数寄存器 0x1a00:DV00 0x1a10:DV01	00000000h
Vsync Counter	Bit	描述		初始值
Vsync Counter	31: 0	Vsync 计数		0

### 20.2.52 PLL\_PIX 配置寄存器 0

寄存器名	偏移地址	R/W	描述	复位值
audioPIXPLLConfig0	0x1a20 0x1a30	R/W	像素 PLL 配置寄存器 0 0x1a20: HDMI0 0x1a30: HDMI1	00000000h
audioPIXPLLConfig0	bit	描述		初始值
PLLDivOut	6:0	PLL 输出时钟 0 分频数		0
PLLDivLoop	29:21	PLL 倍频乘数		0

### 20.2.53 PLL\_PIX 配置寄存器 1

寄存器名	偏移地址	R/W	描述	复位值
audioPIXPLLConfig1	0x1a40 0x1a50	R/W	像素 PLL 配置寄存器 1 0x1a40: HDMI0 0x1a50: HDMI1	00000000h
audioPIXPLLConfig1	bit	描述		初始值
PLLDivRef	6:0	PLL 输入分频数		0
PLLLocked	7	PLL 锁定		0
PLLOut	8	选择 PLL 输出时钟 0		0
PLLSoftSet	11	使能软件配置 PLL		0
PLLByPass	12	PLL 内部 bypass		0
PLLPd	13	PLL powerdown		0
PLLConfigSelect	14	PLL 配置源选择: 1: DC Pixel PLL 配置寄存器 0: PLL3/4 配置寄存器		0

### 20.2.54 VGA 热插拔控制寄存器

寄存器名	偏移地址	RW	描述	复位值
VGA_HPCR	0x1bb0	RW	VGA 热插拔控制寄存器	00000000h
<b>VGA HotPlug Control Reg</b>	<b>bit</b>		<b>描述</b>	<b>初始值</b>
VGA_set_value	15:8		热插拔检测时 B 通道像素值设置	0
Cfg_vdac_pd	4		VDAC 模块低功耗使能	0
Ref_sel	3:2		比较器参考电平选择 00:250mV 01:300mV 10:350mV 11:400mV	0
VGA_off_det_en	1		使能拔出检测	0
VGA_on_det_en	0		使能插入监测	0

### 20.2.55 HDMI 热插拔状态寄存器

寄存器名	偏移地址	RO	描述	复位值
HDMI_HPSR	0x1ba0	RO	HDMI 热插拔状态寄存器	00000000h
<b>HDMI HotPlugStatusReg</b>	<b>bit</b>		<b>描述</b>	<b>初始值</b>
HDMI1 HotPlug Status	1		HDMI1 热插拔状态 1: 有插入 0: 无插入	0
HDMI0 HotPlug Status	0		HDMI0 热插拔状态 1: 有插入 0: 无插入	0

### 20.2.56 HDMI 左声道状态寄存器

寄存器名	偏移地址	R/W	描述	复位值
LeftChannelStatus0	0x2000 0x2010	R/W	左声道状态寄存器 0 0x2000: HDMI0 0x2010: HDMI1	00000000h
LeftChannelStatus1	0x2020 0x2030	R/W	左声道状态寄存器 1 0x2020: HDMI0 0x2030: HDMI1	00000000h
LeftChannelStatus2	0x2040 0x2050	R/W	左声道状态寄存器 2 0x2040: HDMI0 0x2050: HDMI1	00000000h
LeftChannelStatus3	0x2060 0x2070	R/W	左声道状态寄存器 3 0x2060: HDMI0 0x2070: HDMI1	00000000h
LeftChannelStatus4	0x2080 0x2090	R/W	左声道状态寄存器 4 0x2080: HDMI0 0x2090: HDMI1	00000000h
LeftChannelStatus5	0x20a0 0x20b0	R/W	左声道状态寄存器 5 0x20a0: HDMI0 0x20b0: HDMI1	00000000h

### 20.2.57 HDMI 左声道用户数据寄存器

寄存器名	偏移地址	R/W	描述	复位值
LeftUserData0	0x20c0 0x20d0	R/W	左声道用户数据寄存器 0 0x20c0: HDMI0 0x20d0: HDMI1	00000000h
LeftUserData1	0x20e0 0x20f0	R/W	左声道用户数据寄存器 1 0x20e0: HDMI0 0x20f0: HDMI1	00000000h
LeftUserData2	0x2100 0x2110	R/W	左声道用户数据寄存器 2 0x2100: HDMI0 0x2110: HDMI1	00000000h
LeftUserData3	0x2120 0x2130	R/W	左声道用户数据寄存器 3 0x2120: HDMI0 0x2130: HDMI1	00000000h
LeftUserData4	0x2140 0x2150	R/W	左声道用户数据寄存器 4 0x2140: HDMI0 0x2150: HDMI1	00000000h
LeftUserData5	0x2160 0x2170	R/W	左声道用户数据寄存器 5 0x2160: HDMI0 0x2170: HDMI1	00000000h

### 20.2.58 HDMI 右声道状态寄存器

寄存器名	偏移地址	R/W	描述	复位值
RightChannelStatus0	0x2180 0x2190	R/W	右声道状态寄存器 0 0x2000: HDMI0 0x2010: HDMI1	00000000h
RightChannelStatus1	0x21a0 0x21b0	R/W	右声道状态寄存器 1 0x2020: HDMI0 0x2030: HDMI1	00000000h
RightChannelStatus2	0x21c0 0x21d0	R/W	右声道状态寄存器 2 0x2040: HDMI0 0x2050: HDMI1	00000000h
RightChannelStatus3	0x21e0 0x21f0	R/W	右声道状态寄存器 3 0x2060: HDMI0 0x2070: HDMI1	00000000h
RightChannelStatus4	0x2200 0x2210	R/W	右声道状态寄存器 4 0x2080: HDMI0 0x2090: HDMI1	00000000h
RightChannelStatus5	0x2220 0x2230	R/W	右声道状态寄存器 5 0x20a0: HDMI0 0x20b0: HDMI1	00000000h

### 20.2.59 HDMI 右声道用户数据寄存器

寄存器名	偏移地址	R/W	描述	复位值
RightUserData0	0x2240 0x2250	R/W	右声道用户数据寄存器 0 0x2240: HDMI0 0x2250: HDMI1	00000000h
RightUserData1	0x2260 0x2270	R/W	右声道用户数据寄存器 1 0x2260: HDMI0 0x2270: HDMI1	00000000h

RightUserData2	0x2280 0x2290	R/W	右声道用户数据寄存器 2 0x2280: HDMI0 0x2290: HDMI1	00000000h
RightUserData3	0x22a0 0x22b0	R/W	右声道用户数据寄存器 3 0x22a0: HDMI0 0x22b0: HDMI1	00000000h
RightUserData4	0x22c0 0x22d0	R/W	右声道用户数据寄存器 4 0x22c0: HDMI0 0x22d0: HDMI1	00000000h
RightUserData5	0x22e0 0x22f0	R/W	右声道用户数据寄存器 5 0x22e0: HDMI0 0x22f0: HDMI1	00000000h

## 20.2.60 中断寄存器

寄存器名	偏移地址	R/W	描述	复位值
Interrupt	0x1570	R/W	中断寄存器	00000000h
Interrupt	bit	描述		初始值
Display1_Vsync	0	1号显示单元产生了Vsync; 写1清0		0
Display1_Hsync	1	1号显示单元产生了Hsync; 写1清0		0
Display0_Vsync	2	0号显示单元产生了Vsync; 写1清0		0
Display0_Hsync	3	0号显示单元产生了Hsync; 写1清0		0
CursorReadEnd	4	光标数据读取结束; 写1清0		0
FBIReadEnd	5	1号显示单元帧缓冲读取结束; 写1清0		0
FBOReadEnd	6	0号显示单元帧缓冲读取结束; 写1清0		0
DB1Underflow	7	1号显示单元数据缓冲区下溢; 写1清0		0
DB0Underflow	8	0号显示单元数据缓冲区下溢; 写1清0		0
DB1FUnderflow	9	1号显示单元数据缓冲区致命下溢; 写1清0		0
DB0FUnderflow	10	0号显示单元数据缓冲区致命下溢; 写1清0		0
i2c_int	12:11	I2C 中断; 写1清0		0
HDMI0_hotplug	13	HDMI0 热插拔; 写1清0		0
HDMI1_hotplug	14	HDMI1 热插拔; 写1清0		0
VGA HotPlug	15	VGA 热插拔; 写1清0		0
En0	16	第0位中断使能; 写1使能, 写0屏蔽		0
En1	17	第1位中断使能; 写1使能, 写0屏蔽		0
En2	18	第2位中断使能; 写1使能, 写0屏蔽		0
En3	19	第3位中断使能; 写1使能, 写0屏蔽		0
En4	20	第4位中断使能; 写1使能, 写0屏蔽		0
En5	21	第5位中断使能; 写1使能, 写0屏蔽		0
En6	22	第6位中断使能; 写1使能, 写0屏蔽		0
En7	23	第7位中断使能; 写1使能, 写0屏蔽		0
En8	24	第8位中断使能; 写1使能, 写0屏蔽		0
En9	25	第9位中断使能; 写1使能, 写0屏蔽		0

En10	26	第 10 位中断使能；写 1 使能，写 0 屏蔽	0
En11	27	第 11 位中断使能；写 1 使能，写 0 屏蔽	0
En12	28	第 12 位中断使能；写 1 使能，写 0 屏蔽	0
En13	29	第 13 位中断使能；写 1 使能，写 0 屏蔽	0
En14	30	第 14 位中断使能；写 1 使能，写 0 屏蔽	0
En15	31	第 15 位中断使能；写 1 使能，写 0 屏蔽	0



## 21 HDA 控制器 (D7:F0)

HDA 控制器兼容 High Definition Audio Specification Revision 1.0a。

HDA 相关的引脚设置寄存器见 4.6 节。

### 21.1 HDA 配置寄存器 (D7:F0)

表 21-1 HDA 控制器的 PCI 配置头

地址偏移	简称	描述	默认值	访问类型
00h-01h	VID	Vendor ID	0014h	RO
02h-03h	DID	Device ID	7A07h	RO
04h-05h	PCICMD	PCI Command	0000h	R/W, RO
08h	RID	Revision ID	00h	RO
09h	PI	Programming Interface	00h	RO
0Ah	SCC	Sub Class Code	03h	RO
0Bh	BCC	Base Class Code	04h	RO
0Ch	CLS	Cache Line Size	10h	RO
0Eh	HEADTY P	Header Type	00h	RO
10h-17h	CNL_BAR	Control Block Base Address Register	0000000000000004h	R/W,RO
2Ch-2Dh	SVID	Subsystem Vendor ID	0000h	RO
2Eh-2Fh	SID	Subsystem Identification	0000h	RO
3Ch	INT_LN	Interrupt Line	FFh	R/W
3Dh	INT_PN	Interrupt Pin	00h	RO

注:

1. HDA 配置头只在相关引脚配置为 HDA 模式下可见。
2. 表中未列出的地址空间表示保留。

下面列出与 PCI 配置头规范稍有不同的寄存器及其描述。

#### PCICMD-PCI 命令寄存器 (HDA-D7:F0)

地址偏移: 04-05h

属性: R/W, RO

默认值: 0000h

大小: 16 位

位域	名称	访问	描述
15:2	Reserved	RO	保留
1	Memory Space Enable	R/W	该位用来控制是否使能对 HDA 控制寄存器的访问。 0: 禁止访问; 1: 使能对 HDA 控制寄存器的访问。在将该位配置为 1 之前, 必须先配置 PCNL_BAR 寄存器。
0	Reserved	RO	保留

## 21.2HDA 控制寄存器描述

HDA 控制寄存器的设计完全按照 HD audio Rev 1.0 规范进行设计的，下表列举了主要的寄存器的参数信息，具体的参考 HD audio Rev 1.0 手册。

## 22 I2S 控制器 (D7:F0)

### 22.1 I2S 配置寄存器 (D7:F0)

表 22-1 I2S 控制器的 PCI 配置头

地址偏移	简称	描述	默认值	访问类型
00h-01h	VID	Vendor ID	0014h	RO
02h-03h	DID	Device ID	7A27h	RO
04h-05h	PCICMD	PCI Command	0000h	R/W, RO
08h	RID	Revision ID	00h	RO
09h	PI	Programming Interface	00h	RO
0Ah	SCC	Sub Class Code	01h	RO
0Bh	BCC	Base Class Code	04h	RO
0Ch	CLS	Cache Line Size	10h	RO
0Eh	HEADTYP	Header Type	00h	RO
10h-17h	CNL_BAR	Control Block Base Address Register	0000000000000004h	R/W, RO
2Ch-2Dh	SVID	Subsystem Vendor ID	0000h	RO
2Eh-2Fh	SID	Subsystem Identification	0000h	RO
3Ch	INT_LN	Interrupt Line	FFh	R/W
3Dh	INT_PN	Interrupt Pin	00h	RO

注:

1. I2S 配置头只在 hda0\_i2s\_en 配置为 0 时可见。
2. 表中未列出的地址空间表示保留。

下面列出与 PCI 配置头规范稍有不同的寄存器及其描述。

#### PCICMD-PCI 命令寄存器 (I2S-D7:F0)

地址偏移: 04-05h

属性: R/W, RO

默认值: 0000h

大小: 16 位

位域	名称	访问	描述
15:2	Reserved	RO	保留
1	Memory Space Enable	R/W	该位用来控制是否使能对 I2S 控制寄存器的访问。 0: 禁止访问; 1: 使能对 I2S 控制寄存器的访问。在将该位配置为 1 之前, 必须先配置 PCNL_BAR 寄存器。
0	Reserved	RO	保留

## 22.2 I2S 控制器寄存器

I2S 包含 5 个寄存器，定义如下表所示。

表 22-2 寄存器定义

寄存器名称	偏移地址	读/写(R/W)	功能描述	复位值
IISVersion	0x0000	R/W	I2S 标识寄存器	32'h0
IISConfig	0x0004	R/W	I2S 配置寄存器	32'h0
IISControl	0x0008	R/W	I2S 控制寄存器	32'h0
IISRxData	0x000c	R/W	I2S 接收数据寄存器（用于 DMA 接收数据）	32'h0
IISTxData	0x0010	R/W	I2S 发送数据寄存器（用于 DMA 发送数据）	32'h0

I2S 标识寄存器允许主控机读取接收器的相关工作信息。它标识了 IIS 的地址位宽，数据位宽以及版本号等信息。

表 22-3 标识寄存器

寄存器名称	偏移地址	读/写(R/W)	功能描述
IISVersion	位	缺省值	描述
ADRW	9:8	2'h0	地址总线宽度： 00: 地址宽度 8 位 01: 地址宽度 16 位 10: 地址宽度 32 位 11: 地址宽度 64 位
DATW	5:4	2'h0	数据宽度： 00: 地址宽度 8 位 01: 地址宽度 16 位 10: 地址宽度 32 位 11: 地址宽度 64 位
VER	3:0	4'h0	I2S 版本号

I2S 配置寄存器是配置 I2S 的声道字长，音频数据的采样深度以及各个时钟的分频系数等。

表 22-4 配置寄存器

寄存器名称	偏移地址	读/写(R/W)	功能描述
IISConfig	位	缺省值	描述
LR_LEN	31:24	'h0	左右声道处理的字长。
RES_DEPTH	23:16	'h0	采样深度设置： IIS 采样数据长度，有效范围为 8-32, 如果发送或者接收到的数据宽度小于采样数据长度，则低位补 0；如果发送或者接收到的数据宽度大于采样数据长度，则低位忽略。
BCLK_RATIO	15:8	'h0	位时钟 (BCLK) 分频系数： 位时钟分频系数，分频数为总线时钟频率除以 2x (RATIO+1)
MCLK_RATIO	7:0	'h0	系统时钟 (MCLK) 分频系数， 系统时钟分频系数，分频数为总线时钟频率除以 2x (RATIO+1)，系统时钟作为 Codec 的 sysclk

控制寄存器用于配置 IIS 的工作使能信号，缓存 FIFO 的存储状态以及中断相关信息状态。

表 22-5 控制寄存器

寄存器名称	偏移地址	读/写 (R/W)	功能描述
IISControl	位	缺省值	描述
MASTER	15	'h0	1: IIS 工作于主模式
MSB/LSB	14	'h0	1: 高位在左端 0: 高位在右端
RX_EN	13	'h0	控制器接收使能, 为 1 时有效, 开始接收数据
TX_EN	12	'h0	控制器发送使能, 为 1 时有效, 开始发送数据
RX_DMA_EN	11	'h0	DMA 接收使能, 为 1 时有效
Reserved	10: 8	'h0	
TX_DMA_EN	7	'h0	DMA 发送使能, 为 1 时有效
Reserved	6:2	'h0	
RX_INT_EN	1	'h0	RX 中断使能, 为 1 时使能中断, 为 0 时禁止
TX_INT_EN	0	'h0	TX 中断使能, 为 1 时使能中断, 为 0 时禁止

### DMA 命令寄存器

该寄存器用来控制 I2S 内部的 DMA 控制器（共 2 个），其中 DMA0 用于放音，DMA1 用于录音。DMA 控制器的详细描述见下节。

偏移量：0x100/0x110（DMA0/DMA1）

复位值：00000000h

位域	名称	访问	描述
63:5	ask_addr	R/W	DMA 描述符地址的 bit[63:5], 低 5 位为 0
5	Reserved	R/W	保留
4	dma_stop	R/W	停止 DMA 操作。DMA 控制器完成当前数据读写后停止。
3	dma_start	R/W	开始 DMA 操作。DMA 控制器读取描述符地址(ask_addr)后将些位清零。
2	ask_valid	R/W	DMA 工作寄存器写回到(ask_addr)所指向的内存, 完成后清零。
1	Reserved	R/W	保留
0	dma_64bit	R/W	DMA 控制器 64 位地址支持

## 22.3 配置操作

I2S 正常工作，需要先配置好 CODEC 芯片，然后配置 I2S 控制器的配置寄存器和控制寄存器。

7A2000 芯片通过 I2S 接口和 CODEC 芯片通信，CODEC 芯片作为 I2S 总线上的从设备，详细地址、寄存器和配置方法请参考具体 CODEC 芯片的数据手册。配置完 CODEC 之后，需要对 I2S 控制器进行配置。可以将一些配置信息写入标志寄存器（I2SVersion），以供查询。先配置好 I2SConfig 寄存器，再配置 I2SControl 寄存器。

I2SConfig 寄存器建议 LR\_LEN 和 RES\_DEPTH 配成一样，不至于在传输过程中丢数据或者空数据。BCK 时钟根据配置 CODEC 的采样频率、采样深度和倍频系数来计算，计算公式如下：

$BCK=256xfs$ (或者  $512xfs$ )或者 $(768xfs)$ ；(详细见 CODEC 手册推荐配置)

$BCK\_RATIO = Freq\_SDRAM / (256xfs)/2-1$ ;

双通道的话，计算公式下：

$BCK = RES\_DEPTH \times 2 \times fs$ ;

$BCK\_RATIO= Freq\_APB / (RES\_DEPTH \times 2 \times fs) / 2 - 1$ ;

其中  $fs$  为配置的采样频率(即音频文件的码率)，在 7A2000 中，Freq\_APB 为 50MHz。发送和读取数据时，需要先配置好 DMA，再配置控制器，以免发生数据丢失。

MCK 时钟根据配置 CODEC 的采样频率和倍频系数来计算，计算公式如下：

$MCK = 256xfs$ (或者  $512xfs$ ) 或者 $(768xfs)$ ；(详细见 CODEC 手册推荐配置)

$MCK\_RATIO= Freq\_APB / (256 \times fs) / 2 - 1$ 。

## 22.4 DMA 控制器

### 22.4.1 DMA 控制器结构描述

桥片中包含 2 个 DMA 控制器，用来实现内存与 I2S 之间数据搬移，可以节省资源提高系统数据传输的效率。

DMA 的传送数据的过程由三个阶段组成：

1. 传送前的预处理：由 CPU 配置 DMA 描述符相关的寄存器。
2. 数据传送：在 DMA 控制器的控制下自动完成。
3. 传送结束处理：发送中断请求。

本 DMA 控制器限定为以字 (4Byte) 为单位的数据搬运。

DMA 控制器支持 64 位地址空间，这主要通过 dma\_64bit 来控制，当该位设置为 1 时表示 DMA 控制器工作在 64 位地址空间，反之为 32 位地址空间。在 64 位地址模式下，需要扩展 DMA\_ORDER\_ADDR 和 DMA\_SADDR 为 64 位寄存器。

### 22.4.2 DMA 描述符

#### DMA\_ORDER\_ADDR\_LOW

偏移地址： 0x0

复位值：0x00000000

位域	位域名称	位宽	访问	描述
31:1	dma_order_addr	31	R/W	存储器内部下一描述符地址寄存器（低 32 位）
0	Dma_order_en	1	R/W	描述符是否有效信号

说明：存储下一个 DMA 描述符的地址，dma\_order\_en 是下个 DMA 描述符的使能位，如果该位为 1 表示下个描述符有效，该位为 0 表示下个描述符无效，不执行操作，地址 16 字节对齐。在配置 DMA 描述符时，该寄存器存放的是下个描述符的地址，执行完该次 DMA 操作后，通过判断 dma\_order\_en 信号确定是否开始下次 DMA 操作。在 64 位地址模式下，该寄存器存储低 32 位地址。

### DMA\_SADDR

偏移地址：0x4

复位值：0x00000000

位域	位域名称	位宽	访问	描述
31:0	dma_saddr	32	R/W	DMA 操作的系统内存地址（低 32 位）

说明：DMA 操作分为：内存读：从内存中读数据，保存在 DMA 控制器的缓存中，然后写入 I2S 设备，该寄存器指定了读内存的地址；内存写：从 I2S 设备读数据保存在 DMA 缓存中，当 DMA 缓存中的数据超过一定数目，就往内存中写，该寄存器指定了写内存的地址。在 64 位地址模式下，该寄存器存储低 32 位地址。

### DMA\_DADDR

偏移地址：0x8

复位值：0x00000000

位域	位域名称	位宽	访问	描述
27:0	dma_daddr	28	R/W	DMA 操作的 I2S 设备地址

### DMA\_LENGTH

偏移地址：0xc

复位值：0x00000000

位域	位域名称	位宽	访问	描述
31:0	dma_length	32	R/W	传输数据长度寄存器

说明：代表一块被搬运内容的长度，单位是字。当搬运完 length 长度的字之后，开始下个 step 即下一个循环。开始新的循环，则再次搬运 length 长度的数据。当 step 变为 1，单个 DMA 描述符操作结束，开始读下个描述符。

### DMA\_STEP\_LENGTH

偏移地址：0x10

复位值：0x00000000

位域	位域名称	位宽	访问	描述
31:0	dma_step_length	32	R/W	数据传输间隔长度寄存器

说明：间隔长度说明两块被搬运内存数据块之间的长度，前一个 step 的结束地址与后一个 step 的开始地址之间的间隔。

### DMA\_STEP\_TIMES

偏移地址：0x14

复位值：0x00000000

位域	位域名称	位宽	访问	描述
31:0	dma_step_times	32	R/W	数据传输循环次数寄存器

说明：循环次数说明在一次 DMA 操作中需要搬运的块的数目。如果只想搬运一个连续的数据块，循环次数寄存器的值可以赋值为 1。

### DMA\_CMD

偏移地址：0x18

复位值：0x00000000

位域	位域名称	位宽	访问	描述
14:13	Dma_cmd	2	R/W	源、目的地址生成方式
12	dma_r_w	1	R/W	DMA 操作类型，“1”为读 ddr2 写设备，“0”为读设备写 ddr2
11:8	dma_write_state	4	R/W	DMA 写数据状态
7:4	dma_read_state	4	R/W	DMA 读数据状态
3	dma_trans_over	1	R/W	DMA 执行完被配置的所有描述符操作
2	dma_single_trans_over	1	R/W	DMA 执行完一次描述符操作
1	dma_int	1	R/W	DMA 中断信号
0	dma_int_mask	1	R/W	DMA 中断是否被屏蔽掉

说明：dma\_single\_trans\_over=1 指一次 DMA 操作执行结束，此时 length=0 且 step\_times=1，开始取下个 DMA 操作的描述符。下个 DMA 操作的描述符地址保存在 DMA\_ORDER\_ADDR 寄存器中，如果 DMA\_ORDER\_ADDR 寄存器中 dma\_order\_en=0，则 dma\_trans\_over=1，整个 dma 操作结束，没有新的描述符要读；如果 dma\_order\_en=1，则 dma\_trans\_over 置为 0，开始读下个 dma 描述符。dma\_int 为 DMA 的中断，如果没有中断屏蔽，在一次配置的 DMA 操作结束后发生中断。CPU 处理完中断后可以直接将其置低，也可以等到 DMA 进行下次传输时自动置低。dma\_int\_mask 为对应 dma\_int 的中断屏蔽。dma\_read\_state 说明了 DMA 当前的读状态。dma\_write\_state 说明了 DMA 当前的写状态。

DMA 写状态(WRITE\_STATE[3:0])描述，DMA 包括以下几个写状态：



Write_state	[3:0]	描述
Write_idle	4' h0	写状态正处于空闲状态
W_ddr_wait	4' h1	Dma 判断需要执行读设备写内存操作，并发起写内存请求，但是内存还没准备好响应请求，因此 dma 一直在等待内存的响应
Write_ddr	4' h2	内存接收了 dma 写请求，但是还没有执行完写操作
Write_ddr_end	4' h3	内存接收了 dma 写请求，并完成写操作，此时 dma 处于写内存操作完成状态
Write_dma_wait	4' h4	Dma 发出将 dma 状态寄存器写回内存的请求，等待内存接收请求
Write_dma	4' h5	内存接收写 dma 状态请求，但是操作还未完成
Write_dma_end	4' h6	内存完成写 dma 状态操作
Write_step_end	4' h7	Dma 完成一次 length 长度的操作（也就是说完成一个 step）

DMA 读状态(READ\_STATE[3:0])描述，DMA 包括以下几个读状态：

Read_state	[3:0]	描述
Read_idle	4' h0	读状态正处于空闲状态
Read_ready	4' h1	接收到开始 dma 操作的 start 信号后，进入准备好状态，开始读描述符
Get_order	4' h2	向内存发出读描述符请求，等待内存应答
Read_order	4' h3	内存接收读描述符请求，正在执行读操作
Finish_order_end	4' h4	内存读完 dma 描述符
R_ddr_wait	4' h5	Dma 向内存发出读数据请求，等待内存应答
Read_ddr	4' h6	内存接收 dma 读数据请求，正在执行读数据操作
Read_ddr_end	4' h7	内存完成 dma 的一次读数据请求
Read_dev	4' h8	Dma 进入读设备状态
Read_dev_end	4' h9	设备返回读数据，结束此次读设备请求
Read_step_end	4' ha	结束一次 step 操作，step times 减 1

### DMA\_ORDER\_ADDR\_HIGH

偏移地址： 0x20

复位值：0x00000000

位域	位域名称	位宽	访问	描述
31:0	dma_order_addr	32	R/W	存储器内部下一个描述符地址寄存器(高 32 位)

### DMA\_SADDR\_HIGH

偏移地址： 0x24

复位值：0x00000000

位域	位域名称	位宽	访问	描述
31:0	dma_saddr	32	R/W	DMA 操作的内存地址(高 32 位)

## 23 SATA 控制器（D8:F0）

SATA 的特性包括：

- 支持SATA 1代1.5Gbps、SATA2代3Gbps和SATA3代6Gbps的传输
- 兼容串行ATA 3.3规范和AHCI 1.3.1规范

### 23.1 SATA 配置寄存器（D8:F0）

表 23-1 SATA 控制器的 PCI 配置头

地址偏移	简称	描述	默认值	访问类型
00h-01h	VID	Vendor ID	0014h	RO
02h-03h	DID	Device ID	7A18h	RO
04h-05h	PCICMD	PCI Command	0000h	R/W, RO
08h	RID	Revision ID	00h	RO
09h	PI	Programming Interface	01h	RO
0Ah	SCC	Sub Class Code	06h	RO
0Bh	BCC	Base Class Code	01h	RO
0Ch	CLS	Cache Line Size	10h	RO
0Eh	HEADTYP	Header Type	00h	RO
24h-27h	ABAR	AHCI Base Address	00000000h	R/W, RO
2Ch-2Dh	SVID	Subsystem Vendor ID	0000h	RO
2Eh-2Fh	SID	Subsystem Identification	0000h	RO
3Ch	INT_LN	Interrupt Line	FFh	R/W
3Dh	INT_PN	Interrupt Pin	00h	RO

注：表中未列出的地址空间表示保留。

下面列出与 PCI 配置头规范稍有不同的寄存器及其描述。

#### PCICMD-PCI 命令寄存器（SATA-D8:F0）

地址偏移：04-05h

属性：R/W, RO

默认值：0000h

大小：16 位

位域	名称	访问	描述
15:2	Reserved	RO	保留
1	Memory Space Enable	R/W	该位用来控制是否使能对 SATA 控制寄存器的访问。 0：禁止访问； 1：使能对 SATA 控制寄存器的访问。在将该位配置为 1 之前，必须先配置 PCNL_BAR 寄存器。
0	Reserved	RO	保留

## 23.2 SATA 控制寄存器描述

SATA 的基地址是由 SATA 的 BAR0 给定，寄存器的定义和协议标准定义完全一致。

表 23-2 SATA 控制寄存器

偏移地址	位宽	名称	描述
0x000	32	CAP	HBA 特性寄存器
0x004	32	GHC	全局 HBA 控制寄存器
0x008	32	IS	中断状态寄存器
0x00c	32	PI	端口寄存器
0x010	32	VS	AHCI 版本寄存器
0x014	32	CCC_CTL	命令完成合并控制寄存器
0x018	32	CCC_PORTS	命令完成合并端口寄存器
0x024	32	CAP2	HBA 特性扩展寄存器
0x0A0	32	BISTAFR	BIST 激活 FIS
0x0A4	32	BISTCR	BIST 控制寄存器
0x0A8	32	BISTCTR	BIST FIS 计数寄存器
0x0AC	32	BISTSR	BIST 状态寄存器
0x0B0	32	BISTDECR	BIST 双字错计数寄存器
0x0BC	32	OOBR	OOB 寄存器
0x0E0	32	TIMER1MS	1ms 计数寄存器
0x0E8	32	GPARAM1R	全局参数寄存器 1
0x0EC	32	GPARAM2R	全局参数寄存器 2
0x0F0	32	PPARAMR	端口参数寄存器
0x0F4	32	TESTR	测试寄存器
0x0F8	32	VERIONR	版本寄存器
0x0FC	32	IDR	ID 寄存器
0x100	32	PO_CLB	命令列表基地址低 32 位
0x104	32	PO_CLBU	命令列表基地址高 32 位
0x108	32	PO_FB	FIS 基地址低 32 位
0x10c	32	PO_FBU	FIS 基地址高 32 位
0x110	32	PO_IS	中断状态寄存器
0x114	32	PO_IE	中断使能寄存器
0x118	32	PO_CMD	命令寄存器
0x120	32	PO_TFD	任务文件数据寄存器
0x124	32	PO_SIG	签名寄存器
0x128	32	PO_SSTS	SATA 状态寄存器
0x12C	32	PO_SCTL	SATA 控制寄存器
0x130	32	PO_SERR	SATA 错误寄存器
0x134	32	PO_SACT	SATA 激活寄存器

0x138	32	P0_CI	命令发送寄存器
0x13C	32	P0_SNTF	SATA 命令通知寄存器
0x170	32	P0_DMACR	DMA 控制寄存器
0x178	32	P0_PHYCR	PHY 控制寄存器
0x17C	32	P0_PHYSR	PHY 状态寄存器
0x180	32	P1_CLB	命令列表基地址低 32 位
0x184	32	P1_CLBU	命令列表基地址高 32 位
0x188	32	P1_FB	FIS 基地址低 32 位
0x18c	32	P1_FBU	FIS 基地址高 32 位
0x190	32	P1_IS	中断状态寄存器
0x194	32	P1_IE	中断使能寄存器
0x108	32	P1_CMD	命令寄存器
0x1a0	32	P1_TFD	任务文件数据寄存器
0x1a4	32	P1_SIG	签名寄存器
0x1a8	32	P1_SSTS	SATA 状态寄存器
0x1aC	32	P1_SCTL	SATA 控制寄存器
0x1b0	32	P1_SERR	SATA 错误寄存器
0x1b4	32	P1_SACT	SATA 激活寄存器
0x1b8	32	P1_CI	命令发送寄存器
0x1bC	32	P1_SNTF	SATA 命令通知寄存器
0x1f0	32	P1_DMACR	DMA 控制寄存器
0x1f8	32	P1_PHYCR	PHY 控制寄存器
0x1fC	32	P1s_PHYSR	PHY 状态寄存器

## 24 PCIE 控制器 (D9-16,19-20:F0)

桥片的 PCIE 分为 4 个模块: PCIE\_F0, PCIE\_F1, PCIE\_H, PCIE\_G0, 共 32 个 lane。

PCIE\_F0 包括 4 个 lane, 可当作一个 x4 的 PCIE 或者 4 个 x1 的 PCIE 使用。其中, 端口 0 在非 x4 模式下控制 lane0, 在 x4 模式下控制 lane0~3; 在非 x4 模式下, 端口 1 控制 lane1, 端口 2 控制 lane2, 端口 3 控制 lane3。

PCIE\_F1 包括 4 个 lane, 可当作一个 x4 的 PCIE 或者 2 个 x1 的 PCIE 使用。其中, 端口 0 在非 x4 模式下控制 lane0, 在 x4 模式下控制 lane0~3; 在非 x4 模式下, 端口 1 控制 lane1, lane2 和 lane3 不可用。

PCIE\_H 包括 8 个 lane, 可当作一个 x8 的 PCIE 或者 2 个 x4 的 PCIE 使用。其中, 端口 0 在非 x8 模式下控制 lane0~3, 在 x8 模式下控制 lane0~7; 端口 1 在非 x8 模式下控制 lane4~7。

PCIE\_G0 包括 16 个 lane, 可当作 2 个 x8 的 PCIE 使用。其中, 端口 0 控制 lane0~7, 端口 1 控制 lane8~15。

PCIE 共包含 10 个端口 (Port), 每个端口对应一个 PCIE 控制器。PCIE 控制器作为 RC 使用时内部都有一个 TYPE1 类型的配置头; 作为 EP 使用时内部都有一个 TYPE0 类型的配置头。每个 PCIE 端口均有自己独立的 PCI 地址空间。

桥片的 PCIE 控制器中, F0 和 H 的 port0 可以作为 RC 使用也可以作为 EP 使用, 其它的控制端口仅可以作为 RC 使用, 不能作为 EP。

### 24.1 PCI 配置寄存器

下表列出 PCIE 端口的配置头缺省值, 不同端口的 Device ID 可能不同, 其他字段都相同。

表 24-1 PCIE 控制器的 PCI 配置头

地址偏移	简称	描述	默认值	访问类型
00h-01h	VID	Vendor ID	0014h	RO
02h-03h	DID	Device ID	见寄存器描述	RO
04h-05h	PCICMD	PCI Command	0000h	R/W, RO
06h-07h	PCISTS	PCI Status	0010h	RO
08h	RID	Revision ID	00h	RO
09h	PI	Programming Interface	00h	RO
0Ah	SCC	Sub Class Code	04h	RO
0Bh	BCC	Base Class Code	06h	RO
0Ch	CLS	Cache Line Size	00h	RO
0Dh	PLT	Primary Latency Timer	00h	RO
0Eh	HEADTYP	Header Type	01h	RO

10h-17h	CNL_BAR	Control Block Base Address Register	0000000000000004h	R/W,RO
18h	PBNUM	Primary Bus Number	00h	R/W
19h	SBNUM	Secondary Bus Number	00h	R/W
1Ah	SuBNUM	Subordinate Bus Number	00h	R/W
1Bh	SLT	Secondary Latency Timer	00h	RO
1Ch	IOBASE	I/O Base	01h	R/W
1Dh	IOLMT	I/O Limit	01h	R/W
1Eh-1Fh	SSTS	Secondary Status	0000h	RO
20h-21h	MBASE	Memory Base	0000h	R/W
22h-23h	MLMT	Memory Limit	0000h	R/W
25h-24h	PMBASE	Prefetchable Memory Base	0000h	R/W
27h-26h	PMLMT	Prefetchable Memory Limit	0000h	R/W
28h-2Bh	PMBU32	Prefetchable Memory Base Upper 32 Bits	00000000h	R/W
2Ch-2Fh	PMLU32	Prefetchable Memory Limit Upper 32 Bits	00000000h	R/W
30h-31h	IOBU	I/O Base Upper 16 Bits	0000h	R/W
32h-33h	IOLMTU	I/O Limit Upper 16 Bits	0000h	R/W
34h	CAPP	Capabilities Pointer	40h	RO
3Ch	INT_LN	Interrupt Line	FFh	R/W
3Dh	INT_PN	Interrupt Pin	01h	RO
3Eh-3Fh	BCTRL	Bridge Control Register	0000h	R/W

注：表中未列出的地址空间表示保留。

下面列出与 PCI 配置头规范稍有不同的寄存器及其描述。

### DID-设备标识寄存器（PCIE）

地址偏移：02-03h

属性：RO

默认值：见描述

大小：16 位

位域	名称	访问	描述
15:0	DID	RO	PCIE 设备标识寄存器。各个 PCIE 端口对应的 DID 见表 22-2。

表 24-2 PCIE 端口 DID 表

PCI 设备号	描述	设备标识寄存器
D9:F0	PCIE_F0 端口 0	7A49h
D10:F0	PCIE_F0 端口 1	7A39h
D11:F0	PCIE_F0 端口 2	7A39h
D12:F0	PCIE_F0 端口 3	7A39h
D13:F0	PCIE_F1 端口 0	7A49h
D14:F0	PCIE_F1 端口 1	7A39h

D15:F0	PCIE_G0 端口 0	7A69h
D16:F0	PCIE_G0 端口 1	7A59h
D19:F0	PCIE_H 端口 0	7A59h
D20:F0	PCIE_H 端口 1	7A49h

## 24.2 地址空间划分

桥片中的 PCIE 控制器内有标准的 PCIE 配置头，因此 PCIE 控制器的内部寄存器以及其下游设备的地址空间都通过其配置头的信息来管理。配置头中地址相关的寄存器在 PCI 设备扫描时确定。

每个 PCIE 端口作为桥片中的独立设备，每个端口都包含一个 PCIE 配置头。当 F0、F1 中的 PCIE 工作在 X4 模式时，其他 X1 的端口软件不可见，只有当 PCIE 工作在 X1 模式时才可以访问其他 X1 端口。当 H 中 PCIE 工作在 X8 模式时，H 模块的 port1 的 X4 控制器软件不可见，H 模块的 port1 只有在 H 模块的 2X4 模式下可以被访问。当 G0 仅可以工作在 2x8 模式，port0 和 port1 软件均可见。

对于每一个 PCIE 端口，其地址空间可以分为以下几部分：

**配置头地址空间：**该部分空间对应 PCIE 的配置头，通过配置请求来访问，最大 8KB。其中低 4KB 通过将配置请求的 func 设为 0 来访问，用于访问标准配置头；高 4KB 通过将配置请求的 func 设为 1 来访问，用于改写标准配置头中的一些只读寄存器。当控制器作为 EP 使用时，来自外部 PCIE 总线的所有设备号为 0 的 TYPE0 访问被用于该 EP 端口的 PCIE header。

**配置访问地址空间：**该部分地址空间用于通过配置请求访问 PCIE 控制器的下游设备配置头信息。根据下游设备的 Bus 号，由 PCIE 控制器决定发送 TYPE0 类型还是 TYPE1 类型的配置访问。当控制器作为 EP 使用时，来自外部 PCIE 总线的所有设备号非 0 的 TYPE0 访问被用于通过芯片内部的互连网络访问芯片内部各功能接口的 PCI header；所有来自外部 PCIE 总线的 TYPE1 访问则依据其总线号通过内部互连来访问芯片的其它作为 RC 的 PCIE 端口。

以上两个地址空间的地址由配置地址空间基地址、BUS 号、设备号、功能号以及寄存器偏移地址计算得出，访问以字为单位。

**PCIE 控制器内部寄存器空间：**该部分地址空间用于访问 PCIE 控制器的内部寄存器。这些寄存器用于控制 PCIE 控制器的行为和特性，与 PCIE 配置头空间属于两个地址空间。该地址空间为 MEM 类型，64 位地址空间，大小为 4KB，基地址等于 64 位 BAR0 的值，该值在初始化时由 PCI 扫描软件分配。

**MEM 地址空间：**该部分地址空间包含了 PCIE 控制器下游设备的所有 MEM 地址空间。对于 32 位地址空间，由 PCIE 配置头的 memory base 和 memory limit 决定；对于 64 位地址空间，由 PCIE 配置头的 prefetchable memory base（组合 upper 32bits）和 prefetchable memory limit（组合 upper 32bits）决定。该段地址空间由 PCIE 配置头的 command 寄存器 bit1 位来使能控制。当控制器作为 EP 使用时，来自外部 PCIE 总线的所有 MEM 访问将通过芯片内部的互连网络直接访问芯片内部的资源。

IO 地址空间: 该部分地址空间包含了 PCIE 控制器下游设备的所有 IO 地址空间。由 PCIE 配置头的 io base (组合 upper 16bits) 和 io limit (组合 upper 16bits) 决定。该段地址空间由 PCIE 配置头的 command 寄存器 bit0 位来使能控制。当控制器作为 EP 使用时, 来自外部 PCIE 总线的所有 IO 访问将通过芯片内部的互连网络直接访问芯片内部的资源。

对于 MEM 地址空间和 IO 地址空间来说, 如果在 X1 工作模式下, 某个 X1 端口下游没有连接设备, 通过设置 command 寄存器的 bit0 和 bit1 为 0 即可禁用其 MEM 和 IO 地址空间。



## 25 SPI 控制器 (D22:F0)

### 25.1 SPI 配置寄存器 (D22:F0)

表 25-1 SPI 控制器的 PCI 配置头

地址偏移	简称	描述	默认值	访问类型
00h-01h	VID	Vendor ID	0014h	RO
02h-03h	DID	Device ID	7A1Bh	RO
04h-05h	PCICMD	PCI Command	0000h	R/W, RO
08h	RID	Revision ID	00h	RO
09h	PI	Programming Interface	00h	RO
0Ah	SCC	Sub Class Code	80h	RO
0Bh	BCC	Base Class Code	08h	RO
0Ch	CLS	Cache Line Size	10h	RO
0Eh	HEADTYP	Header Type	00h	RO
10h-17h	CBAR	Control Base Address Register	0000000000000004h	R/W,RO
18h-1Fh	MBAR	Memory Base Address Register	0000000000000004h	R/W,RO
2Ch-2Dh	SVID	Subsystem Vendor ID	0000h	RO
2Eh-2Fh	SID	Subsystem Identification	0000h	RO
3Ch	INT_LN	Interrupt Line	FFh	R/W
3Dh	INT_PN	Interrupt Pin	00h	RO

注：表中未列出的地址空间表示保留。

下面列出与 PCI 配置头规范稍有不同的寄存器及其描述。

#### PCICMD-PCI 命令寄存器 (SPI-D22:F0)

地址偏移：04-05h

属性：R/W, RO

默认值：0000h

大小：16 位

位域	名称	访问	描述
15:2	Reserved	RO	保留
1	Memory Space Enable	R/W	该位用来控制是否使能对 SPI 控制寄存器的访问。 0：禁止访问； 1：使能对 SPI 控制寄存器的访问。在将该位配置为 1 之前，必须先配置 PCNL_BAR 寄存器。
0	Reserved	RO	保留

SPI 控制器包括两个地址空间：控制寄存器空间和内存空间，仅内存空间支持 QSPI (四线模式)。

#### CBAR-控制基址寄存器

该寄存器用来配置 SPI 控制器的控制寄存器的基地址。

地址偏移：10-13h 属性：R/W, RO

默认值：00000004h 大小：32 位

位域	名称	访问	描述
31:12	Base Address	RW	软件向该寄存器域写入分配给 SPI 控制寄存器的基地址的低地址。
11:4	Memory Size	RO	SPI 控制寄存器的地址空间大小为 4KB。
3	Prefetchable Memory	RO	设置为 0，表示不可预取。
2:1	Memory Type	RO	设置为 10b，表示 64 位 BAR。
0	Memory/ I/O Space	RO	设置为 0，表示 Memory 空间 BAR。

地址偏移：14-17h 属性：R/W

默认值：00000000h 大小：32 位

位域	名称	访问	描述
31:0	Base Address	RW	软件向该寄存器域写入分配给 SPI 控制寄存器的基地址的高 32 位地址。

## MBAR-内存空间基址寄存器

该寄存器用来配置 SPI 的内存空间基地址。

地址偏移：18-1Bh 属性：R/W, RO

默认值：00000004h 大小：32 位

位域	名称	访问	描述
31:24	Base Address	RW	软件向该寄存器域写入分配给 SPI 内存空间的基地址的 bit[31:24]。
23:4	Memory Size	RO	SPI 内存空间大小为 16MB。
3	Prefetchable Memory	RO	设置为 0，表示不可预取。
2:1	Memory Type	RO	设置为 10b，表示 64 位 BAR。
0	Memory/ I/O Space	RO	设置为 0，表示 Memory 空间 BAR。

地址偏移：1C-1Fh 属性：R/W

默认值：00000000h 大小：32 位

位域	名称	访问	描述
31:0	Base Address	RW	软件向该寄存器域写入分配给 SPI 内存空间的基地址的高 32 位地址。

## 25.2 SPI 控制寄存器

表 25-2 SPI 控制寄存器列表

偏移	名称	描述
0	SPCR	控制寄存器
1	SPSR	状态寄存器

2	TxFIFO/RxFIFO	数据寄存器
3	SPER	外部寄存器
4	SFC_PARAM	参数控制寄存器
5	SFC_SOFTCS	片选控制寄存器
6	SFC_TIMING	时序控制寄存器
7	CTRL	自定义控制寄存器
8	CMD	自定义命令寄存器
9	BUF0	自定义数据寄存器 0
10	BUF1	自定义数据寄存器 1
11	TIMER0	自定义时序寄存器 0
12	TIMER1	自定义时序寄存器 1
13	TIMER2	自定义时序寄存器 2

### 控制寄存器(SPCR)

偏移地址: 0x0

位域	名称	访问	初值	描述
7	spie	R/W	0	中断输出使能信号高有效
6	spe	R/W	0	系统工作使能信号高有效
5	-	-	0	保留
4	mstr	-	1	master 模式选择位, 此位一直保持 1
3	cpol	R/W	0	时钟极性位
2	cpha	R/W	0	时钟相位位 1 则相位相反, 为 0 则相同
1:0	spr	R/W	0	sclk_o 分频设定, 需要与 sper 的 spre 一起使用

### 状态寄存器(SPSR)

偏移地址: 0x1

位域	名称	访问	初值	描述
7	spif	R/W	0	中断标志位 1 表示有中断申请, 写 1 则清零
6	wcol	R/W	0	写寄存器溢出标志位为 1 表示已经溢出, 写 1 则清零
5:4	-	-	0	保留
3	wfull	R	0	写寄存器满标志 1 表示已经满
2	wfempty	R	1	写寄存器空标志 1 表示空
1	rffull	R	0	读寄存器满标志 1 表示已经满
0	rfempty	R	1	读寄存器空标志 1 表示空

### 数据寄存器(TxFIFO/RxFIFO)

偏移地址: 0x2

位域	名称	访问	初值	描述
7:0	TxFIFO RxFIFO	W R	-	数据发送端口 数据接收端口

### 外部寄存器(SPER)

偏移地址: 0x3

位域	名称	访问	初值	描述
7:6	icnt	R/W	0	传输完多少个字节后发中断 00: 1 01: 2 10: 3 11: 4
5:3	-	-	-	保留
2	mode	R/W	0	spl 接口模式控制 0: 采样与发送时机同时 1: 采样与发送时机错开半周期
1:0	spre	R/W	0	与 spr 一起设定分频的比率

表 25-3 SPI 分频系数

spre	00	00	00	00	01	01	01	01	10	10	10	10
spr	00	01	10	11	00	01	10	11	00	01	10	11
分频系数	2	4	16	32	8	64	128	256	512	1024	2048	4096

### 参数控制寄存器(SFC\_PARAM)

偏移地址: 0x4

位域	名称	访问	初值	描述
7:4	clk_div	R/W	2	时钟分频数选择 分频系数与{spre, spr}组合相同
3	dual_io	R/W	0	双 I/O 模式, 优先级高于快速读
2	fast_read	R/W	0	快速读模式
1	burst_en	R/W	0	SPI flash 支持连续地址读模式
0	memory_en	R/W	1	SPI flash 读使能, 无效时 csn[0]可由软件控制。

### 片选控制寄存器(SFC\_SOFTCS)

偏移地址: 0x5

位域	名称	访问	初值	描述
7:4	csn	R/W	0	csn 引脚输出值
3:0	csen	R/W	0	为 1 时对应位的 csn 线由 7:4 位控制

### 时序控制寄存器(SFC\_TIMING)

偏移地址: 0x6

位域	名称	访问	初值	描述
7:4	-	-	-	保留
3	quad_io	R/W	0	4线模式使能, 1有效
2	tFAST	R/W	0	SPI flash 读采样模式 0: 上沿采样, 间隔半个 SPI 周期 1: 上沿采样, 间隔一个 SPI 周期
1:0	tCSH	R/W	3	SPI Flash 的片选信号最短无效时间, 以分频后时钟周期 T 计算 00: 1T 01: 2T 10: 4T 11: 8T

### 自定义控制寄存器 (CTRL)

偏移地址: 0x8

位域	名称	访问	初值	描述
7:4	nbyte	RW	0	一次传输的字节数
3:2	reserve	RW	0	保留
1	nbmode	RW	0	多字节传输模式
0	start	RW	0	开始多字节传输, 完成后自动清零

### 自定义命令寄存器 (CMD)

偏移地址: 0x9

位域	名称	访问	初值	描述
7:0	cmd	RW	0	设置发送给 spi flash 的命令

### 自定义数据寄存器 0 (BUF0)

偏移地址: 0xa

位域	名称	访问	初值	描述
7:0	buf0	RW	0	向 SPI 发送写命令时, 该寄存器配置发送的第一个字节的数据; 向 SPI 发送读命令时, 该寄存器存储第一个读回来的数据。

### 自定义数据寄存器 1 (BUF1)

偏移地址: 0xb

位域	名称	访问	初值	描述
7:0	buf1	RW	0	向 SPI 发送写命令时, 该寄存器配置发送的第二个字节的数据; 向 SPI 发送读命令时, 该寄存器存储第二个读回来的数据。

### 自定义时序寄存器 0 (TIMER0)

偏移地址：0xc

位域	名称	访问	初值	描述
7:0	time0	RW	0	自定义命令所需时间值的低 8 位

### 自定义时序寄存器 1 (TIMER1)

偏移地址：0xd

位域	名称	访问	初值	描述
7:0	time1	RW	0	自定义命令所需时间值的中间 8 位

### 自定义时序寄存器 2 (TIMER2)

偏移地址：0xe

位域	名称	访问	初值	描述
7:0	time2	RW	0	自定义命令所需时间值的高 8 位

## 25.3 SPI 软件编程指南

### SPI 主控制器的读写操作

#### 模块初始化

1. 停止 SPI 控制器工作，对控制寄存器 spcr 的 spe 位写 0
2. 重置状态寄存器 spsr，对寄存器写入 1100\_0000b
3. 设置外部寄存器 sper，包括中断申请条件 sper[7:6]和分频系数 sper[1:0]，具体参考寄存器说明
4. 配置 SPI 时序，包括 spcr 的 cpol、cpa 和 sper 的 mode 位。mode 为 1 时是标准 SPI 实现，为 0 时为兼容模式。
5. 配置中断使能，spsr 的 spie 位
6. 启动 SPI 控制器，对控制寄存器 spcr 的 spe 位写 1

#### 模块的发送/传输操作

1. 往数据传输寄存器写入数据
2. 传输完成后从数据传输寄存器读出数据。由于发送和接收同时进行，即使 SPI 从设备没有发送有效数据也必须进行读出操作。

#### 中断处理

1. 接收到中断申请
2. 读状态寄存器 spsr 的值，若 spsr[2]为 1 则表示数据发送完成，若 spsr[0]为 1 则表示已经接收数据

3. 读或写数据传输寄存器
4. 往状态寄存器 spsr 的 spif 位写 1，清除控制器的中断申请

## 硬件 SPI Flash 读

### 初始化

1. 将 SFC\_PARAM 的 memory\_en 位写 1。当 SPI 被选为启动设备时此位复位为 1。
2. 设置读参数(时钟分频、连续地址读、快速读、双 I/O、tCSH 等)。这些参数复位值均为最保守的值。

### 更改参数

如果所使用的 SPI Flash 支持更高的频率或者提供增强功能，修改相应参数可以大大加快 Flash 的访问速度。参数的修改不需要关闭 SPI Flash 读使能(memory\_en)。具体参考寄存器说明。

## 混合访问 SPI Flash 和 SPI 主控制器

### 对 SPI Flash 进行读以外的访问

将 SPI Flash 读使能关闭后，软件就可直接控制 csn[0]，并通过 SPI 主控制器访问 SPI 总线。这意味着在进行此操作时，不能从 SPI Flash 中取指。

除了读以外，SPI Flash 还实现了很多命令(如擦除、写入)，具体参见相关 Flash 的文档。

## SPI 双线四线使用指南

除了传统的单线模式，SPI 控制器还支持以双线(dual mode)和四线(quad mode)两种工作模式从 SPI flash 启动。通过设置 dual\_io 寄存器可以使 SPI 控制器进入双线模式，设置 quad\_io 寄存器可以使 SPI 控制器进入四线模式。可以在 BIOS 代码的前几条指令中增加对这两个寄存器的配置代码，配置完成后控制器即按照配置对应的工作模式进行取指，以此可提高开机速度。

需要注意的是，有的 SPI FLASH 默认并没有使能四线模式，或者在四线模式下需要配置时序相关的参数(如 dummy clocks)。为了增加 SPI 控制器对各种 FLASH 的适用性，本控制器增加自定义的寄存器(0x8-0xe)。其具体使用方法为：

1. 设置自定义命令寄存器(CMD)(0x9)，该寄存器为向 SPI FLASH 发送的命令；
2. 如果 SPI FLASH 要求本次发送的命令需要过一段时间才完成，则把等待的时间配置到自定义时序寄存器 TIMER0-TIMER2(0xc-0xe)中，否则这些寄存器保持默认值 0；

3. 如果向 SPI FLASH 写配置信息，则需要把配置信息写入自定义数据寄存器 BUF0-BUF1 (0xa-0xb)；如果向 SPI FLASH 读配置信息，则这两个寄存器存储读回来的值；
4. 配置自定义控制寄存器 CTRL[7: 1]其中 CTRL[1] (nbmode)代表将进行多字节传输模式，此次传输字节数通过 CTRL[7:4] (nbyte)给定；
5. 配置自定义控制寄存器 CTRL[0]开始此次传输。

一般来说，所需要配置的寄存器位于 FLASH 的非易失性存储区，所以上配置仅需要配置一次。



## 26 LPC 控制器 (D23:F0)

LPC 控制器具有以下特性:

- 符合LPC1.1规范
- 支持LPC访问超时计数器
- 支持Memory Read/write访问类型
- 支持Firmware Memory Read/Write访问类型 (单字节)
- 支持I/O read/write访问类型
- 支持TPM I/O read/write访问类型
- 支持Memory访问类型地址转换
- 支持Serial IRQ规范, 支持17个中断源

### 26.1 LPC 配置寄存器 (D23:F0)

表 26-1 LPC 控制器的 PCI 配置头

地址偏移	简称	描述	默认值	访问类型
00h-01h	VID	Vendor ID	0014h	RO
02h-03h	DID	Device ID	7A0Ch	RO
04h-05h	PCICMD	PCI Command	0001h	R/W, RO
08h	RID	Revision ID	01h	RO
09h	PI	Programming Interface	00h	RO
0Ah	SCC	Sub Class Code	80h	RO
0Bh	BCC	Base Class Code	08h	RO
0Ch	CLS	Cache Line Size	10h	RO
0Eh	HEADTYP	Header Type	00h	RO
10h-17h	FIXCREG	Fixed Control Register	0000000010002004h	RO
18h-1Fh	FIXMREG	Fixed Memory Register	0000000012000004h	RO
20h-27h	FIXIOREG	Fixed I/O Register	000000FDFC000001h	RO
2Ch-2Dh	SVID	Subsystem Vendor ID	0000h	RO
2Eh-2Fh	SID	Subsystem Identification	0000h	RO
3Ch	INT_LN	Interrupt Line	FFh	R/W
3Dh	INT_PN	Interrupt Pin	01h	RO

**注:** 表中未列出的地址空间表示保留。

下面列出与 PCI 配置头规范稍有不同的寄存器及其描述。

#### PCICMD-PCI 命令寄存器 (LPC-D23:F0)

地址偏移：04-05h

属性：R/W, RO

默认值：0000h

大小：16 位

位域	名称	访问	描述
15:2	Reserved	RO	保留
1	Memory Space Enable	R/W	该位用来控制是否使能对 LPC 控制寄存器和 MEM 空间的访问。 0: 禁止访问; 1: 使能对 LPC 控制寄存器和 MEM 空间的访问。
0	I/O Space Enable	R/W	该位用来控制是否使能对 LPC I/O 空间的访问。LPC I/O 空间的地址固定从 I/O 空间的地址 0 开始。 0: 禁止访问; 1: 使能对 LPC I/O 空间的访问。

### FIXCREG-Fixed 控制寄存器

该寄存器不作为 LPC 配置头的 BAR 使用。

地址偏移：10-17h

属性：RO

默认值：0000000010002004h

大小：64 位

位域	名称	访问	描述
63:0	Reserved	RO	保留

### FIXMREG-Fixed MEM 寄存器

该寄存器不作为 LPC 配置头的 BAR 使用。

地址偏移：18-1Fh

属性：RO

默认值：0000000012000004h

大小：64 位

位域	名称	访问	描述
63:0	Reserved	RO	保留

### FIXIOREG-Fixed I/O 寄存器

该寄存器不作为 LPC 配置头的 BAR 使用。

地址偏移：20-27h

属性：RO

默认值：000000FDFC000001h

大小：64 位

位域	名称	访问	描述
63:0	Reserved	RO	保留

FIXCREG、FIXMREG、FIXIOREG 的地址和 PCI 配置头的 BAR 寄存器相同，但这几个寄存器不作为 LPC 配置头的 BAR 寄存器使用。软件可以通过修改 PCI 配置读函数的方法来绕过该硬件 bug，使得上层软件不受影响。

## 26.2 LPC 地址空间

LPC 控制器包括三个地址空间：控制寄存器空间、MEM 空间、I/O 空间。

LPC 控制寄存器空间用来配置 LPC 控制器。LPC 控制寄存器空间位于桥片的固定设备地址空间内，起始地址为 0x1000,2000，大小为 4KB。

LPC MEM 空间用来访问 LPC 总线上挂载的 Memory/Firmware Memory 设备。LPC MEM 空间位于桥片的固定设备地址空间内，起始地址为 0x1200,0000，大小为 32MB。处理器发往 LPC MEM 空间的访问会被转换成 LPC 协议的 Memory 访问发往 LPC 总线。LPC 控制器发出哪种类型的 Memory 访问，由 LPC 控制器的控制寄存器决定。处理器发往这个地址空间的地址可以进行地址转换。转换后的地址由 LPC 控制器的配置寄存器 LPC\_MEM\_TRANS 设置。

LPC I/O 空间用来访问 LPC 总线上挂载的 I/O 设备，LPC I/O 空间的地址从 PCI I/O 空间的 0 地址开始，大小为 128KB。处理器发往该空间的访问会被转换成 LPC 协议的 I/O 访问发到 LPC 总线。其中 LPC I/O 空间的低 64KB 空间用来访问 LPC I/O 设备，高 64KB 空间用来访问 TPM 设备。

## 26.3 LPC 中断

LPC 控制器内部包括两类中断：SIRQ 中断和访问超时中断。LPC 控制器共支持 17 个 SIRQ 中断，对应中断相关寄存器的比特位 [16:0]。访问超时中断对应中断相关寄存器的比特位 [17]。

SIRQ 中断为电平触发中断，触发电平的值可由寄存器配置。软件应先配置好 SIRQ 中断的触发电平，然后再使能 LPC 控制器的 SIRQ 中断。SIRQ 中断不需要软件清除。

访问超时中断为边沿触发中断，因此，如果发生 LPC 访问超时中断，则软件需要写中断清除寄存器的 bit [17] 来清除该中断。

## 26.4 LPC 控制寄存器

### 控制寄存器 0

地址偏移：00-03h 属性：R/W

默认值：0000FFFFh 大小：4

位域	名称	访问	描述
31	SIRQ_EN	R/W	SIRQ 中断使能控制
23	LPC_MEM_TRANS_EN	R/W	LPC Memory 空间地址转换使能
22:16	LPC_MEM_TRANS	R/W	LPC Memory 空间地址转换后的高 7 位地址 (bit [31:25])。
15:0	LPC_SYNC_TIMEOUT	R/W	LPC 访问超时的阈值 (最小为 64)

### 控制寄存器 1

地址偏移：04-07h 属性：R/W

默认值: 00000000h

大小: 4

位域	名称	访问	描述
31	FIRMWARE_TYPE	R/W	LPC Memory 空间 Firmware Memory 访问类型设置
17:0	LPC_INT_EN	R/W	LPC 中断使能, 每个比特位对应一个中断源。对于每个中断源, 0: 关闭中断; 1: 使能中断。

### LPC 中断状态寄存器

地址偏移: 08-0Bh

属性: RO

默认值: 00000000h

大小: 4

位域	名称	访问	描述
17:0	LPC_INT_SRC	RO	LPC 中断源指示, 每个比特位对应一个中断源。对于每个中断源, 0: 没有中断; 1: 有中断。

### LPC 中断清除寄存器

地址偏移: 0C-0Fh

属性: WO

默认值: 00000000h

大小: 4

位域	名称	访问	描述
17	LPC_TIMEOUT_INT_CLEAR	WO	LPC 访问超时中断清除 (写 1 清除)。比特 17 对应 LPC 访问超 时中断, 写 1 清除, 写 0 无效。

### LPC SIRQ 中断极性寄存器

地址偏移: 10-13h

属性: R/W

默认值: 0000FFBh

大小: 4

位域	名称	访问	描述
16:0	SIRQ_INT_POLARITY	R/W	LPC SIRQ 中断极性寄存器, 每个比特位对应一个中断源。对于每 个中断源, 0: 低电平触发; 1: 高电平触发。