

# LOONGSON

## 龙芯 1C 处理器用户手册

2013 年 7 月

中国科学院计算技术研究所

龙芯中科技术有限公司

自主决定命运, 创新成就未来

北京市海淀区科学院南路10号 100190  
10 Kexueyuan South Road, Zhongguancun  
Haidian District, Beijing

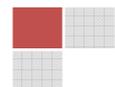


[www.loongson.cn](http://www.loongson.cn)

## 阅读指南

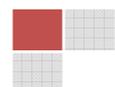
《龙芯 1C 处理器用户手册》主要介绍龙芯 1C 的架构与寄存器描述；软件编程指南介绍对 BIOS 和操作系统开发过程中的常见问题。

关于龙芯 1C 处理器所集成的 LS232 高性能处理器核的相关资料，请参阅《龙芯 LS232 处理器用户手册》。



## 修订历史

日期	编辑	版本	变化
2013-3-14	研发中心	0.0.1	初稿
2013-3-16	研发中心	0.0.2	
2013-3-29	研发中心	0.0.3	补充 DWT、中断配置寄存器描述；CAMIF、UART、OTG 等模块的描述修订
2013-4-15	研发中心	0.1	地址空间修订
2013-4-26	研发中心	0.2	
2013-7-16	广州龙芯	0.4	修订模块描述、寄存器描述等错误地方



## 目录

1	概述	10
1.1	体系结构框图	10
1.2	芯片主要功能	11
1.2.1	处理器核	11
1.2.2	SDRAM 控制器	11
1.2.3	SRAM/NOR FLASH 接口	11
1.2.4	NAND 控制器	11
1.2.5	时钟发生器	12
1.2.6	I <sup>2</sup> S 接口	12
1.2.7	AC97	12
1.2.8	LCD 控制器	12
1.2.9	Camera 接口	12
1.2.10	MAC 控制器	13
1.2.11	USB2.0 控制器	13
1.2.12	SPI	13
1.2.13	I <sup>2</sup> C 总线	14
1.2.14	UART	14
1.2.15	GPIO	14
1.2.16	PWM	14
1.2.17	RTC	14
1.2.18	CAN	15
1.2.19	SDIO	15
1.2.20	ADC	15
2	引脚定义	16
2.1	约定	16
2.2	LCD 接口	16
2.3	SDRAM 接口	16
2.4	I2S 接口	17
2.5	I2C 接口	17
2.6	UART 接口	17
2.7	PWM 接口	18
2.8	ADC 接口	18
2.9	SPI 接口	18
2.10	EJTAG 接口	18
2.11	CAMERA 接口	19
2.12	NAND 接口	19
2.13	MAC 接口	19
2.14	OTG 接口	20
2.15	USB 接口	20
2.16	RTC 接口	21
2.17	时钟配置信号	21
2.18	电源地	21
2.19	初始化信号	21
3	寄存器定义	23
3.1	基本寄存器定义	23
3.2	各模块寄存器定义	25
4	时钟结构	30
4.1	时钟架构	30
4.2	时钟信号说明	30
5	芯片配置与控制	31



5.1	地址空间分配.....	31
5.2	芯片配置寄存器 (CONFREG) .....	32
5.2.1	PLL/SDRAM 频率配置寄存器.....	32
5.2.2	CPU/CAMERA/DC 频率配置寄存器 .....	33
5.2.3	SDRAM 参数寄存器 .....	34
5.2.4	SHUT_CTRL 寄存器 .....	34
5.2.5	MISC_CTRL 寄存器 .....	35
5.2.6	GPIO0 配置寄存器(对应 GPIO[31:0]) .....	36
5.2.7	GPIO1 配置寄存器(对应 GPIO[63:32]) .....	36
5.2.8	GPIO2 配置寄存器(对应 GPIO[95:64]) .....	37
5.2.9	GPIO3 配置寄存器(对应 GPIO[127:96]) .....	37
5.2.10	PAD[31:0]引脚复用关系配置寄存器.....	37
5.2.11	PAD[63:32]引脚复用关系配置寄存器.....	38
5.2.12	PAD[95:64]引脚复用关系配置寄存器.....	39
5.2.13	PAD[127:96]引脚复用关系配置寄存器.....	39
5.3	中断配置寄存器.....	39
6	SDRAM 控制器.....	45
6.1	专用寄存器.....	45
6.2	配置说明.....	45
7	SRAM 控制器 .....	48
7.1	功能描述.....	48
7.2	配置寄存器.....	48
8	Camera 接口 .....	49
8.1	功能概述.....	49
8.2	接口协议.....	50
8.3	专用寄存器.....	52
8.4	配置操作.....	55
8.4.1	访存方式.....	55
8.4.2	配置顺序.....	56
9	I2S 控制器 .....	58
9.1	功能概述.....	58
9.2	接口协议.....	59
9.3	专用寄存器.....	59
9.4	配置操作.....	61
10	显示控制器(DC) .....	62
10.1	概述.....	62
10.2	寄存器定义和说明.....	62
10.2.1	帧缓冲配置寄存器 .....	62
10.2.2	帧缓冲地址寄存器 0 .....	62
10.2.3	帧缓冲地址寄存器 1 .....	62
10.2.4	帧缓冲跨度寄存器 .....	63
10.2.5	颜色抖动配置寄存器.....	63
10.2.6	颜色抖动查找表低位寄存器 .....	63
10.2.7	颜色抖动查找表高位寄存器 .....	63
10.2.8	颜色抖动说明.....	64
10.2.9	液晶面板配置寄存器.....	64
10.2.10	水平显示宽度寄存器.....	64
10.2.11	行同步配置寄存器 .....	64
10.2.12	垂直显示高度寄存器.....	65
10.2.13	场同步配置寄存器 .....	65
10.2.14	输出定序说明.....	65



10.2.15	伽马校正目录寄存器.....	65
10.2.16	伽马校正值寄存器.....	66
10.2.17	伽马校正说明.....	66
11	MAC 控制器.....	67
11.1	DMA 寄存器描述.....	67
11.1.1	Register0 (Bus Mode Register).....	67
11.1.2	Register1 (Transmit Poll Demand Register).....	68
11.1.3	Register2 (Receive Poll Demand Register).....	68
11.1.4	Register3 (Receive Descriptor List Address Register).....	69
11.1.5	Register4 (Transmit Descriptor List Address Register).....	69
11.1.6	Register5 (Status Register).....	69
11.1.7	Register6 (Operation Mode Register).....	72
11.1.8	Register7 (Interrupt Enable Register).....	74
11.1.9	Register8 (Missed Frame and Buffer Overflow Counter Register).....	76
11.1.10	Register9(Receive Interrupt Watchdog Timer Register).....	77
11.1.11	Register10 (AXI Bus Mode Register).....	77
11.1.12	Register11 (AXI Status Register).....	79
11.1.13	Register18 (Current Host Transmit Descriptor Register).....	79
11.1.14	Register19 (Current Host Receive Descriptor Register).....	79
11.1.15	Register20 (Current Host Transmit Buffer Address Register).....	80
11.1.16	Register21 (Current Host Receive Buffer Address Register).....	80
11.2	MAC 控制器寄存器描述.....	80
11.2.1	Register0 (MAC Configuration Register).....	80
11.2.2	Register1 (MAC Frame Filter).....	82
11.2.3	Register2 (Hash Table High Register).....	83
11.2.4	Register3 (Hash Table Low Register).....	84
11.2.5	Register4 (GMII Address Register).....	84
11.2.6	Register5 (GMII Data Register).....	85
11.2.7	Register6 (Flow Control Register).....	85
11.2.8	Register7 (VLAN Tag Register).....	86
11.2.9	Register8 (Version Register).....	86
11.2.10	Register14 (Interrupt Status Register).....	86
11.2.11	Register15 (Interrupt Mask Register).....	87
11.2.12	Register16 (MAC Address0 High Register).....	88
11.2.13	Register17 (MAC Address0 Low Register).....	88
11.2.14	Register18 (MAC Address1 High Register).....	89
11.2.15	Register19 (MAC Address1 Low Register).....	89
11.2.16	Register48 (AN Control Register).....	89
11.2.17	Register49 (AN Status Register).....	90
11.2.18	Register50 (Auto-Negotiation Advertisement Register).....	91
11.2.19	Register51 (Auto-Negotiation Link Partner Ability Register).....	92
11.2.20	Register52 (Auto-Negotiation Expansion Register).....	92
11.2.21	Register54 (SGMII/RGMII Status Register).....	93
11.2.22	Register448 (Time Stamp Control Register).....	93
11.2.23	Register449 (Sub-Second Increment Register).....	96
11.2.24	Register 450 (System Time - Seconds Register).....	96
11.2.25	Register 451 (System Time - Nanoseconds Register).....	96
11.2.26	Register 452 (System Time - Seconds Update Register).....	96
11.2.27	Register 453 (System Time - Nanoseconds Update Register).....	97
11.2.28	Register 454 (Time Stamp Addend Register).....	97
11.2.29	Register 455 (Target Time Seconds Register).....	97
11.2.30	Register 456 (Target Time Nanoseconds Register).....	97
11.2.31	Register 457 (System Time - Higher Word Seconds Register).....	97
11.2.32	Register 458 (Time Stamp Status Register).....	98
11.2.33	Register 459 (PPS Control Register).....	98
11.2.34	Register 460 (PPS Auxiliary Time Stamp - Nanoseconds Register).....	99
11.2.35	Register 461 (PPS Auxiliary Time Stamp - Seconds Register).....	99



11.3	DMA 描述符 .....	99
11.3.1	DMA 描述符的基本格式.....	99
11.3.2	DMA 接收描述符 .....	100
11.3.3	DMA 发送描述符 .....	103
11.4	软件编程向导.....	108
11.5	IEEE 1588 支持 .....	109
12	USB 控制器.....	112
12.1	总体概述.....	112
12.2	USB 主机控制器寄存器 .....	112
12.2.1	EHCI 相关寄存器.....	112
12.2.2	Capability 寄存器.....	113
12.2.3	Operational 寄存器.....	113
12.2.4	EHCI 实现相关寄存器.....	114
12.3	OHCI 相关寄存器 .....	116
12.3.1	Operational 寄存器 .....	116
12.3.2	OHCI 实现相关寄存器 .....	117
12.3.3	INSNREG06 寄存器 .....	117
12.3.4	INSNREG07 寄存器 .....	118
13	OTG 控制器 .....	119
13.1	概述.....	119
13.2	寄存器描述.....	119
13.2.1	全局控制与状态寄存器 (Global CSR Map).....	120
13.2.2	Host 模式控制与状态寄存器 (Host Mode CSR) .....	121
13.2.3	Device 模式控制与状态寄存器 (Device Mode CSR) .....	121
13.2.4	数据 FIFO 访问寄存器组 (DFIFO Access Register MAP) .....	122
13.2.5	功耗控制与门控时钟寄存器组 (Power and Clock Gating CSR Map) ..	123
13.3	寄存器描述.....	123
13.3.1	寄存器访问特性.....	123
13.3.2	全局寄存器 .....	123
13.3.3	Host 模式寄存器.....	148
13.3.4	Device 模式寄存器 .....	158
13.3.5	功耗与门控时钟寄存器 .....	174
14	通用 DMA.....	176
14.1	DMA 控制器结构描述 .....	176
14.2	DMA 寄存器 .....	176
14.2.1	ORDER_ADDR_IN .....	176
14.2.2	DMA_ORDER_ADDR.....	177
14.2.3	DMA_SADDR.....	177
14.2.4	DMA_DADDR .....	177
14.2.5	DMA_LENGTH.....	178
14.2.6	DMA_STEP_LENGTH .....	178
14.2.7	DMA_STEP_TIMES.....	178
14.2.8	DMA_CMD.....	178
14.3	多个模块使用 DMA 配置说明.....	179
15	SPI0 控制器 .....	181
15.1	SPI 控制器结构.....	181
15.2	配置寄存器.....	181
15.2.1	控制寄存器(SPCR).....	181
15.2.2	状态寄存器(SPSR).....	182
15.2.3	数据寄存器(TxFIFO/RxFIFO) .....	182
15.2.4	外部寄存器(SPER).....	182
15.2.5	参数控制寄存器(SFC_PARAM).....	183



15.2.6	片选控制寄存器(SFC_SOFTCS)	183
15.2.7	时序控制寄存器(SFC_TIMING)	183
15.3	接口时序	183
15.3.1	SPI 主控制器接口时序	183
15.3.2	SPI Flash 访问时序	184
15.4	使用指南	184
15.4.1	SPI 主控制器的读写操作	184
15.4.2	硬件 SPI Flash 读	185
15.4.3	混合访问 SPI Flash 和 SPI 主控制器	185
16	SPI1	187
16.1	SPI 主控制器结构	187
17	AC97	188
17.1	概述	188
17.2	AC97 控制器寄存器	188
17.2.1	CSR 寄存器	189
17.2.2	OCC 寄存器	189
17.2.3	ICC 寄存器	190
17.2.4	Codec 寄存器访问命令	190
17.2.5	中断状态寄存器/中断掩膜寄存器	191
17.2.6	中断状态/清除寄存器	191
17.2.7	OC 中断清除寄存器	191
17.2.8	IC 中断清除寄存器	191
17.2.9	CODEC WRITE 中断清除寄存器	192
17.2.10	CODEC READ 中断清除寄存器	192
18	I2C	193
18.1	概述	193
18.2	I2C 控制器结构	193
18.3	I2C 控制器寄存器说明	193
18.3.1	分频锁存器低字节寄存器 (PRERlo)	194
18.3.2	分频锁存器高字节寄存器 (PRERhi)	194
18.3.3	控制寄存器 (CTR)	194
18.3.4	发送数据寄存器 (TXR)	194
18.3.5	接受数据寄存器 (RXR)	195
18.3.6	命令控制寄存器 (CR)	195
18.3.7	状态寄存器 (SR)	195
19	UART 控制器	197
19.1	概述	197
19.2	控制器结构	197
19.3	寄存器描述	198
19.3.1	数据寄存器(DAT)	198
19.3.2	中断使能寄存器 (IER)	199
19.3.3	中断标识寄存器 (IIR)	199
19.3.4	FIFO 控制寄存器 (FCR)	200
19.3.5	线路控制寄存器 (LCR)	200
19.3.6	MODEM 控制寄存器 (MCR)	201
19.3.7	线路状态寄存器 (LSR)	201
19.3.8	MODEM 状态寄存器 (MSR)	202
19.3.9	分频锁存器	202
19.4	USART 寄存器描述	203
19.4.1	控制寄存器 (CR)	203
19.4.2	模式寄存器 (MR)	203



19.4.3	FI/DI 参数寄存器 (FIDIR) .....	203
20	NAND.....	204
20.1	NAND 控制器结构描述 .....	204
20.2	NAND 控制器寄存器配置描述 .....	204
20.2.1	NAND_CMD (地址: 0x1fe7_8000) .....	204
20.2.2	ADDR_C (地址: 0x1fe7_8004) .....	205
20.2.3	ADDR_R (地址: 0x1fe7_8008) .....	205
20.2.4	NAND_TIMING (地址: 0x1fe7_800C) .....	205
20.2.5	ID_L (地址: 0x1fe7_8010) .....	205
20.2.6	STATUS & ID_H (地址: 0x1fe7_8014) .....	205
20.2.7	NAND_PARAMETER (地址: 0x1fe7_8018) .....	205
20.2.8	NAND_OP_NUM (地址: 0x1fe7_801C) .....	206
20.2.9	CS_RDY_MAP (地址: 0x1fe7_8020) .....	206
20.2.10	DMA_ADDRESS (地址: 0x1fe7_8040) .....	207
20.3	NAND ADDR 说明 .....	207
20.4	NAND-FLASH 读写操作举例.....	209
20.5	NAND ECC 说明.....	209
20.6	NAND BOOT 说明 .....	210
21	RTC .....	212
21.1	概述.....	212
21.2	寄存器描述.....	212
21.2.1	寄存器地址列表.....	212
21.2.2	SYS_TOYWRITE0.....	213
21.2.3	SYS_TOYWRITE1.....	213
21.2.4	SYS_TOYMATCH0/1/2.....	214
21.2.5	SYS_RTCCTRL.....	214
21.2.6	SYS_RTCMATCH0/1/2.....	215
22	SDIO.....	216
22.1	功能概述.....	216
22.2	SDIO 协议概述 .....	216
22.3	专用寄存器.....	217
22.4	配置寄存器说明.....	225
23	CAN .....	228
23.1	概述.....	228
23.2	CAN 控制器结构 .....	228
23.3	标准模式.....	229
23.3.1	标准模式地址表.....	229
23.3.2	控制寄存器 (CR) .....	230
23.3.3	命令寄存器 (CMR) .....	231
23.3.4	状态寄存器 (SR) .....	231
23.3.5	中断寄存器 (IR) .....	231
23.3.6	验收代码寄存器 (ACR) .....	232
23.3.7	验收屏蔽寄存器 (AMR) .....	232
23.3.8	发送缓冲区列表.....	232
23.3.9	接收缓冲区列表.....	232
23.4	扩展模式.....	233
23.4.1	扩展模式地址表.....	233
23.4.2	模式寄存器 (MOD) .....	233
23.4.3	命令寄存器 (CMR) .....	234
23.4.4	状态寄存器 (SR) .....	234
23.4.5	中断寄存器 (IR) .....	235
23.4.6	中断使能寄存器 (IER) .....	235



23.4.7	仲裁丢失捕捉寄存器 (IER)	235
23.4.8	错误警报限制寄存器 (EMLR)	236
23.4.9	RX 错误计数寄存器 (RXERR)	236
23.4.10	TX 错误计数寄存器 (TXERR)	237
23.4.11	验收滤波器	237
23.4.12	RX 信息计数寄存器 (RMCR)	237
23.5	公共寄存器	237
23.5.1	总线定时寄存器 0 (BTR0)	237
23.5.2	总线定时寄存器 1 (BTR1)	237
23.5.3	输出控制寄存器 (OCR)	238
24	ADC 接口	239
24.1	功能概述	239
24.2	专用寄存器	239
24.3	配置操作	244
25	PWM	247
25.1	概述	247
25.2	PWM 寄存器说明	247
26	看门狗 (WDT)	250
26.1	概述	250
26.2	WATCH DOG 寄存器描述	250
26.2.1	WDT_EN 地址: (0x1fe7_c060)	250
26.2.2	WDT_SET (地址: 0x1fe7_c068)	250
26.2.3	WDT_timer (地址: 0x1fe7_c064)	251

## 图目录

图 1-1 龙芯 1C 结构图 .....	10
图 4-1 系统时钟结构 .....	30
图 9-1 显示控制器输出定序 .....	65
图 12-1 USB 主机控制器模块图 .....	112
图 15-1 SPI 控制器结构 .....	181
图 15-2 SPI 主控制器接口时序 .....	183
图 15-3 SPI Flash 标准读时序 .....	184
图 15-4 SPI Flash 快速读时序 .....	184
图 15-5 SPI Flash 双向 I/O 读时序 .....	184



## 表目录

表 2-1 信号类型代码 .....	16
表 4-1 龙芯 1C 时钟信号 .....	30
表 9-1 帧缓冲配置寄存器 .....	62
表 9-2 帧缓冲地址寄存器 0 .....	62
表 9-3 帧缓冲地址寄存器 1 .....	62
表 9-4 帧缓冲跨度寄存器 .....	63
表 9-6 颜色抖动配置寄存器 .....	63
表 9-7 颜色抖动查找表低位寄存器 .....	63
表 9-8 颜色抖动查找表高位寄存器 .....	63
表 9-9 液晶面板配置寄存器 .....	64
表 9-10 水平显示宽度寄存器 .....	64
表 9-11 行同步配置寄存器 .....	64
表 9-12 垂直显示高度寄存器 .....	65
表 9-13 场同步配置寄存器 .....	65
表 9-14 伽马校正目录寄存器 .....	65
表 9-15 伽马校正值寄存器 .....	66
表 15-1 SPI 配置寄存器列表 .....	181
表 15-2 SPI 控制寄存器(SPCR) .....	181
表 15-3 SPI 状态寄存器(SPSR).....	182
表 15-4 SPI 数据寄存器(TxFIFO/RXFIFO) .....	182
表 15-5 SPI 外部寄存器(SPER).....	182
表 15-6 SPI 分频系数 .....	182
表 15-7 SPI 参数控制寄存器(SFC_PARAM).....	183
表 15-8 SPI 片选控制寄存器(SFC_SOFTCS).....	183
表 15-9 SPI 时序控制寄存器(SFC_TIMING) .....	183



# 1 概述

龙芯 1C 芯片是基于 LS232 处理器核的高性价比单芯片系统，可应用于指纹生物识别、物联传感等领域。

1C 包含浮点处理单元，可以有效增强系统浮点数据处理能力。1C 的内存接口，支持多种类型的内存，允许灵活的系统设计。支持 8-bit SLC NAND 或 MLC NAND FLASH，提供高容量的存储扩展接口。

1C 为开发者提供了丰富的外设接口及片上模块，包括 Camera 控制器，USB OTG 2.0 及 USB HOST 2.0 接口，AC97/I2S 控制器，LCD 控制器，ADC 控制器，高速 SPI 接口，全功能 UART 接口等，提供足够的计算能力和多应用的连接能力。

## 1.1 体系结构框图

龙芯 1C 内部采用多级总线结构。处理器核、内存控制器、图形显示控制器、CAMERA 接口模块和 AXI\_MUX 使用交叉开关互连。OTG、MAC、USB、DMA 控制器、SPI 通过 AXI\_MUX 连接至交叉开关。低速外设（I2C、I2S、PWM、UART 等）通过 AXI2APB 连接至交叉开关。

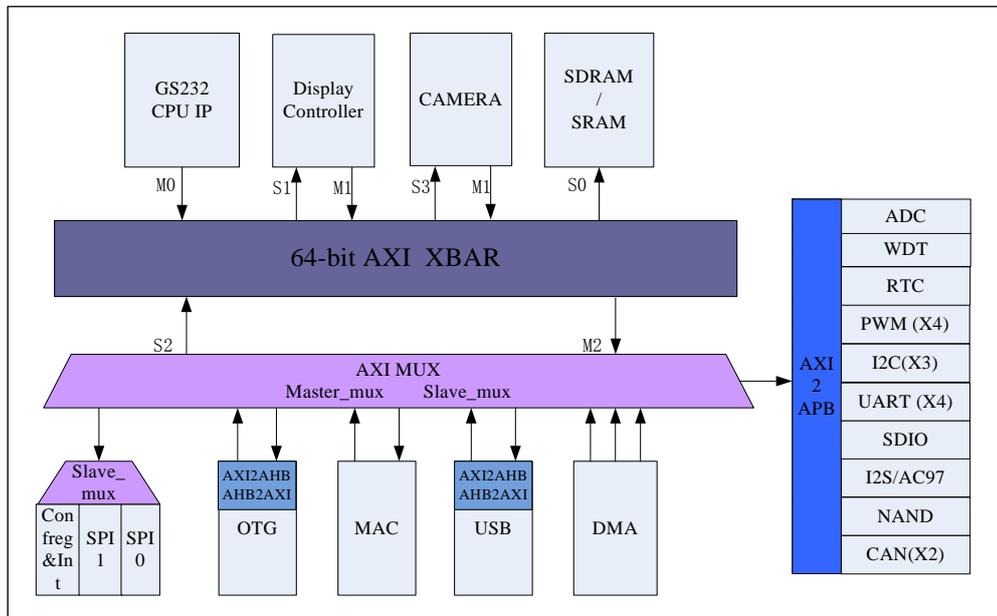


图 1-1 龙芯 1C 结构图

## 1.2 芯片主要功能

### 1.2.1 处理器核

- 单核心 LS232, MIPS32 指令集兼容, 主频 300MHZ
- 支持高效双发射 (一个时钟节拍执行两条指令) 技术
- 支持寄存器重命名、动态调度、转移预测等乱序发射、乱序执行技术
- 五级流水线 (取指、译码、发射、执行并写回、提交) 微体系构架
- 16KB 数据 cache 和 16KB 指令 cache
- 集成 64 位浮点处理部件, 支持全流水的 64 位浮点加法和浮点乘法运算, 硬件实现浮点除法运算

### 1.2.2 SDRAM 控制器

- SDRAM 接口, 工作频率 45~133MHz
- 支持 8/16 位并行数据总线宽度
- 支持自动刷新和自刷新功能, 支持页面模式

### 1.2.3 SRAM/NOR FLASH 接口

- SRAM 以及 NOR Flash 直连接口, 工作频率 66~133MHz
- 支持静态存储器片选引脚, 可以单独配置
- 支持 8bit/16bit 并行数据总线宽度
- 支持 SRAM 的 BANK 基地址及大小可编程

### 1.2.4 NAND 控制器

- 最大支持 H 单颗容量为 4GB NAND FLAS
- 支持 512 字节、2K 字节页及更大页面类型 FLASH
- 硬件 ECC 生成、检测和指示 (软件纠错)
- 支持 FLASH 数据读取速度 8~10MB/S, 写入速度 5MB/s
- 支持从 NAND Flash 启动
- 支持小尾端模式

### 1.2.5 时钟发生器

- 1 个标准 PLL 输入接口，支持外部无源晶体作为芯片时钟输入
- 支持片内输出可配置时钟一路，供片外外设使用
- PLL 频率软件可配置

### 1.2.6 I<sup>2</sup>S 接口

- 支持 master 模式下 I<sup>2</sup>S 输入
- 支持 master 模式下 I<sup>2</sup>S 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

### 1.2.7 AC97

- 可变采样率 AC97 编解码器接口（48KHz 及以下）
- 支持立体声 PCM 和单声道 MIC 输入
- 支持 2 通道立体声 PCM 输出
- 支持 DMA 和中断操作
- 支持 16、18 和 20 位采样精度，支持可变速率
- 支持 16 位、16 个入口 FIFO 每通道

### 1.2.8 LCD 控制器

- 支持 16/24 位像素模式
- 支持 RGB444/555/565/888 显示输出
- 支持 1024x768、800x600、640×480、320×240 分辨率
- 支持 DMA 传输模式

### 1.2.9 Camera 接口

- 支持 ITU-R BT.601/656 8 位输入
- 支持 RAW RGB、RGB565 及 YUV4:2:2 数据输入
- 支持 YUV、RGB888、RGB0888、RGB565 输出

- 支持 320x240 和 640x480 分辨率缩放
- 支持最大 2Kx2K 分辨率输入，分辨率可配置
- 支持 DMA 传输模式

#### 1.2.10 MAC 控制器

- 支持 10/100Mbps PHY 器件，包括 10 Base-T、100 Base-TX、100 Base-FX 和 100 Base-T4；
- 完全兼容 IEEE 标准 802.3
- 完全兼容 802.3x 全双工流控和半双工背压流控
- 支持 VLAN 帧
- 支持 DMA 传输模式
- 支持标准的媒体独立接口（MII）
- 支持标准的简化 MII 接口（RMII）可连接外部 PHY 芯片

#### 1.2.11 USB2.0 控制器

- 1 个 USB OTG 2.0 控制器
- 1 个 USB HOST 2.0 控制器
- 支持高速和全速模式
- 支持 DMA 传输模式
- 兼容 USB Rev 1.1 、USB Rev 2.0 协议

#### 1.2.12 SPI

- 支持两路独立 SPI 接口，每路 SPI 接口均支持 4 个片选
- 遵循串行外设接口(SPI)规范
- 支持同步、串行、全双工通信
- 支持 SPI 主、从模式
- 每次传输 8~16 位
- 支持查询、中断传输模式
- 支持 SPI nor flash 启动
- 支持 SPI 接口双向输入输出，最高数据传输速度 24~96 Mbps
- 支持最低速率通讯要求，速率达 25KB 以下，方便匹配特殊设备。

### 1.2.13 I<sup>2</sup>C 总线

- 三路标准 I<sup>2</sup>C 总线接口
- 支持主、从、或主/从模式配置
- 总线的时钟频率可编程

### 1.2.14 UART

- 支持一个全功串口，可复用为 4 个两线串口，支持智能卡协议
- 基于中断操作的 RxD0, TxD0, RxD1, TxD1, RxD2 和 TxD2;
- UART 通道 0, 1 和 2 带 IrDA 1.0
- UART 通道 0 和 1 带 RTS0, CTS0, RTS1 和 CTS1
- IrDA 接口速率可达 115200bps 波特率
- UART 接口速率可达 921.6Kbps 波特率

### 1.2.15 GPIO

- 最多支持 102 个 GPIO
- 所有 GPIO（启动和系统配置除外）在复位后默认为输入
- 所有 GPIO 支持中断功能
- 每个 GPIO 管脚均支持电平触发、边沿触发模式，可独立配置
- GPIO 管脚速率至少可达 4MHz

### 1.2.16 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

### 1.2.17 RTC

- 计时精确到 0.1 秒
- 可产生 3 个计时中断
- 支持外部无源晶体作为 RTC 时钟输入
- 支持外部电池供电运行，断电后由电池供电
- 专门的电源管脚，可以与电池或者 3.3V 主电源相连

- 提供秒、分、小时、日期（月）、月、年、星期和日期（年）

#### 1.2.18 CAN

- 2 路独立 CAN 控制器
- 兼容 CAN2.0A 和 CAN2.0B 协议（PCA82C200 兼容模式中的无源扩展帧）
- 支持 CAN 协议扩展
- 位速率可达 1Mbits/s

#### 1.2.19 SDIO

- 1 路独立 SDIO 控制器
- 兼容 SD 卡/MMC/SDIO 协议
- 支持 SDIO 启动

#### 1.2.20 ADC

- 采样率最高 1MHz
- 4 路 ADC 输入
- 支持 4 线和 5 线触摸屏
- 支持连续采样和单次采样
- 支持模拟看门狗

## 2 引脚定义

### 2.1 约定

本章对龙芯 1C 引脚定义的说明使用以下约定：

信号的输入输出类型由一个代码表示，见表 2-1。

表 2-1 信号类型代码

代码	描述
A	模拟
DIFF I/O	双向差分
DIFF IN	差分输入
DIFF OUT	差分输出
I	输入
I/O	双向
O	输出
OD	开漏输出
P	电源
G	地

### 2.2 LCD 接口

信号名称	类型	描述
LCD_CLK	O	LCD 时钟信号
LCD_HSYNC	O	LCD 水平同步信号
LCD_VSYNC	O	LCD 垂直同步信号
LCD_EN	O	LCD 使能信号
LCD_DAT[15:0]	O	LCD 数据信号

**【注】**在 QFP100 封装下，LCD 接口不能用。在 QFP176 封装下，LCD 可以用 16 位和 24 位模式。16 位模式下引脚不需要复用；在 24 位模式下，低位需要复用 CAM\_DAT[7:0]或者 MAC 信号。

### 2.3 SDRAM 接口

信号名称	类型	描述
SD_CLK	O	SDRAM 时钟信号
SD_CKE	O	SDRAM 时钟使能信号
SD_CS <sub>n</sub>	O	SDRAM 片选信号
SD_RAS <sub>n</sub>	O	SDRAM 行选通信号
SD_CAS <sub>n</sub>	O	SDRAM 列选通信号

SD_WE	O	SDRAM 读写信号
SD_BA[1:0]	O	SDRAM 的 Bank 信号，一共 4 个 Bank
SD_ADDR[12:0]	O	SDRAM 地址信号
SD_DATA[15:0]	I/O	SDRAM 数据信号
SD_DQM[1:0]	O	SDRAM 数据屏蔽信号

## 2.4 I2S 接口

信号名称	类型	描述
I2S_MCLK	O	I2S 时钟信号
I2S_BCLK	O	I2S 比特位时钟信号
I2S_LRCK	O	I2S 通道选择信号
I2S_DI	I	I2S 数据串行输入信号
I2S_DO	O	I2S 数据串行输出信号

【注】在 QFP100 封装下，I2S 接口没有引出来，需要和 MAC 引脚复用。在 QFP176 封装下，I2S 接口引出来了，可以直接使用。

## 2.5 I2C 接口

信号名称	类型	描述
I2C[2:0]_SCL	O	I <sup>2</sup> C 串行时钟
I2C[2:0]_SDA	I/O	I <sup>2</sup> C 串行数据

【注】在 QFP100 和 QFP176 封装下，I2C 接口都没有引出来，需要和 MAC、EJTAG、LCD 或 CAM 引脚复用。

## 2.6 UART 接口

信号名称	类型	描述
UART0_TX	O	UART0 数据发送
UART0_RX	I	UART0 数据接收
UART0_RTS	I	UART0 请求接收
UART0_CTS	I	UART0 允许接收
UART0_DSR	I	UART0 设备准备好
UART0_DTR	O	UART0 数据终端准备好
UART0_DCD	I	UART0 载波检测
UART0_RI	I	UART0 振铃提示

【注】在 QFP100 封装下，无全功能串口，只有两线式。在 QFP176D 封装下，全功能串口可直接使用。

## 2.7 PWM 接口

信号名称	类型	描述
PWM0	O	PWM0 脉冲输出
PWM1	O	PWM1 脉冲输出
PWM2	O	PWM2 脉冲输出
PWM3	O	PWM3 脉冲输出

【注】在 QFP100 封装下，PWM 没有引出来，需要和其它复用。在 QFP176 封装下，PWM0 和 PWM1 可直接使用，PWM2 和 PWM3 需要和其它复用。

## 2.8 ADC 接口

信号名称	类型	描述
ADC_REXT	I	ADC 参考电阻
ADC_VREF	I	ADC 参考电压
ADC_VDDA	I	ADC 模拟电源
ADC_VSSA	I	ADC 模拟地
ADC_D0	I	ADC 第 0 通道采样输入
ADC_D1	I	ADC 第 1 通道采样输入
ADC_XP	I	触摸屏 X+/ADC 第 2 通道采样输入
ADC_YP	I	触摸屏 Y+/ADC 第 3 通道采样输入

【注】AD 接口仅在 QFP176A 封装下可用。

## 2.9 SPI 接口

信号名称	类型	描述
SPI_SCK	O	SPI 时钟输出
SPI[1:0]_CSn	O	SPI 片选 0
SPI_MOSI	O	SPI 数据输出
SPI_MISO	I	SPI 数据输入

## 2.10 EJTAG 接口

信号名称	类型	描述
EJTAG_SEL	I	JTAG 选择(0: JTAG, 1: EJTAG)

JTAG_SEL	I	JTAG 引脚功能复用(为 1 选择复用功能)
EJTAG_TCK	I	JTAG 时钟
EJTAG_TDI	I	JTAG 数据输入
EJTAG_TMS	I	JTAG 模式
EJTAG_TRST	I	JTAG 复位, 需下拉
EJTAG_TDO	O	JTAG 数据输出

【注】EJTAG\_SEL 是选择 JTAG 还是 EJTAG, JTAG\_SEL 是用来选择 JTAG 复用功能, 不要混淆。

## 2.11 CAMERA 接口

信号名称	类型	描述
CAM_CLKOUT	O	摄像头参考时钟输出
CAM_PCLK_I	I	摄像头像素时钟输入
CAM_HSYNC	I	摄像头水平同步信号
CAM_VSYNC	I	摄像头垂直同步信号
CAMDATA[7:0]	I	摄像头数据输入

【注】在 QFP100 封装下, CAM 没有引出来, 需要和 NAND、MAC 复用。  
在 QFP176 封装下, CAM 可直接使用。

## 2.12 NAND 接口

信号名称	类型	描述
NAND_CLE	O	NAND 命令锁存
NAND_ALE	O	NAND 地址锁存
NAND_RD	O	NAND 读信号
NAND_WR	O	NAND 写信号
NAND_CE	O	NAND 片选 0
NAND_RDY	I	NAND 准备好 0
NAND_D[7:0]	I/O	NAND 地址/数据线

## 2.13 MAC 接口

信号名称	类型	描述
MAC[1:0]_TXCK	O	MII 发送时钟
MAC[1:0]_TXEN	O	MII 发送控制
MAC[1:0]_TXD[3:0]	O	MII 发送数据

MAC[1:0]_RXCK	I	MII 接收时钟
MAC[1:0]_RXDV	I	MII 接收控制
MAC[1:0]_RXD[3:0] ]	I	MII 接收数据
MAC[1:0]_MDCK	O	SMA 接口时钟
MAC[1:0]_MDIO	I/O	SMA 接口数据
MAC[1:0]_COL	I	MAC 冲突检测
MAC[1:0]_CRS	I	MAC 载波侦测

【注】在 QFP100 封装下，MAC 只能使用 RMII 模式。在 QFP176 封装下，可使用 MII 和 RMII 模式。

## 2.14 OTG 接口

信号名称	类型	描述
OTG_DVDD		OTG 数字电源
OTG_DVSS		OTG 数字地
OTG_VDD33		OTG 模拟电源
OTG_VSS33		OTG 模拟地
OTG_REXT		OTG 参考电阻
OTG_DM	DIFF I/O	OTG 差分信号线 D-
OTG_DP	DIFF I/O	OTG 差分信号线 D+
OTG_VBUS		OTG_VBUS
OTG_ID		OTG_ID

## 2.15 USB 接口

信号名称	类型	描述
USB_DVDD		USB 数字电源
USB_DVSS		USB 数字地
USB_VDD33		USB 模拟电源
USB_VSS33		USB 模拟地
USB_REXT		USB 参考电阻
USB_DP	DIFF I/O	USB 差分信号线 D+
USB_DM	DIFF I/O	USB 差分信号线 D-

【注】在 QFP100 封装下，USB HOST 不能使用。在 QFP176 封装下，可以使用。

## 2.16 RTC 接口

信号名称	类型	描述
RTC_CLK_I	I	RTC 晶振输入，接 32.768K 晶振
RTC_CLK_O	O	RTC 晶振输出
VR_VDDA		RTC 电源

## 2.17 时钟配置信号

信号名称	类型	描述
XTALI	I	系统时钟晶振输入，接 24M
XTALO	O	系统时钟晶振输出

## 2.18 电源地

信号名称	类型	描述
PLL_VDD33	P	Core PLL 模拟电源
PLL_VSS33	G	Core PLL 模拟地
PLL_VDD12	P	Core PLL 数字电源
PLL_VSS12	G	Core PLL 数字地
CORE_VDD	P	核电压电源
CORE_VSS	G	核电压地
IO_VDD	P	IO 电源

## 2.19 初始化信号

龙芯 1C 有四种启动方式，OTG、SPI FLASH、NAND FLASH、SDIO。复用功能引脚，通过在系统复位期间采样外部上下拉的值得到配置信息，供软件判定上电状态。

表 2-1 配置信号

引脚名称	信号名称	描述
NAND_D[3:0]	start_freq	硬件控制模式下 CPU PLL 倍频配置 频率计算公式为： $6*(4*NAND\_D[3:0] + 40)$

NAND_D[5:4]	boot_sel	<p>启动选择, 在 QFP100 和 QFP176 封装下不同 (在 QFP100 封装下, NAND 的控制信号没有封装成引脚)</p> <p>在 QFP100 封装下, boot_sel 为</p> <ul style="list-style-type: none"> <li>00: 表示从 OTG 启动</li> <li>01: 表示从 SPI flash 启动</li> <li>10: 表示从 NAND flash 启动 (复用 SDRAM 引脚)</li> <li>11: 表示从 NAND flash 启动 (复用 MAC 引脚)</li> </ul> <p>在 QFP176 封装下, boot_sel 为</p> <ul style="list-style-type: none"> <li>00: 表示从 OTG 启动</li> <li>01: 表示从 SPI flash 启动</li> <li>10: 表示从 NAND flash 启动</li> <li>11: 表示从 SDIO 启动</li> </ul>
NAND_D[7:6]	nand_type	<p>NAND boot 时配置 NAND flash 颗粒容量大小</p> <ul style="list-style-type: none"> <li>11: 表示容量大等于 2GB (2K 页)</li> <li>10: 表示容量为 1GB (2K 页)</li> <li>01: 表示容量为 512MB (512 Bytes 页)</li> <li>00: 表示容量为小等于 256MB (512 Bytes 页)</li> </ul>
NAND_CLE	rs_rd_cfg	<p>NAND boot 时是否采用 ECC, 仅在 boot_sel 选择 NAND flash 启动时有效</p> <ul style="list-style-type: none"> <li>为 0 时, 表示 NAND 非 ECC 启动</li> <li>为 1 时, 表示 NAND ECC 启动</li> </ul>
SPIO_CLK	usb_refclksel	<p>USB_HOST 和 USB_OTG 的时钟选择信号。</p> <ul style="list-style-type: none"> <li>为 1 时, 表示由内部的 PLL 提供时钟。</li> <li>为 0 时, 表示由外部的晶振提供时钟。</li> </ul>

## 3 寄存器定义

### 3.1 基本寄存器定义

寄存器名称	地址	读/写 (R/W)	功能描述	复位值
基本寄存器				
START_FREQ	0xbfe7_8030	R/W	PLL 频率配置及 SDRAM 分频系数	上电状态
CLK_DIV_PARAM	0xbfe7_8034	R/W	CPU/CAMERA/DC 分频系数	0x0
CBUS_CONFSIGNALS	0xbf0_0400	R/W		0x0
CBUS_SD_CTRL0	0xbf0_0410	R/W	SDRAM 参数配置寄存器	0x001438a3
CBUS_SD_CTRL1	0xbf0_0414	R/W	SDRAM 参数配置寄存器	0x80000080
SHUT_CTRL	0xbf0_0420	R/W	各模块开关寄存器	0x0
MISC_CTRL	0xbf0_0424	R/W	功能复用寄存器 0	0x00000004
中断寄存器				
INTISR0	0xbf0_0140	R	中断状态寄存器 0	0x0
INTEN0	0xbf0_0144	R/W	中断使能寄存器 0	0x0
INTSET0	0xbf0_0148	R/W	中断置位寄存器 0	0x0
INTCLR0	0xbf0_014c	R/W	中断清零寄存器 0	0x0
INTPOL0	0xbf0_0150	R/W	中断触发电平标志寄存器 0	0x0
INTEGE0	0xbf0_0154	R/W	中断触发边沿标志寄存器 0	0x0
INTISR1	0xbf0_0158	R	中断状态寄存器 1	0x0
INTEN1	0xbf0_015c	R/W	中断使能寄存器 1	0x0
INTSET1	0xbf0_0160	R/W	中断置位寄存器 1	0x0
INTCLR1	0xbf0_0164	R/W	中断清零寄存器 1	0x0
INTPOL1	0xbf0_0168	R/W	中断触发电平标志寄存器 1	0x0
INTEGE1	0xbf0_016c	R/W	中断触发边沿标志寄存器 1	0x0
INTISR2	0xbf0_0170	R	中断状态寄存器 2	0x0
INTEN2	0xbf0_0174	R/W	中断使能寄存器 2	0x0
INTSET2	0xbf0_0178	R/W	中断置位寄存器 2	0x0
INTCLR2	0xbf0_017c	R/W	中断清零寄存器 2	0x0
INTPOL2	0xbf0_0180	R/W	中断触发电平标志寄存器 2	0x0

INTEDGE2	0xbf0_0184	R/W	中断触发边沿标志寄存器 2	0x0
INTISR3	0xbf0_0188	R	中断状态寄存器 3	0x0
INTEN3	0xbf0_018c	R/W	中断使能寄存器 3	0x0
INTSET3	0xbf0_0190	R/W	中断置位寄存器 3	0x0
INTCLR3	0xbf0_0194	R/W	中断清零寄存器 3	0x0
INTPOL3	0xbf0_0198	R/W	中断触发电平标志寄存器 3	0x0
INTEDGE3	0xbf0_019c	R/W	中断触发边沿标志寄存器 3	0x0
INTISR4	0xbf0_01a0	R	中断状态寄存器 4	0x0
INTEN4	0xbf0_01a4	R/W	中断使能寄存器 4	0x0
INTSET4	0xbf0_01a8	R/W	中断置位寄存器 4	0x0
INTCLR4	0xbf0_01ac	R/W	中断清零寄存器 4	0x0
INTPOL4	0xbf0_01b0	R/W	中断触发电平标志寄存器 4	0x0
INTEDGE4	0xbf0_01b4	R/W	中断触发边沿标志寄存器 4	0x0
GPIO 寄存器				
GPIO_CFG0	0xbf0_10c0	R/W	GPIO 配置寄存器 0	0xffe00040
GPIO_CFG1	0xbf0_10c4	R/W	GPIO 配置寄存器 1	0xfc00_3f3f
GPIO_CFG2	0xbf0_10c8	R/W	GPIO 配置寄存器 2	0x1ff0_3fff
GPIO_CFG3	0xbf0_10cc	R/W	GPIO 配置寄存器 3	0x0
GPIO_OE0	0xbf0_10d0	R/W	GPIO 输出使能寄存器 0	
GPIO_OE1	0xbf0_10d4	R/W	GPIO 输出使能寄存器 1	0xfc00_3f3f
GPIO_OE2	0xbf0_10d8	R/W	GPIO 输出使能寄存器 2	0x1ff0_3fff
GPIO_OE3	0xbf0_10dc	R/W	GPIO 输出使能寄存器 3	0x0
GPIO_IN0	0xbf0_10e0	R/W	GPIO 输入寄存器 0	0x0
GPIO_IN1	0xbf0_10e4	R/W	GPIO 输入寄存器 1	0x0
GPIO_IN2	0xbf0_10e8	R/W	GPIO 输入寄存器 2	0x0
GPIO_IN3	0xbf0_10ec	R/W	GPIO 输入寄存器 3	0x0
GPIO_OUT0	0xbf0_10f0	R/W	GPIO 输出寄存器 0	0x0
GPIO_OUT1	0xbf0_10f4	R/W	GPIO 输出寄存器 1	0x0
GPIO_OUT2	0xbf0_10f8	R/W	GPIO 输出寄存器 2	0x0
GPIO_OUT3	0xbf0_10fc	R/W	GPIO 输出寄存器 3	0x0
复用关系寄存器				
CBUS_FIRST0	0xbf0_11c0	R/W	第一复用寄存器 0	0x0

CBUS_FIRST1	0xbf0_11c4	R/W	第一复用寄存器 1	0x0
CBUS_FIRST2	0xbf0_11c8	R/W	第一复用寄存器 2	0x0
CBUS_FIRST3	0xbf0_11cc	R/W	第一复用寄存器 3	0x0
CBUS_SECOND0	0xbf0_11d0	R/W	第二复用寄存器 0	0x0
CBUS_SECOND1	0xbf0_11d4	R/W	第二复用寄存器 1	0x0
CBUS_SECOND2	0xbf0_11d8	R/W	第二复用寄存器 2	0x0
CBUS_SECOND3	0xbf0_11dc	R/W	第二复用寄存器 3	0x0
CBUS_THIRD0	0xbf0_11e0	R/W	第三复用寄存器 0	0x0
CBUS_THIRD1	0xbf0_11e4	R/W	第三复用寄存器 1	0x0
CBUS_THIRD2	0xbf0_11e8	R/W	第三复用寄存器 2	0x0
CBUS_THIRD3	0xbf0_11ec	R/W	第三复用寄存器 3	0x0
CBUS_FOURTH0	0xbf0_11f0	R/W	第四复用寄存器 0	0x0
CBUS_FOURTH1	0xbf0_11f4	R/W	第四复用寄存器 1	0x0
CBUS_FOURTH2	0xbf0_11f8	R/W	第四复用寄存器 2	0x0
CBUS_FOURTH3	0xbf0_11fc	R/W	第四复用寄存器 3	0x0
DMA 配置寄存器				
ORDER_ADDR_IN	0xbf0_1160	R/W	DMA 配置寄存器	

### 3.2 各模块寄存器定义

寄存器名称	地址	读/写 (R/W)	功能描述	复位值
SDRAM 接口				
SD_CONFIG[31:0]	0xbf0_0410	R/W	SDRAM 参数配置寄存器	0x30d0db5a
SD_CONFIG[63:32]	0xbf0_0414	R/W	SDRAM 参数配置寄存器	0x50
PWM0				
CNTR	0x1fe5_c000	R/W	主计数器	0x0
HRC	0x1fe5_c004	R/W	高脉冲定时参考寄存器	0x0
LRC	0x1fe5_c008	R/W	低脉冲定时参考寄存器	0x0
CTRL	0x1fe5_c00c	R/W	控制寄存器	0x0
PWM1				
CNTR	0x1fe5_c010	R/W	主计数器	0x0
HRC	0x1fe5_c014	R/W	高脉冲定时参考寄存器	0x0

LRC	0x1fe5_c018	R/W	低脉冲定时参考寄存器	0x0
CTRL	0x1fe5_c01c	R/W	控制寄存器	0x0
PWM2				
CNTR	0x1fe5_c020	R/W	主计数器	0x0
HRC	0x1fe5_c024	R/W	高脉冲定时参考寄存器	0x0
LRC	0x1fe5_c028	R/W	低脉冲定时参考寄存器	0x0
CTRL	0x1fe5_c02c	R/W	控制寄存器	0x0
PWM3				
CNTR	0x1fe5_c030	R/W	主计数器	0x0
HRC	0x1fe5_c034	R/W	高脉冲定时参考寄存器	0x0
LRC	0x1fe5_c038	R/W	低脉冲定时参考寄存器	0x0
CTRL	0x1fe5_c03c	R/W	控制寄存器	0x0
I2S 接口				
IISVersion	0xbfe6_0000	R/W	I2S 标识寄存器	0x0
IISConfig	0xbfe6_0001	R/W	I2S 配置寄存器	0x0
IISState	0xbfe6_0002	R/W	I2S 状态寄存器	0x0
IISRxData	0xbfe6_0003	R/W	I2S 接收数据寄存器	0x0
IISTxData	0xbfe6_0004	R/W	I2S 发送数据寄存器	0x0
AC97				
CSR	0xbfe6_0000	R/W	配置状态寄存器	0x0
OCC0	0xbfe6_0004	R/W	输出通道配置寄存器 0	0x4141
OCC1	0xbfe6_0008		保留	
OCC2	0xbfe6_000c		保留	
ICC	0xbfe6_0010	R/W	输入通道配置寄存器	0x410000
CODEC_ID	0xbfe6_0014		Codec ID 寄存器	
CRAC	0xbfe6_0018		Codec 寄存器访问命令	0x0
OC0	0xbfe6_0020		输出声道 0	
OC1	0xbfe6_0024		输出声道 1	
OC2	0xbfe6_0028		保留	
OC3	0xbfe6_002c		保留	
OC4	0xbfe6_0030		保留	
OC5	0xbfe6_0034		保留	

OC6	0xbfe6_0038		保留	
OC7	0xbfe6_003c		保留	
OC10	0xbfe6_0040		保留	
IC0	0xbfe6_0044		保留	
IC1	0xbfe6_0048		保留	
IC2	0xbfe6_004c		输入声道 2	
INTRAW	0xbfe6_0054	R/W	中断状态寄存器	
INTM	0xbfe6_0058	R/W	中断掩膜	0x0
INT_CLR	0xbfe6_005c	R	中断状态/清除寄存器	0x0
INT_OC_CLR	0xbfe6_0060	R	OC 中断清除寄存器	0x0
INT_IC_CLR	0xbfe6_0064	R	IC 中断清除寄存器	0x0
INT_CW_CLR	0xbfe6_0610	R	CODEC 写中断清除寄存器	0x0
INT_CR_CLR	0xbfe6_006c	R	CODEC 读中断清除寄存器	0x0
<b>RTC</b>				
sys_toytrim	0x1fe6_4020	RW	对 32.768kHz 分频系数（计数器时钟）	
sys_toywrite0	0x1fe6_4024	W	TOY 低 32 位数值写入	
sys_toywrite1	0x1fe6_4028	W	TOY 高 32 位数值写入	
sys_toyread0	0x1fe6_402c	R	TOY 低 32 位数值读出	
sys_toyread1	0x1fe6_4030	R	TOY 高 32 位数值读出	
sys_toymatch0	0x1fe6_4034	RW	TOY 定时中断 0	
sys_toymatch1	0x1fe6_4038	RW	TOY 定时中断 1	
sys_toymatch2	0x1fe6_403c	RW	TOY 定时中断 2	
sys_rtcctrl	0x1fe6_4040	RW	TOY 和 RTC 控制寄存器	
sys_rtctrim	0x1fe6_4060	RW	对 32.768kHz 的分频系数（定时器时钟）	
sys_rtcwrite0	0x1fe6_4064	W	RTC 定时计数写入	
sys_rtcread0	0x1fe6_4068	R	RTC 定时计数读出	
sys_rtcmatch0	0x1fe6_406c	RW	RTC 时钟定时中断 0	
sys_rtcmatch1	0x1fe6_4070	RW	RTC 时钟定时中断 1	
sys_rtcmatch2	0x1fe6_4074	RW	RTC 时钟定时中断 2	
<b>SDIO 接口</b>				
sdi_con	0xbfe6_c000	R/W	SDIO 控制寄存器	0x0
sdi_pre	0xbfe6_c004	R/W	SDIO 预分频寄存器	0x1

sdi_cmd_arg	0xbf6_c008	R/W	SDIO 命令参数寄存器	0x0
sdi_cmd_con	0xbf6_c00c	R/W	SDIO 命令控制寄存器	0x0
sdi_cmd_sta	0xbf6_c010	R	SDIO 命令状态寄存器	0x0
sdi_rsp0	0xbf6_c014	R/W	SDIO 响应寄存器 0	0x0
sdi_rsp1	0xbf6_c018	R/W	SDIO 响应寄存器 1	0x0
sdi_rsp2	0xbf6_c01c	R/W	SDIO 响应寄存器 2	0x0
sdi_rsp3	0xbf6_c020	R/W	SDIO 响应寄存器 3	0x0
sdi_dtimer	0xbf6_c024	R/W	SDIO 定时寄存器	0x1388
sdi_bsize	0xbf6_c028	R/W	SDIO 块大小寄存器	0x0
sdi_dat_con	0xbf6_c02c	R/W	SDIO 数据控制寄存器	0x0
sdi_dat_cnt	0xbf6_c030	R/W	SDIO 数据计数器	0x0
sdi_dat_sta	0xbf6_c034	R	SDIO 数据状态寄存器	0x0
sdi_fsta	0xbf6_c038	R	SDIO FIFO 状态寄存器	0x0
sdi_int_mask	0xbf6_c03c	R/W	SDIO 中断屏蔽寄存器	0x0
sdi_wdat	0xbf6_c040	R/W	SDIO 附加寄存器	0x0
sdi_sta_add0	0xbf6_c044	R/W	SDIO 附加寄存器 0	0x0
sdi_sta_add1	0xbf6_c048	R/W	SDIO 附加寄存器 1	0x0
sdi_sta_add2	0xbf6_c04c	R/W	SDIO 附加寄存器 2	0x0
sdi_sta_add3	0xbf6_c050	R/W	SDIO 附加寄存器 3	0x0
sdi_sta_add4	0xbf6_c054	R/W	SDIO 附加寄存器 4	0x0
sdi_sta_add5	0xbf6_c058	R/W	SDIO 附加寄存器 5	0x0
sdi_sta_add6	0xbf6_c05c	R/W	SDIO 附加寄存器 6	0x0
sdi_sta_add7	0xbf6_c060	R/W	SDIO 附加寄存器 7	0x0
ADC 接口				
adc_cnt	0xbf7_4000	R/W	分频和采样间隔寄存器	0x400010
adc_s_ctrl	0xbf7_4004	R/W	单次采样控制寄存器	0x0
adc_c_ctrl	0xbf7_4008	R/W	连续采样控制寄存器	0x0
x_range	0xbf7_4010	R/W	触摸屏 X 方向阈值	0x3ff0000
y_range	0xbf7_4014	R/W	触摸屏 Y 方向阈值	0x3ff0000
awatchdog_range	0xbf7_4018	R/W	模拟看门狗阈值	0x3f0000f
axis	0xbf7_401c	R/W	触摸屏坐标	0x0
adc_s_dout0	0xbf7_4020	R/W	通道 0 和通道 1 单次采样的结果	0x0

adc_s_dout1	0xbf7_4024	R/W	通道 2 和通道 3 单次采样的结果	0x0
adc_c_dout	0xbf7_4028	R/W	连续采样的结果	0x0
adc_debounce_cnt	0xbf7_402c	R/W	触摸屏去抖动时间	0x640
adc_int	0xbf7_4030	R/W	ADC 中断标志和控制	0x0
NAND				
NAND_CMD	0X1fe7_8000	R/W		0x0
ADDR_C	0X1fe7_8004	R/W		0x0
ADDR_R	0X1fe7_8008	R/W		0x0
NAND_TIMING	0X1fe7_800C	R/W		0x0412
ID_L	0X1fe7_8010	R		0x0
STATUS & ID_H	0X1fe7_8014	R		0x0
NAND_PARAMETER	0X1fe7_8018	R/W		0x
NAND_OP_NUM	0X1fe7_801C	R/W		0x800
CS_RDY_MAP	0X1fe7_8020	R/W		0x0
DMA access address	0X1fe7_8040	R/W		0x0
CAMERA 接口				
DMA_ADDR0_CONFIG	0x1c28_0000	R/W	帧缓冲区基地址 0	0x0
DMA_ADDR1_CONFIG	0x1c28_0008	R/W	帧缓冲区基地址 1	0x0
DMA_ADDR2_CONFIG	0x1c28_0010	R/W	帧缓冲区基地址 2	0x0
DMA_ADDR3_CONFIG	0x1c28_0018	R/W	帧缓冲区基地址 3	0x0
Camif_config_pix	0x1c28_0020	R/W	浮动分辨率像素	0x0
Camif_config_uoffset	0x1c28_0028	R/W	U 基地址	0x0
Camif_config_voffset	0x1c28_0030	R/W	V 基地址	0x0
Camif_config	0x1c28_0038	R/W	状态寄存器	0x0
I2S 接口				
IISVersion	0xbf7_0000	R/W	I2S 标识寄存器	0x0
IISConfig	0xbf7_0001	R/W	I2S 配置寄存器	0x0
IISState	0xbf7_0002	R/W	I2S 状态寄存器	0x0
IISRxData	0xbf7_0003	R/W	I2S 接收数据寄存器	0x0
IISTxData	0xbf7_0004	R/W	I2S 发送数据寄存器	0x0

## 4 时钟结构

### 4.1 时钟架构

龙芯 1C 的时钟结构如下图所示，来自 XTALI/O 的时钟送到 2 个 USBPHY 和 1 个 PLL。PLL 经过分频产生 CPU、SDRAM、MAC、USB 等主要模块的工作时钟；USB、MAC、OTG 等模块在接口处使用各自的时钟（或参考时钟）。

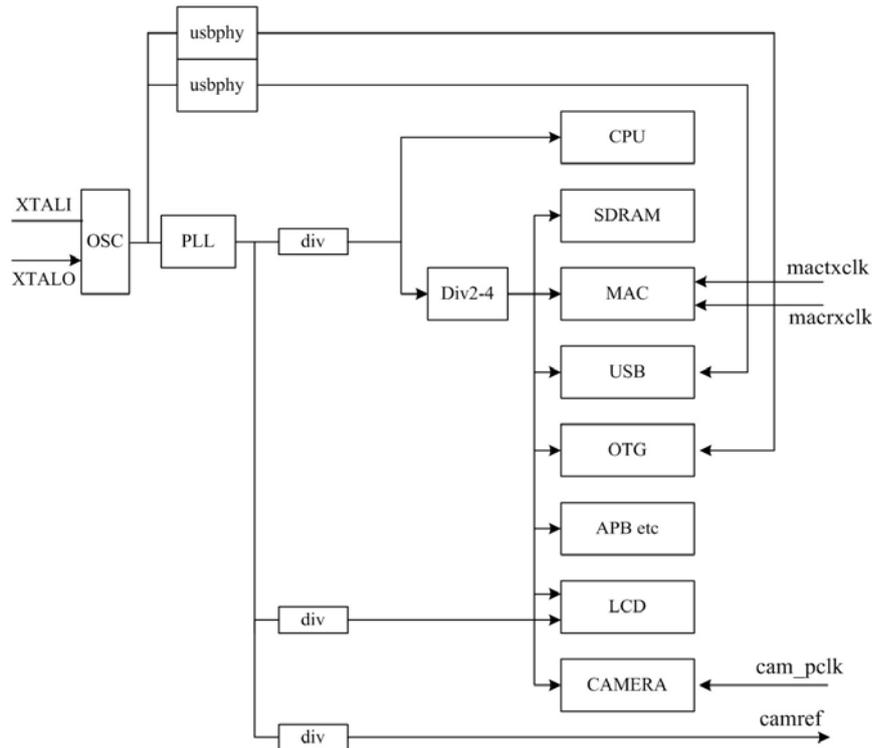


图 4-1 系统时钟结构

### 4.2 时钟信号说明

表 4-1 统一介绍了龙芯 1C 的所有时钟引脚。

表 4-1 龙芯 1C 时钟信号

信号名称	频率(MHz)	类型	描述
XTALI XTALO		I/O	PLL 和 USBPHY 参考时钟，连接 24MHz 晶体。
MACTXCLK		I	MAC 发送参考时钟
MACRXCLK		I	MAC 接收参考时钟
CAM_PCLK		I	Camera 像素时钟

## 5 芯片配置与控制

### 5.1 地址空间分配

龙芯 1C 的地址空间分为三级，包括一级 AXI 交叉开关上模块的地址空间、AXI\_MUX 下各模块的地址空间、以及 APB 上各模块的地址空间。

表 5-1 AXI 各模块地址分配

地址空间	设备	说明
0x0000_0000 – 0x0fff_ffff	SDRAM	256MB
0x1c28_0000 – 0x1c2f_ffff	CAMERA_IF	512K
0x1c30_0000 – 0x1c3f_ffff	DC	1MB
0x1f00_0000 – 0x1fff_ffff	AXI MUX	16MB

表 5-2 AXI-MUX 各模块地址分配

地址空间	设备	说明
0x1d00,0000 – 0x1d7f,ffff	SPI0-memory	8MB
0x1e00,0000 – 0x1e3f,ffff	SPI1-memory	4MB
0x1fc0_0000 – 0x1fcf_ffff	Boot	1MB，根据系统启动方式映射到 SPI 或 NAND
0x1fd0_0000 – 0x1fdf_ffff	CONFREG	1MB
0x1fe0_0000 – 0x1fe0_ffff	OTG	64KB
0x1fe1_0000 – 0x1fe1_ffff	MAC	64KB
0x1fe2_0000 – 0x1fe2_ffff	USB	64KB
0x1fe4_0000 – 0x1fe7_ffff	APB-devices	256KB
0x1fe8,0000 – 0x1feb,ffff	SPI0-IO	256KB
0x1fec,0000 – 0x1fef,ffff	SPI1-IO	256KB

表 5-3 APB 各模块地址分配

地址空间	模块	说明
0x1fe4_0000-0x1fe4_3fff	UART0	16KB
0x1fe4_4000-0x1fe4_7fff	UART1	16KB
0x1fe4_8000-0x1fe4_bfff	UART2	16KB

0x1fe4_c000-0x1fe4_ffff	UART3	16KB
0x1fe5_0000-0x1fe5_3fff	CAN0	16KB
0x1fe5_4000-0x1fe5_7fff	CAN1	16KB
0x1fe5_8000-0x1fe5_bfff	I2C-0	16KB
0x1fe5_c000-0x1fe5_ffff	PWM	16KB
0x1fe6_0000-0x1fe6_3fff	AC97/I2S	16KB
0x1fe6_4000-0x1fe6_7fff	RTC	16KB
0x1fe6_8000-0x1fe6_bfff	I2C-1	16KB
0x1fe6_c000-0x1fe6_ffff	SDIO	16KB
0x1fe7_0000-0x1fe7_3fff	I2C-2	16KB
0x1fe7_4000-0x1fe7_7fff	ADC	16KB
0x1fe7_8000-0x1fe7_bfff	NAND	16KB
0x1fe7_c000-0x1fe7_ffff	BOOT_ROM	16KB
	WDT	

注：ac97 和 i2s 引脚和物理地址都是复用的，通过软件配置选择是 ac97 还是 i2s。

## 5.2 芯片配置寄存器（CONFREG）

龙芯 1C 芯片内部专门有一个配置寄存器模块（CONFREG），用于芯片的一些基本配置及复用关系等。以下是寄存器列表及说明。

### 5.2.1 PLL/SDRAM 频率配置寄存器

寄存器名称	地址	读/写(R/W)	功能描述	复位值
START_FREQ	0xbfe7_8030	R/W	PLL 配置频率及 SDRAM 分频系数	

START_FREQ	位	缺省值	描述
PLL_VALID	31	0	PLL 倍频系数有效位
Reserved	30:24	0	
FRAC_N	23:16	0	PLL 倍频系数的小数部分

M_PLL	15:8		PLL 倍频系数的整数部分 (理论可以达到 255, 建议不要超过 100)
Reserved	7:4	0	
RST_TIME	3:2		配置完 PLL 频率后复位时间 00: 表示 no reset 01: 表示 0x100 CLK 10: 表示 0x400 CLK 11: 表示 0xff0 CLK
SDRAM_DIV	1:0		CPU 到 SDRAM 的分频系数 00: 表示 2 分频 01: 表示 4 分频 10 或 11: 表示 3 分频

注: PLL 的分频系数 N 固定为 4, PLL 的频率计算公式如下:

$$\text{Freq\_PLL} = \text{XIN} * (\text{M\_PLL} + \text{FRAC\_N}) / 4$$

### 5.2.2 CPU/CAMERA/DC 频率配置寄存器

寄存器名称	地址	读/写(R/W)	功能描述	复位值
CLK_DIV_PARAM	0xbfef_8034	R/W	CPU/CAMERA/DC 分频系数	

CLK_DIV_PARAM	位	缺省值	描述
PIX_DIV	31:24	0x24	DC 像素时钟分频系数 PIX_DIV[7]为配置参数有效位 PIX_DIV[6:0]为分频系数, 为 0 时不分频, 其它范围为 1~127
CAM_DIV	23:16	0x24	CAMERA 时钟分频系数 CAM_DIV[7]为配置参数有效位 CAM_DIV[6:0]为分频系数, 为 0 时不分频, 其它范围为 1~127
CPU_DIV	15:8	0x2	CPU 时钟分频系数 CPU_DIV[7]为配置参数有效位 CPU_DIV[6:0]为分频系数, 为 0 时不分频, 其它范围为 1~127
Reserved	7:6	0	
PIX_DIV_VALID	5	0	DC 像素时钟分频系数有效
PIX_SEL	4	0	DC 像素时钟选择信号 为 1 时选择分频时钟 为 0 时选择晶振输入时钟(bypass 模式)
CAM_DIV_VALID	3	0	CAMERA 时钟分频系数有效

CAM_SEL	2	0	CAMERA 时钟选择信号 为 1 时选择分频时钟 为 0 时选择晶振输入时钟(bypass 模式)
CAM_DIV_VALID	1	0	CPU 时钟分频系数有效
CAM_SEL	0	0	CPU 时钟选择信号 为 1 时选择分频时钟 为 0 时选择晶振输入时钟(bypass 模式)

$\text{Freq\_PIX} = \text{Freq\_PLL}/\text{PIX\_DIV};$

$\text{Freq\_CAM} = \text{Freq\_PLL}/\text{CAM\_DIV};$

$\text{Freq\_CPU} = \text{Freq\_PLL}/\text{CPU\_DIV};$

$\text{Freq\_SDRAM} = \text{Freq\_CPU}/\text{SDRAM\_DIV};$

### 5.2.3 SDRAM 参数寄存器

寄存器名称	地址	读/写(R/W)	功能描述	复位值
SD_CONFIG[31:0]	0xbf0_0410	R/W	SDRAM 参数配置寄存器	0x001438a3
SD_CONFIG[63:32]	0xbf0_0414	R/W	SDRAM 参数配置寄存器	0x80000080

SDRAM 参数寄存器的详细说明和配置见第七章。

### 5.2.4 SHUT\_CTRL 寄存器

寄存器名称	地址	读/写(R/W)	功能描述	复位值
shut_ctrl	0xbf0_0420	R/W	各模块开关寄存器	0x0

SHUT_CTRL	位	缺省值	描述
UART_split	31:30	0	用于智能卡??
OUTPUT_CLK	29:26	0	时钟输出配置。 每一位表示一个时钟的输出 第 0 位为 1 时表示 DC 的 PIX_CLK 输出; 第 1 位为 1 时表示 CAMERA 时钟从 SPI0_CS 引脚输出; 第 2 位为 1 时表示 CAMERA 时钟从 PWM0 引脚输出; 第 3 位为 1 时表示晶振时钟输出

ADC_shut	25	0	ADC 模块关闭，为 1 时有效
SDIO_shut	24	0	SDIO 模块关闭，为 1 时有效
DMA2_shut	23	0	DMA2 模块关闭，为 1 时有效
DMA1_shut	22	0	DMA1 模块关闭，为 1 时有效
DMA0_shut	21	0	DMA0 模块关闭，为 1 时有效
SPI1_shut	20	0	SPI1 模块关闭，为 1 时有效
SPI0_shut	19	0	SPI0 模块关闭，为 1 时有效
I2C2_shut	18	0	I2C2 模块关闭，为 1 时有效
I2C1_shut	17	0	I2C1 模块关闭，为 1 时有效
I2C0_shut	16	0	I2C0 模块关闭，为 1 时有效
AC97_shut	15	0	AC97 模块关闭，为 1 时有效
I2S_shut	14	0	I2S 模块关闭，为 1 时有效
UART3_shut	13	0	UART3 模块关闭，为 1 时有效
UART2_shut	12	0	UART2 模块关闭，为 1 时有效
UART1_shut	11	0	UART1 模块关闭，为 1 时有效
UART0_shut	10	0	UART0 模块关闭，为 1 时有效
CAN1_shut	9	0	CAN1 模块关闭，为 1 时有效
CAN0_shut	8	0	CAN0 模块关闭，为 1 时有效
ECC_shut	7	0	NAND ECC 功能关闭，为 1 时有效
MAC_shut	6	0	MAC 模块关闭，为 1 时有效
USBHOST_shut	5	0	USBHOST 模块关闭，为 1 时有效
USBOTG_shut	4	0	USBOTG 模块关闭，为 1 时有效
SDRAM_shut	3	0	SDRAM 模块关闭，为 1 时有效
SRAM_shut	2	0	SRAM 模块关闭，为 1 时有效
CAM_shut	1	0	CAMERA 模块关闭，为 1 时有效
LCD_shut	0	0	LCD 模块关闭，为 1 时有效

### 5.2.5 MISC\_CTRL 寄存器

寄存器名称	地址	读/写(R/W)	功能描述	复位值
misc_ctrl	0xbf0_0424	R/W	功能复用寄存器	

MISC_CTRL	位	缺省值	描述
USBHOST_RSTn	31	0	USBHOST 模块软件复位
PHY_INTF_SEL_i	30:28	0	MAC 模块 MII 接口配置信号 000: MII 模式 100: RMI 模式
Reserved	27:26	0	
AC97_EN	25	0	AC97 和 I2S 复用选择, 为 1 时使用 AC97, 为 0 时使用 I2S
SDIO_DMA_EN	24:23	0	SDIO 使用 DMA, 为 0 时不使用 DMA, 为 1 时使用 DMA0, 为 2 时使用 DMA1, 为 3 时使用 DMA2
ADC_DMA_EN	22	0	ADC 使用 DMA, 为 1 时有效
Reserved	21:18	0	
SDIO_USE_SPI1	17	0	SDIO 使用 SPI1 引脚, 为 1 时有效
SDIO_USE_SPI0	16	0	SDIO 使用 SPI0 引脚, 为 1 时有效
SRAM_CTRL	15:0	0	SRAM 配置参数, 详细见 SRAM

### 5.2.6 GPIO0 配置寄存器(对应 GPIO[31:0])

寄存器名称	地址	读/写(R/W)	功能描述	复位值
GPIO_CFG0	0xbf0_10c0	R/W	GPIO0 配置寄存器	0xfc0_3f3f
GPIO_EN0	0xbf0_10d0	R/W	GPIO0 输入使能寄存器	0xfc0_3f3f
GPIO_IN0	0xbf0_10e0	R/W	GPIO0 输入寄存器	0x0
GPIO_OUT0	0xbf0_10f0	R/W	GPIO0 输出寄存器	0x0

如果要配置 GPIO[16]输出 1, 则需要先配置 GPIO\_CFG[16]为 1, 然后配置 GPIO\_EN[16]为 0, 然后配置 GPIO\_OUT[16]为 1。同理可配置 GPIO0~GPIO127 输入或者输出。

### 5.2.7 GPIO1 配置寄存器(对应 GPIO[63:32])

寄存器名称	地址	读/写(R/W)	功能描述	复位值
GPIO_CFG1	0xbf0_10c4	R/W	GPIO1 配置	0x0

			寄存器	
GPIO_EN1	0xbf0_10d4	R/W	GPIO1 输入使能寄存器	0x0
GPIO_IN1	0xbf0_10e4	R/W	GPIO1 输入寄存器	0x0
GPIO_OUT1	0xbf0_10f4	R/W	GPIO1 输出寄存器	0x0

### 5.2.8 GPIO2 配置寄存器(对应 GPIO[95:64])

寄存器名称	地址	读/写(R/W)	功能描述	复位值
GPIO_CFG2	0xbf0_10c8	R/W	GPIO2 配置寄存器	0x0
GPIO_EN2	0xbf0_10d8	R/W	GPIO2 输入使能寄存器	0x0
GPIO_IN2	0xbf0_10e8	R/W	GPIO2 输入寄存器	0x0
GPIO_OUT2	0xbf0_10f8	R/W	GPIO2 输出寄存器	0x0

### 5.2.9 GPIO3 配置寄存器(对应 GPIO[127:96])

寄存器名称	地址	读/写(R/W)	功能描述	复位值
GPIO_CFG3	0xbf0_10cc	R/W	GPIO3 配置寄存器	0x0
GPIO_EN3	0xbf0_10dc	R/W	GPIO3 输入使能寄存器	0x0
GPIO_IN3	0xbf0_10ec	R/W	GPIO3 输入寄存器	0x0
GPIO_OUT3	0xbf0_10fc	R/W	GPIO3 输出寄存器	0x0

### 5.2.10 PAD[31:0]引脚复用关系配置寄存器

寄存器名称	地址	读/写(R/W)	功能描述	复位值
CBUS_FIRST0	0xbf0_11c0	R/W	PAD[31:0] 第	0x0

			一复用寄存器	
CBUS_SECOND0	0xbf0d_11d0	R/W	PAD[31:0] 第二复用寄存器	0x0
CBUS_THIRD0	0xbf0d_11e0	R/W	PAD[31:0] 第三复用寄存器	0x0
CBUS_FOURTH0	0xbf0d_11f0	R/W	PAD[31:0] 第四复用寄存器	0x0

1C 芯片的引脚有多种复用关系，最多有五种功能和一种 GPIO 复用关系，具体见数据手册第七章复用关系表格。每个引脚的复用关系优先级如下：

第一复用>第二复用>第三复用>第四复用>GPIO 复用>默认功能

比如要使用 PAD[21]的第四复用功能，则需要将 CBUS\_FIRST0[21]~CBUS\_THIRD0[21]都置为 0，将 CBUS\_FOURTH0[21]置为 1。如果要使用默认功能，则需要将 CBUS\_FIRST0[21]~CBUS\_FOURTH0[21]和 GPIO[21]都置为 0。

#### 5.2.11 PAD[63:32]引脚复用关系配置寄存器

寄存器名称	地址	读/写(R/W)	功能描述	复位值
CBUS_FIRST1	0xbf0d_11c4	R/W	PAD[63:32] 第一复用寄存器	0x0
CBUS_SECOND1	0xbf0d_11d4	R/W	PAD[63:32] 第二复用寄存器	0x0
CBUS_THIRD1	0xbf0d_11e4	R/W	PAD[63:32] 第三复用寄存器	0x0
CBUS_FOURTH1	0xbf0d_11f4	R/W	PAD[63:32] 第四复用寄存器	0x0

### 5.2.12 PAD[95:64]引脚复用关系配置寄存器

寄存器名称	地址	读/写(R/W)	功能描述	复位值
CBUS_FIRST2	0xbf0_11c8	R/W	PAD[95:64] 第一复用寄存器	0x0
CBUS_SECOND2	0xbf0_11d8	R/W	PAD[95:64] 第二复用寄存器	0x0
CBUS_THIRD2	0xbf0_11e8	R/W	PAD[95:64] 第三复用寄存器	0x0
CBUS_FOURTH2	0xbf0_11f8	R/W	PAD[95:64] 第四复用寄存器	0x0

### 5.2.13 PAD[127:96]引脚复用关系配置寄存器

寄存器名称	地址	读/写(R/W)	功能描述	复位值
CBUS_FIRST3	0xbf0_11cc	R/W	PAD[127:96] 第一复用寄存器	0x0
CBUS_SECOND3	0xbf0_11dc	R/W	PAD[127:96] 第二复用寄存器	0x0
CBUS_THIRD3	0xbf0_11ec	R/W	PAD[127:96] 第三复用寄存器	0x0
CBUS_FOURTH3	0xbf0_11fc	R/W	PAD[127:96] 第四复用寄存器	0x0

## 5.3 中断配置寄存器

中断寄存器一共有五组，对应 CPU 的 CAUSE 寄存器里的五位中断位。每一组都有状态寄存器、使能寄存器、置位寄存器、清零寄存器、电平选择寄存器和边沿选择寄存器五个寄存器。这五个寄存器每一位都一一对应。所有的

GPIO[104:0]都可以当作中断状态寄存器，特殊中断寄存器分布在 INT0\_SR 和 INT1\_SR。

第一组中断寄存器

寄存器名称	地址	读/写(R/W)	功能描述	复位值
INT0_SR	0xbf0_1040	R	中断状态寄存器 0	0x0
INT0_EN	0xbf0_1044	R/W	中断使能寄存器 0	0x0
INT0_SET	0xbf0_1048	R/W	中断置位寄存器 0	0x0
INT0_CLR	0xbf0_104C	R/W	中断清 0 寄存器 0	0x0
INT0_POL	0xbf0_1050	R/W	中断电平选择寄存器 0	0x0
INT0_EDGE	0xbf0_1054	R/W	中断边沿选择寄存器 0	0x0

寄存器名称	地址	读/写(R/W)	功能描述	复位值
INT0_SR	0xbf0_1040	R	中断状态寄存器 0	0x0

INT0_SR	位	缺省值	描述
NAND_int	31	0	NAND 中断状态位
SDIO_int	30	0	SDIO 中断状态位
UART3_int	29	0	UART3 中断状态位
TOY_TICK	28	0	
RTC_TICK	27	0	
TOY_int	26:24	0	
RTC_int	23:21	0	
PWM_int	20:17	0	对应四路 PWM 中断状态位
Reversed	16	0	
DMA_int	15:13	0	对应三路 DMA 中断状态位

Reversed	12:11	0	
I2S_AC97_int	10	0	I2S 或者 AC97 中断状态位, ac97_en 为 1 时此位为 AC97 中断状态位, 反之则为 I2S 中断状态位
SPI1_int	9	0	SPI1 中断状态位
SPI0_int	8	0	SPI0 中断状态位
CAN1_int	7	0	CAN1 中断状态位
CAN0_int	6	0	CAN0 中断状态位
UART2_int	5	0	UART2 中断状态位
UART1_int	4	0	UART1 中断状态位
UART0_int	3:0	0	

第二组中断寄存器

寄存器名称	地址	读/写(R/W)	功能描述	复位值
INT1_SR	0xbf0_1058	R	中断状态寄存器 1	0x0
INT1_EN	0xbf0_105c	R/W	中断使能寄存器 1	0x0
INT1_SET	0xbf0_1060	R/W	中断置位寄存器 1	0x0
INT1_CLR	0xbf0_1064	R/W	中断清 0 寄存器 1	0x0
INT1_POL	0xbf0_1068	R/W	中断电平选择寄存器 1	0x0
INT1_EDGE	0xbf0_106c	R/W	中断边沿选择寄存器 1	0x0

寄存器名称	地址	读/写(R/W)	功能描述	复位值
INT1_SR	0xbf0_1058	R	中断状态寄存器 1	

INT1_SR	位	缺省值	描述
Gpio[104:96]	31:23	0	GPIO[104:96]作为中断输入, 中断状态位

Reserved	22:5	0	
cam_int	4	0	CAMERA 中断状态位
Gmac_sbd_int	3	0	MAC 中断状态位
otg_int	2	0	OTG 中断状态位
ohci_int	1	0	USB_OHCI 中断状态位
ehci_int	0	0	USB_EHCI 中断状态位

第三组中断寄存器

寄存器名称	地址	读/写(R/W)	功能描述	复位值
INT2_SR	0xbf0_1070	R	中断状态寄存器 2	0x0
INT2_EN	0xbf0_1074	R/W	中断使能寄存器 2	0x0
INT2_SET	0xbf0_1078	R/W	中断置位寄存器 2	0x0
INT2_CLR	0xbf0_107c	R/W	中断清 0 寄存器 2	0x0
INT2_POL	0xbf0_1080	R/W	中断电平选择寄存器 2	0x0
INT2_EDGE	0xbf0_1084	R/W	中断边沿选择寄存器 2	0x0

寄存器名称	地址	读/写(R/W)	功能描述	复位值
INT3_SR	0xbf0_1070	R	中断状态寄存器 2	0x0

INT3_SR	位	缺省值	描述
Gpio[31:0]	31:0	0	GPIO[31:0]作为中断输入, 中断状态位

第四组中断寄存器

寄存器名称	地址	读/写(R/W)	功能描述	复位值
INT3_SR	0xbf0_1088	R	中断状态寄存器 3	0x0

INT3_EN	0xbf0_108c	R/W	中断使能寄存器 3	0x0
INT3_SET	0xbf0_1090	R/W	中断置位寄存器 3	0x0
INT3_CLR	0xbf0_1094	R/W	中断清 0 寄存器 3	0x0
INT3_POL	0xbf0_1098	R/W	中断电平选择寄存器 3	0x0
INT3_EDGE	0xbf0_109c	R/W	中断边沿选择寄存器 3	0x0

寄存器名称	地址	读/写(R/W)	功能描述	复位值
INT3_SR	0xbf0_1088	R	中断状态寄存器 3	0x0

INT1_SR	位	缺省值	描述
Gpio[64:32]	31:0	0	GPIO[63:32]作为中断输入，中断状态位

#### 第五组中断寄存器

寄存器名称	地址	读/写(R/W)	功能描述	复位值
INT4_SR	0xbf0_10a0	R	中断状态寄存器 4	0x0
INT4_EN	0xbf0_10a4	R/W	中断使能寄存器 4	0x0
INT4_SET	0xbf0_10a8	R/W	中断置位寄存器 4	0x0
INT4_CLR	0xbf0_10ac	R/W	中断清 0 寄存器 4	0x0
INT4_POL	0xbf0_10b0	R/W	中断电平选择寄存器 4	0x0
INT4_EDGE	0xbf0_10b4	R/W	中断边沿选择寄存器 4	0x0

寄存器名称	地址	读/写(R/W)	功能描述	复位值
INT4_SR	0xbf0_10a0	R	中断状态寄存器 4	0x0

INT1_SR	位	缺省值	描述
Gpio[95:64]	31:0	0	GPIO[95:64]作为中断输入，中断状态位

## 6 SDRAM 控制器

龙芯 1C 处理器内部集成的 SDRAM 内存控制器，能实现的一般的内存读/写/休眠操作，支持 8/16 位数据宽度。

### 6.1 专用寄存器

SDRAM 有一个 64 位的配置寄存器 SD\_CONFIG[63:0]，其中有效为[41:0]，系统分配的配置寄存器为 SD\_CONFIG [63:0]，地址为 0x0410~0x0414，详细定义如下：

寄存器名称	地址	读/写(R/W)	功能描述	复位值
SD_CONFIG[31:0]	0xbf0_0410	R/W	SDRAM 参数配置寄存器	0x00143803
SD_CONFIG[63:32]	0xbf0_0414	R/W	SDRAM 参数配置寄存器	0x80000080

SD_CONFIG	位	缺省值	描述
CONFIG_VALID	41		配置寄存器有效
HANG_UP	40		挂起标志位
DEF_SEL	39		采用默认配置
TWR	38:37		颗粒的写后预充电时间
TREF	36:25		颗粒的自动刷新周期
TRAS	24:21		颗粒的行打开最小时间
TRFC	20:17		颗粒的自动刷新时间
TRP	16:14		颗粒的预充电时间
TCL	13:11		颗粒的读数据延时
TRCD	10:8		颗粒的行命令到列命令延时
SD_BIT	7:6		颗粒的位宽
SD_CSIZE	5:3	3'h0	颗粒的列数
SD_RSIZE	2:0	3'h0	颗粒的行数

### 6.2 配置说明

SD\_CONFIG 寄存器详细说明如下：

### 1. SD\_RSIZ、SD\_CSIZ

SD\_RSIZ 有效值为 00、01、10、11，分别对应 2K、4K、8K、16K；

SD\_CSIZ 有效值为 00、01、10、11、111，分别对应 512、1K、2K、4K、256；

这两项配置需要与颗粒对应，其中列数不能比行数大，所以有效配置为：

列 / 行	00	01	10	11
00	2Kx512	4Kx512	8Kx512 (不常见)	16Kx512(不常见)
01	2Kx1K	4Kx1K	8Kx1K	16Kx1K (不常见)
10	无	4Kx2K	8Kx2K	16Kx2K
11	无	无	8Kx4K	16Kx4K
111	2Kx256	4Kx256	8Kx256(不常见)	16Kx256(不常见)

### 2. SD\_BIT

SD\_BIT 的有效值为 00、01、10,分别对应 8bit、16bit 和 32bit。

### 3. TRCD、TCL、TRP、TRFC、TRAS、TREF、TWR

这些参数都和颗粒的物理特性相关，为物理时间，要根据频率换算成周期数，下面列出了一些典型频率值时的参数。

参数 / 频率(MHZ)	150	133	100	75	33
TRCD	3	3	2	2	1
TCL	3	3	3	2	2
TRP	3	3	2	2	1
TRFC	9	8	6	5	2
TRAS	7	6	5	4	2
TREF	'h926	'h818	'h620	'h494	'h204
TWR	2	2	2	1	1

### 4. DEF\_SEL

为 1 时采用默认配置（133MHZ 配置），为 0 时采用外部输入配置。采用默认配置时外部输入的寄存器配置无效。

### 5. HANG\_UP

为 1 时通知控制器进入挂起状态，SDRAM 完成队列里面的操作后进入休眠状态，直至 HANG\_UP 为 0 退出休眠状态；为 0 时正常工作，从休眠状态退出时需要将所有行都刷新一遍，至少需要 240~480us。SD\_CONFIG 寄存器只有这 1 位在正常工作是有效。

#### 6. CONFIG\_VALID

为 1 时寄存器配置有效；为 0 时无效，如果一直为低则 SDRAM 不能正常初始化，不能正常工作。建议先将参数配置好，最后再将这 1 位置高。

注：配置这两个寄存器，必须先写低位 0x410，再写高位 0x414，否则会出现 SDRAM 控制器会在写 0x414 时，将 0x410 初始化的数据当成配置参数，而之后再写的 0x410 数据无效，造成 SDRAM 配置错误，不能正常工作。建议写三次寄存器，最后一次将最高位置 1，配置寄存器使能。(valid 位在 0x414)

# 7 SRAM 控制器

## 7.1 功能描述

龙芯 1C 处理器内部集成的 SRAM 内存控制器，和 SDRAM 引脚复用，最大可支持 32MB，支持 8/16 位数据宽度。SRAM 控制器可以用于和 SRAM 接口时序相同的 IO 设备。

## 7.2 配置寄存器

内部只有一个配置寄存器 SRAM\_CTRL[15:0]，详细配置如下：

寄存器名称	地址	读/写(R/W)	功能描述	复位值
MISC_CTRL[15:0]	0xbf0_0424	R/W	SRAM 参数配置寄存器	0x0004

SRAM_CTRL	位	缺省值	描述
SRAM_EN	15	0x0	SRAM 使能，为 1 时 SDRAM 接口用作 SRAM，为 0 时用作 SDRAM 接口
IO_EN	14	0x0	IO 使能信号，为 1 时用作 IO 设备，配置 SRAM_CTRL[13:8]有效；反之用作 ROM 设备，配置 SRAM_CTRL[7:2]有效
IO_WIDTH16	13	0x0	IO 操作数据宽度，为 1 时表示 16 位，为 0 时表示 8 位
IO_INIT_CNT	12:8	0x0	IO 操作的计数初值，用于调整输出信号的相位，范围从 0~0x1f
ROM_WIDTH16	7	0x0	ROM 操作数据宽度，为 1 时表示 16 位，为 0 时表示 8 位
ROM_INIT_CNT	6:2	0x1	ROM 操作的计数初值，用于调整输出信号的相位，范围从 0~0x1f
CLOCK_PERIOD	1:0	0x0	SRAM 操作的计数步进值，用于调整输出信号的时间长度及操作快慢 00: 步进为 1，最慢 01: 步进为 2 10: 步进为 4，最快 11: 步进为 1，最慢

## 8 Camera 接口

### 8.1 功能概述

CAMERA 接口 (Camera Interface), 支持 ITU-R BT.601/656 YCbCr 8-bit 标准和 RGB565/888 8-bit 标准的输入。输入视频为 640x480 和 320x240 以及其他任意分辨率模式, 支持 RGB565\888、ITU-R BT.601 模式 640x480 分辨率下的缩小一倍, 其余分辨率不支持缩放功能, 以及有选择的提供 yuv4:2:2/rgb565/rgb888/rgb0888 格式输出。

Camera 接口控制器的功能特性包括:

- 支持 ITU-R BT.601/656 8-bit 和 RGB565/RGB888 8-bit 模式外部接口;
- 支持任意分辨率输入的配置;
- 仅 640x480 支持缩小 1 倍;
- 输出格式: YCbCr 4:2:2, RGB565, RGB888, RGB0888 (32 位), 当输出格式为 RGB565/ YCbCr 4:2:2, 输入像素必须是 32 的整数倍; 输出格式为 RGB888/RGB0888, 输入像素为 16 的整数倍;
- 支持任意像素分辨率下格式 ITU-R BT.601/656 8-bit 输入转 RGB (565/0888) 格式的输出, 不支持 RGB 输入转 YCbCr4:2:2 输出;
- 输出区域为 4 段各能容纳一帧图象的地址空间, 可分别配置各地址空间的基地址以及 YCbCr4:2:2 格式下 u 和 v 分量存放的偏移地址;
- 支持 RGB565 转 RGB565;
- 支持 RGB888 转 RGB888、RGB0888;
- 支持 BT601 转 RGB565、RGB0888、YUV;
- 支持 BT656 转 RGB565、RGB0888、YUV;
- 支持 640\*480 像素的缩小 1 倍(TU-R BT.656 格式除外);
- 支持 RGB 格式输出的矩阵显示。

该模块的接口框图:

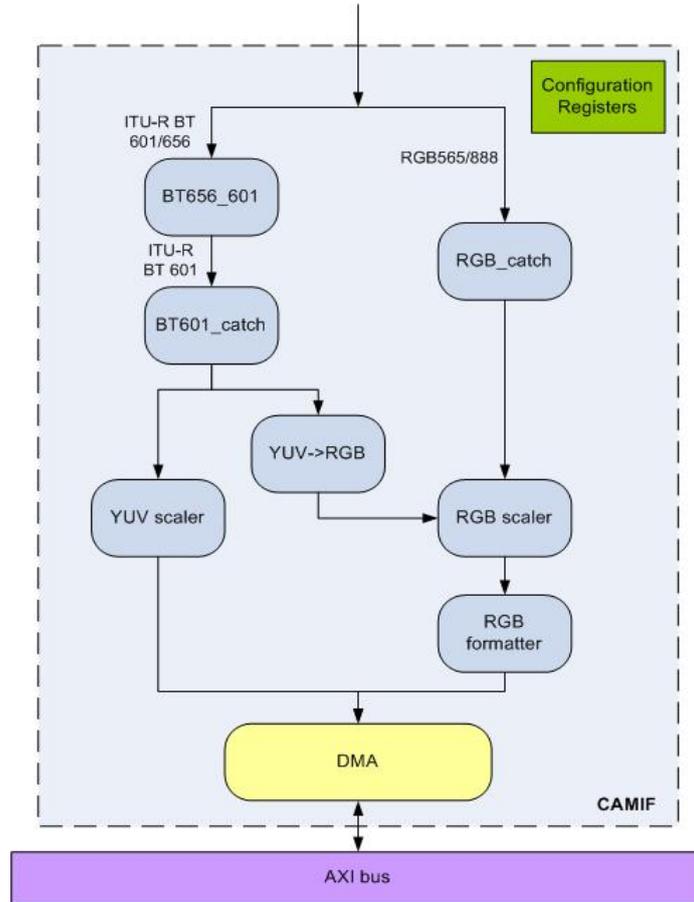


图 8-1 Camera 接口(CAMIF)功能框图

## 8.2 接口协议

PCLK : 1 位输入信号; Camera 处理器驱动的像素时钟。

VSYNC: 1 位输入信号; Camera 处理器驱动的帧同步信号。

HREF : 1 位输入信号; Camera 处理器驱动的行同步信号。

DATA : 8 位输入信号; Camera 处理器启动的像素数据。

ITU-R BT.601 8-bit 输入时序如图 1 所示, 其中数据输入顺序可以为 YCbY Cr 或者 YCrYCb 或者 CrYCbY 或者 CbYCrY。

RGB565/RGB888 8-bit 输入时序与 ITU-R BT.601 8-bit 输入时序一致, 仅数据输入顺序有区别。对于 RGB888, 数据输入顺序可以为 R G B 或 B G R。对于 RGB565, 数据输入顺序可以为 R5G3 G3B5 或 B5G3 G3R5。RGB565/RGB888 和 ITU-R BT.601 输入模式 可以根据状态寄存器的低两位来设置行有效和帧有效的高低电平。

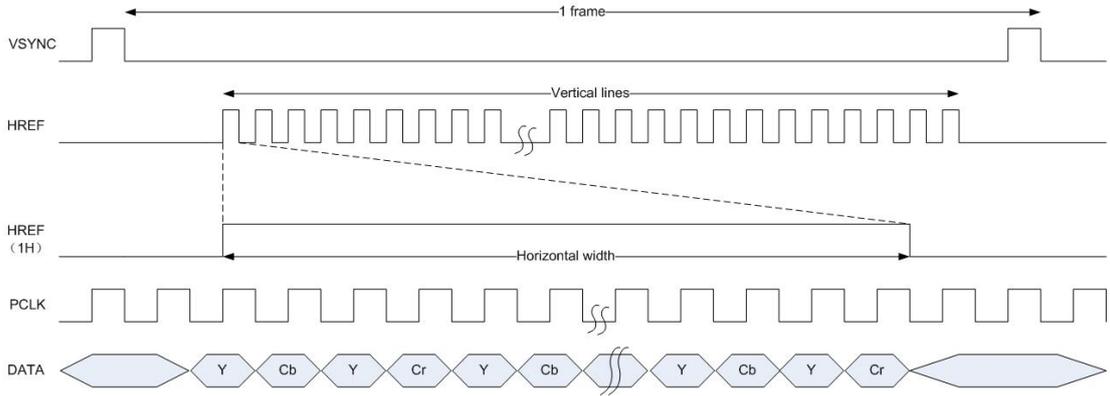


图 8-2 ITU-R BT.601 输入时序

ITU-R BT.656 输入时序如图 8-3 所示，其中数据输入顺序可以为 Y CbYCr 或者 YCrYCb 或者 CrYCbY 或者 CbYCrY。SAV 为行起始码，EAV 为行结束码。基准码（reference code）定义如表 8-1 所示，其中 XY 值的定义如表 8-2 所示。仅当 XY 为 80、9D 组合或者 C7、DA 组合时，为有效行数据。

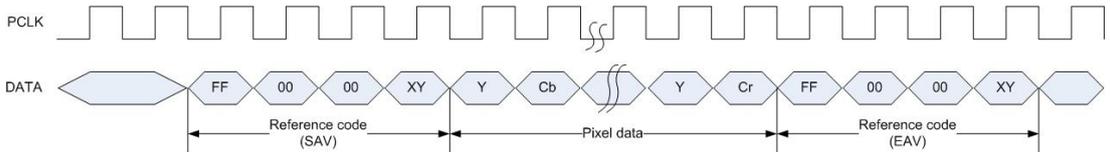


图 8-3 ITU-R BT.656 输入时序

表 8-1 ITU-R BT.656 基准码

数据比特编号	第一个字(3FF)	第二个字(000)	第三个字(000)	第四个字(XYZ)
7(MSB)	1	0	0	1
6	1	0	0	F
5	1	0	0	V
4	1	0	0	H
3	1	0	0	P <sub>3</sub>
2	1	0	0	P <sub>2</sub>
1	1	0	0	P <sub>1</sub>
0	1	0	0	P <sub>0</sub>

- F 在场 1 中为 0;在场 2 中为 1
- V 其它位置为 0;场消隐期间为 1
- H SAV 中为 0; EAV 中为 1
- P<sub>0</sub>, P<sub>1</sub>, P<sub>2</sub>, P<sub>3</sub>: 保护比特（见表 8-2）

表 8-2 第四字节 XY 值

MSB				LSB				XY
1	F	V	H	P <sub>3</sub>	P <sub>2</sub>	P <sub>1</sub>	P <sub>0</sub>	16 进制
1	0	0	0	0	0	0	0	80
1	0	0	1	1	1	0	1	9D
1	0	1	0	1	0	1	1	AB
1	0	1	1	0	1	1	0	B6
1	1	0	0	0	1	1	1	C7
1	1	0	1	1	0	1	0	DA
1	1	1	0	1	1	0	0	EC
1	1	1	1	0	0	0	1	F1

### 8.3 专用寄存器

帧缓冲区基地址配置寄存器 DMA\_ADDR<sub>i</sub>\_CONFIG (i=0,...,3)的配置定义如下:

寄存器名称	地址	读/写(R/W)	功能描述	复位值
DMA_ADDR0_CONFIG	0x1c280000	R/W	帧缓冲区基地址 0	32'h0
DMA_ADDR1_CONFIG	0x1c280008	R/W	帧缓冲区基地址 1	32'h0
DMA_ADDR2_CONFIG	0x1c280010	R/W	帧缓冲区基地址 2	32'h0
DMA_ADDR3_CONFIG	0x1c280018	R/W	帧缓冲区基地址 3	32'h0

DMA_ADDR <sub>i</sub> _CONFIG	位	缺省值	描述
帧缓冲区基地址	31:0	32'h0	软件分配 4 个缓冲区的基地址, 硬件从地址 0 到地址 3 依次存放好 4 帧数据, 然后循环回地址 0 存放。每个缓冲区大小为一帧图象大小。

视频分辨率配置寄存器 CAMIF\_CONFIG\_PIX 的配置定义如下。其中, 除

640x480 和 320x240 外，其余视频分辨率需要通过配置 CAMIF\_CONFIG\_PIX 寄存器进行定义。

寄存器名称	地址	读/写(R/W)	功能描述	复位值
CAMIF_CONFIG_PIX	0x1c280020	R/W	视频分辨率配置寄存器	32'h0

CAMIF_CONFIG_PIX	位	缺省值	描述
浮动像素分辨率行数 y	23:12	12'h0	分辨率行数，例如 1280x720 分辨率下，配置为 720
浮动像素分辨率列数 x	11:0	12'h0	分辨率列数，例如 1280x720 分辨率下，配置为 1280

U、V 基地址配置寄存器（CAMIF\_CONFIG\_UOFFSET 和 CAMIF\_CONFIG\_VOFFSET）的定义如下表所示。除 640x480 和 320x240 之外的视频分辨率，配置缓冲区需要通过基址配置寄存器指定 YUV4:2:2 格式中 U 和 V 分量在帧缓冲区中的存放基地址。而在进行 RGB 输出显示时，CAMIF\_CONFIG\_UOFFSET 配置矩阵显示地址的偏移量，即一副图像行与行之间的地址间隔。例如，640x480 的图像显示在 1280x720 的屏幕上，则  $uoffset=(1280-640) \times$  像素字节。

寄存器名称	地址	读/写(R/W)	功能描述	复位值
CAMIF_CONFIG_UOFFSET	0x1c280028	R/W	U 基地址	32'h0
CAMIF_CONFIG_VOFFSET	0x1c280030	R/W	V 基地址	32'h0

CAMIF_CONFIG_UOFFSET	位	缺省值	描述
U 基地址配置寄存器	31:0	32'h0	YUV4:2:2 格式帧缓冲区的 1/2 处为 u 分量的基址。例如 1280x720 分辨率下，voffset 为十进制 1280x720，单位为字节。
CAMIF_CONFIG_VOFFSET	位	缺省值	描述
V 基地址配置寄存器	31:0	32'h0	YUV4:2:2 格式帧缓冲区的 3/4 处为 v 分量的基址。例如 1280x720 分辨率下，voffset 为十进制 1280x720x3/2，单位为字节。

CAMIF\_CONFIG 配置是启动和关闭 camera 模块的控制寄存器，定义如下：

寄存器名称	地址	读/写(R/W)	功能描述	复位值
CAMIF_CONFIG	0x1c280038	R/W	状态寄存器	32'h0

CAMIF_CONFIG	位	缺省值	描述
CONFIG_PARA_CLR	31:31	1'b0	1: 启动 CAMERA 0: 关闭 CAMERA
CONFIG_BUF_FULL	23:20	4'b0	由低到高分别对应 4 个帧缓冲区。 1: 写满 0: 没写过或者正在写
CONFIG_PARA_H_WD	19:16	4'b0	等待 Hsync 有效的延迟拍数
CONFIG_PARA_OUTPUT_DATA_MODE	14:13	2'b0	0: YUV422 1: RGB565 2: RGB888 3: RGB0888
CONFIG_PARA_INPUT_DATA_MODE	12:11	2'b0	0: RGB 1: ITU-R BT.601 2: ITU-R BT.656
CONFIG_PARA_SCALE_MODE	10:9	2'b0	0: 不缩放 1: 缩 1 倍 可配像素时固定配为 0。
CONFIG_PARA_640X480	8:7	2'b0	0: 320x240 1: 640x480 2: 可配像素
CONFIG_PARA_SINGLEFIELDEN	6:6	1'b0	1: 单场 0: 非单场 (帧模式下配 0)
CONFIG_PARA_YUV_ORDER	5:4	2'b0	YUV422 的排列格式。 0: YCbYCr 1: YCrYCb 2: CrYCbY 3: CbYCrY
CONFIG_PARA_BGR_EN	3:3	1'b0	1: RGB 排列为 BGR 0: RGB 排列为 RGB (ITU-R BT.601/656 转

			RGB 有效)
CONFIG_PARA_RGB_FORM AT	2:2	1'b0	0: RGB 输入格式为 565 1: RGB 输入格式为 888
CONFIG_PARA_HS	1:1	1'b0	1: HSYNC 低电平有效 0: HSYNC 高电平有效
CONFIG_PARA_VS	0:0	1'b0	1: VSYNC 低电平有效 0: VSYNC 高电平有效

注：输入格式为 ITU-R BT.656 时 Config\_para\_hs、Config\_para\_vs 只能配 0。

## 8.4 配置操作

### 8.4.1 访存方式

Camera 接口控制器的输出数据，其帧存储层次如下：

CAMIF 输出的帧存储由 4 个循环的存储空间组成。对于 YCbCr 输出，每个存储空间由亮度 Y、色度 Cb 和 Cr 三个存储空间组成，yuv3 种分量存放的起始地址分别对应空间的基地址处，1/2 地址处，3/4 地址处；对于 RGB 输出，则每个存储空间顺次存放每个 RGB 像素。

Camera 接口控制器的输出数据在内存中的存储方式如下：

帧存储中采用小尾端存储方式。AXI 总线宽度为 64 位。输出为 YCbCr 格式时，每个 Y 或 Cb 或 Cr 各一个字节，分别排放到帧地址的各自对应区域中；输出为 RGB565 时，每个像素 2 个字节；输出为 RGB888（32 位）时，每个像素 4 个字节；输出为 RGB888（24 位）时，每个像素 3 个字节。具体如图 8-4 所示。

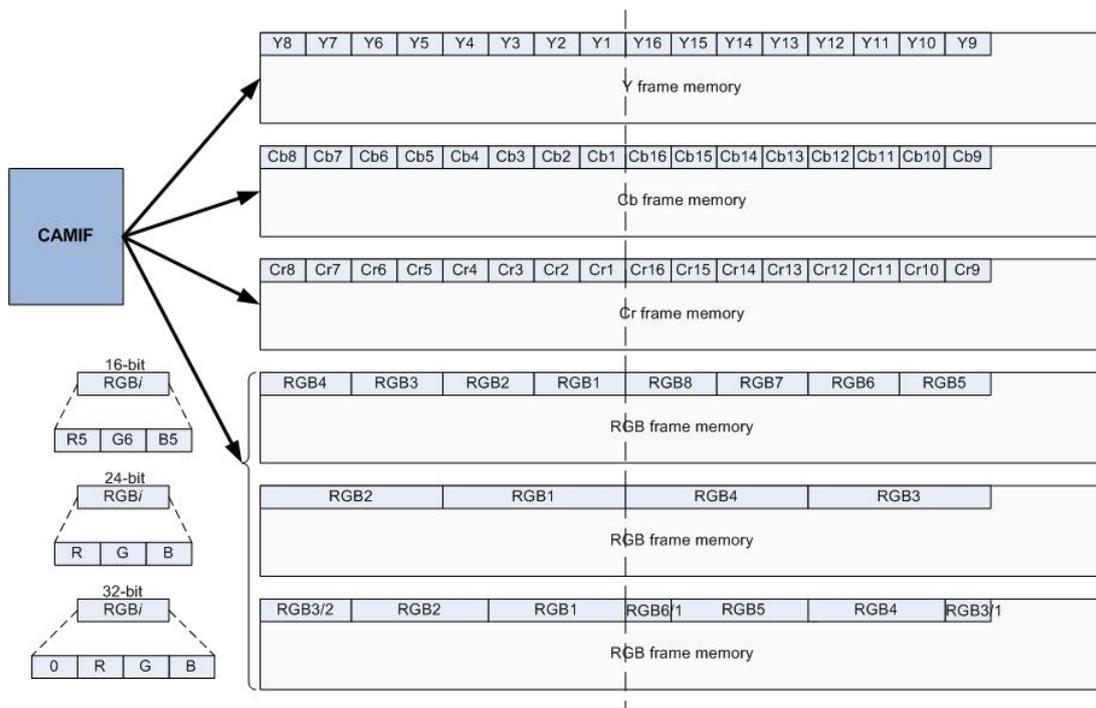


图 8-4 存储方式

### 8.4.2 配置顺序

CAMERA 接口在工作之前需要根据不同的转化模式对一些寄存器进行配置，系统在复位之后，寄存器都是缺省值，只有对各个寄存器进行配置后模块才能工作。配置 CAMERA 接口的工作流程如下：

1. 读取 CAMERA 接口中状态寄存器信息，得到其第 30 位，如果为 1 则表示里面的 DMA 处于闲置状态，可以进行下次工作配置。这是因为 DMA 需要和内存交互，在我们控制 CAMERA 接口工作和停止时，DMA 无法做到实时停止。
2. 在进行工作配置时，配置顺序为先配置获取图象所要存储到的 4 段地址空间基地址，如果采用任意像素的配置并输出 YUV 格式则还需要配置 U 和 V 地址，最后配置状态寄存器，将其第 31 位配置为 1 即可开始工作。
3. CAMERA 接口在工作过程中，每完成一个帧地址空间的填写或者 DMA 模块 buffer 中的数据被覆盖都会发送一个中断给 CPU，软件收到中断后可以查询状态寄存器的第 23-20 位和 28 位，这 5 位在刚启动 CAMERA 接口时初始化为全 0，其后每填满一个帧地址空间 23-20 位就由低到高

值依次变 1，当正在填写一个帧地址空间时，该空间所对应的位变为 0，软件也可自己一直查询状态寄存器来判断帧地址空间的存储状况。当有数据被覆盖 28 位就会置 1。

4. 当软件要停止 CAMERA 接口工作时，只需要将状态寄存器的第 31 位写为 0 即可。CAMERA 接口在此情况下会停止对 CAMERA 的输入进行分析，DMA 会停止发送任何数据到帧地址空间，并在接受到所有已发送的请求应答后，将状态寄存器的第 30 位置 1。
5. 当 CAMERA 接口的应用软件被其他中断所打断时，可以在中断处理程序中加入关闭 CAMERA 接口的配置请求等待恢复后重启（首先将状态寄存器的第 31 位写为 0，然后去读取第 30 的值，为 1 则将第 29 位写 1，再次读取第 30 位，为 0 则将第 31 位写 1），也可不关闭 CAMERA 接口而在恢复后根据状态寄存器的信息自己判断帧地址空间的存储信息进行图象的取舍。

# 9 I2S 控制器

## 9.1 功能概述

1C 中 I2S 控制器，通过 APB 接口，数据宽度是 32 位，支持 DMA 传输，支持多家公司的 codec 芯片。I2S 控制器仅支持的主模式，由 I2S 产生位时钟信号和左右声道选择时钟信号。

I2S 的功能特性包括：

- 支持 8、16、20、24、32 位的音频数据采样位宽。
- 支持 8、16、20、24、32 位的左右声道处理字宽。
- 包含两个缓存 FIFO，FIFO 的缓存容量为 8 bytes。
- I2S 的中断处理模式可配，在 I2S 的发送和接收中断功能都使能后，当两个通道的缓存 fifo 为满仍要写以及为空仍要读时，则向 CPU 发出中断信号。
- I2S 可以为 codec 芯片提供系统时钟，时钟频率可配。

该模块的接口框图：

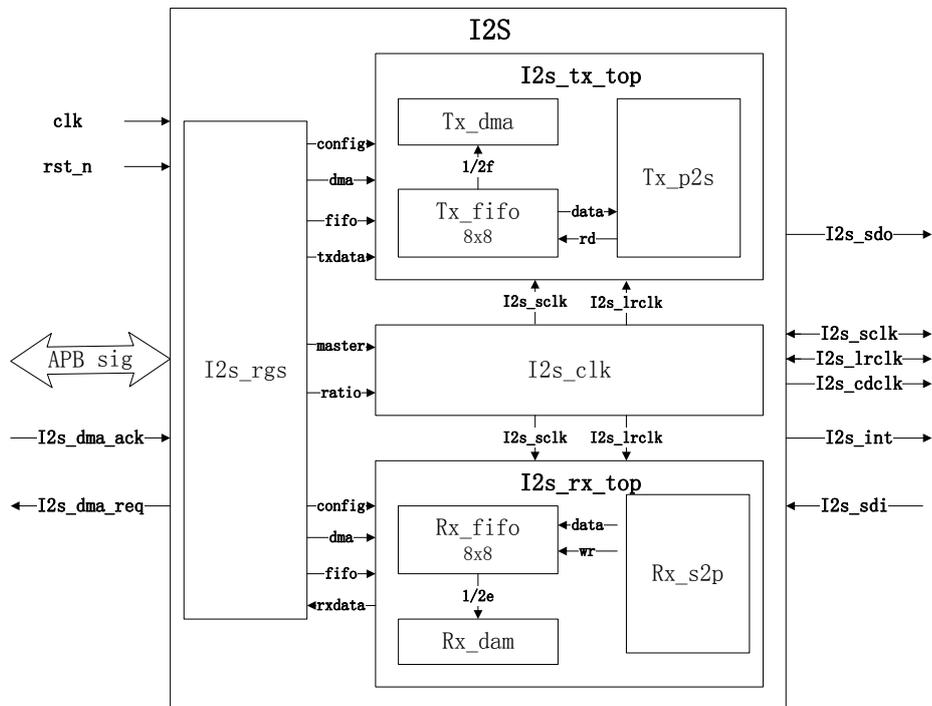


图 9-1 I2S 接口框图

## 9.2 接口协议

I2S 接收和接收器的操作时序如图 9-2 所示。发送器和接收器的 I2S 格式传输操作时序要求都是在信道选择信号（WS）改变之后的第二个位时钟开始传输下一帧数据，数据先传 MSB 位，再传输 LSB 位。发送器和接收器能处理的字位宽可以不一致，若发送器所需发送的数据的位宽比系统所支持的数据位宽要短，则 LSB 补零发送，反之，则忽略部分 LSB 数据；同理，对接受器而言，当接收到的数据的位宽比自己处理的位宽小时，则 LSB 补零接收，反之，则忽略接收部分 LSB 数据。所以，对收发的数据而言，MSB 是具有固定的，而 LSB 则取决于要收发的数据的字长已经系统所配置的字长位宽。

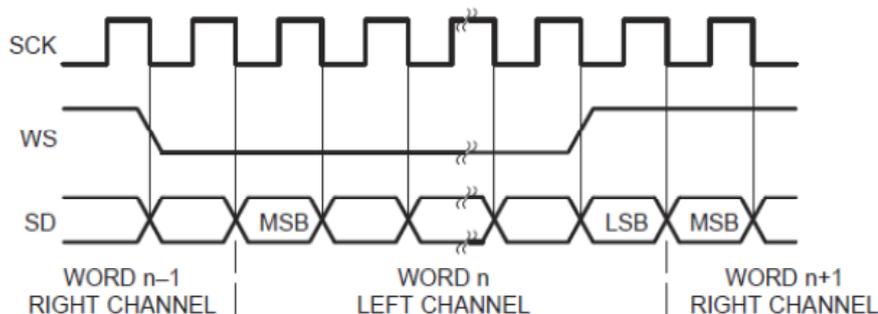


图 9-2 I2S 传输协议

## 9.3 专用寄存器

I2S 包含 5 个寄存器，定义如下图所示。

寄存器名称	地址	读/写(R/W)	功能描述	复位值
IISVersion	0xbfe6_0000	R/W	I2S 标识寄存器	32'h0
IISConfig	0xbfe6_0004	R/W	I2S 配置寄存器	32'h0
IISControl	0xbfe6_0008	R/W	I2S 控制寄存器	32'h0
IISRxDData	0xbfe6_000c	R/W	I2S 接收数据寄存器(用于 DMA 接收数据)	32'h0
IISTxDData	0xbfe6_0010	R/W	I2S 发送数据寄存器(用于 DMA 发送数据)	32'h0

接收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。

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

接收配置寄存器是配置 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的。

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。
SCK_RATIO	15:8	'h0	SCK 时钟分频系数： IIS 作用于主模式，系统时钟分频系数，总线时钟频率除以 2x(RATIO+1)得到 SCK 的时钟频率
BCK_RATIO	7:0	'h0	BIT_CLK 时钟的分频系数 IIS 作用于主模式，系统时钟分频系数，总线时钟频率除以 2x(RATIO+1)得到 BCK 的时钟频率

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。

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	接收使能，为 1 时有效，开始接收数据
TX_EN	12	'h0	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
RX_FIFO_THR	10:9	'h0	01: 发 DMA 请求
TX_DMA_EN	7	'h0	DMA 发送使能，为 1 时有效
Reserved	6:2	'h0	
RX_INT_EN	1	'h0	0: 禁止 RX 中断输出 1: 允许 RX 中断输出
TX_INT_EN	0	'h0	0: 禁止 TX 中断输出 1: 中断 TX 允许输出

## 9.4 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。

1C 芯片通过 I2C 接口和 CODEC 芯片通信，CODEC 芯片作为 I2C 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。SCK 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：

$$SCK = 256xfs(\text{或者 } 512xfs) \text{ 或者 } (768xfs); \text{ (详见 CODEC 手册推荐配置)}$$

$$SCK\_RATIO = \text{Freq\_SDRAM} / (256 \times fs) / 2 - 1;$$

双通道的话，计算公式下：

$$BCK = \text{RES\_DEPTH} \times 2 \times fs;$$

$$BCK\_RATIO = \text{Freq\_SDRAM} / (\text{RES\_DEPTH} \times 2 \times fs) / 2 - 1;$$

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。DMA 复用的配置方法见第 14.3 节。

# 10 显示控制器(DC)

## 10.1 概述

显示控制器从内存中取帧缓冲输出到外部显示接口上。

龙芯 1C 的显示控制器支持的特性包括：

- 一路 DVI 显示，最大支持至 2048x2048@60Hz
- RGB444,RGB555,RGB565,RGB888, RGB8888 五种数据格式
- 输出抖动和伽马校正

## 10.2 寄存器定义和说明

### 10.2.1 帧缓冲配置寄存器

表 10-1 帧缓冲配置寄存器

寄存器名	地址	R/W	描述	复位值
frameBufferConfig	0xbc30_1240	R/W	帧缓冲配置寄存器	32' h0
frameBufferConfig	bit		描述	初始值
Reset	20		从值 1 变为 0 时软复位	0
Gamma	12		写 1 使能伽马校正	0
Output Enable	8		写 1 使能显示输出	0
Format	2:0		色深格式： 0: none 1: RGB444 2: RGB555 3: RGB565 4: RGB888 5: RGB8888	0

### 10.2.2 帧缓冲地址寄存器 0

表 10-2 帧缓冲地址寄存器 0

寄存器名	地址	R/W	描述	复位值
frameBufferAddr0	0xbc301260	R/W	帧缓冲地址寄存器 0	32' h0
frameBufferAddr0	bit		描述	初始值
Address	31:0		缓冲区 0 在内存中的物理地址	0

### 10.2.3 帧缓冲地址寄存器 1

表 10-3 帧缓冲地址寄存器 1

寄存器名	地址	R/W	描述	复位值
frameBufferAddr1	0xbfe51580	R/W	帧缓冲地址寄存器 1	32' h0
frameBufferAddr1	bit		描述	初始值
Address	31:0		缓冲区 1 在内存中的物理地址(需与帧缓冲地址寄存器 0 相同)	0

### 10.2.4 帧缓冲跨度寄存器

表 10-4 帧缓冲跨度寄存器

寄存器名	地址	R/W	描述	复位值
frameBufferStride	0xbc301280	R/W	帧缓冲跨度寄存器	32'h0
frameBufferStride	bit		描述	初始值
Stride	31:0		显示缓冲一行的字节数。 当使用 RGB888 (24 位) 格式时, stride 需要按 120 字节向上取整。 当使用 RGB8888 (32 位) 格式时, stride 需要按 128 字节向上取整。	0

### 10.2.5 颜色抖动配置寄存器

表 10-5 颜色抖动配置寄存器

寄存器名	地址	R/W	描述	复位值
ditherConfig	0xbc301360	R/W	颜色抖动配置寄存器	32'h0
ditherConfig	bit		描述	初始值
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

### 10.2.6 颜色抖动查找表低位寄存器

表 10-6 颜色抖动查找表低位寄存器

寄存器名	地址	R/W	描述	复位值
ditherTableLow	0xbc301380	R/W	颜色抖动查找表低位寄存器	32'h0
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

### 10.2.7 颜色抖动查找表高位寄存器

表 10-7 颜色抖动查找表高位寄存器

寄存器名	地址	R/W	描述	复位值
ditherTableHigh	0xbc3013a0	R/W	颜色抖动查找表高位寄存器	32'h0
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

### 10.2.8 颜色抖动说明

颜色抖动功能用于根据一定规则来增强像素值。

在下面的这个例子中，将解释颜色抖动功能的实现步骤。

首先，确定是对哪一数据位进行增强（即该数据位加 1），为此需要配置寄存器 Display Dither Configuration。比如配置 RedSize 为 6（1 到 8 之间，包含 1 和 8），则表明对从 MSB 位数起第 6 位进行增强，即 RedColor[7:0]的 RedColor[2]位增强。GreenSize 和 BlueSize 同理。

其次，需要建立查找表，即配置寄存器 Display Dither Table。

查找表包含 16 个条目，即 16 个阈值，每个条目 4 位宽。

查找表通过屏幕像素计数器的横坐标 x 的最低两位 x[1:0]和纵坐标 y 的最低两位 y[1:0]进行索引，得到一个阈值 U[3:0]。

对应屏幕 (x, y) 位置的像素值的最低四位被拿来跟查到的这个阈值比较。

如果 RedColor[3:0] > U[3:0]，并且 RedColor[7:2]不是 6'b111111，则 RedColor[2]位加 1，实现颜色增强。

### 10.2.9 液晶面板配置寄存器

表 10-8 液晶面板配置寄存器

寄存器名	地址	R/W	描述	复位值
panelConfig	0xbc3013c0	R/W	液晶面板配置寄存器	32'h101
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

### 10.2.10 水平显示宽度寄存器

表 10-9 水平显示宽度寄存器

寄存器名	地址	R/W	描述	复位值
HDisplay	0xbc301400	R/W	水平显示宽度寄存器	32'h0
HDisplay	bit		描述	初始值
Total	27:16		显示屏一行的总像素数（包括非显示区）	0
Display	11:0		显示屏一行中显示区的像素数	0

### 10.2.11 行同步配置寄存器

表 10-10 行同步配置寄存器

寄存器名	地址	R/W	描述	复位值
HSync	0xbc301420	R/W	行同步配置寄存器	32'h4000000
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	27:16	行同步结束时的像素数	0
Start	11:0	行同步开始时的像素数	0

### 10.2.12 垂直显示高度寄存器

表 10-11 垂直显示高度寄存器

寄存器名	地址	R/W	描述	复位值
VDisplay	0xbc301480	R/W	垂直显示高度寄存器	32' h0
VDisplay	bit		描述	初始值
Total	26:16		显示屏一列的总像素数（包括非显示区）	0
Display	10:0		显示屏一列中显示区的像素数	0

### 10.2.13 场同步配置寄存器

表 10-12 场同步配置寄存器

寄存器名	地址	R/W	描述	复位值
VSync	0xbc3014a0	R/W	场同步配置寄存器	32' h40000000
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	26:16		场同步结束时的像素数	0
Start	10:0		场同步开始时的像素数	0

### 10.2.14 输出定序说明

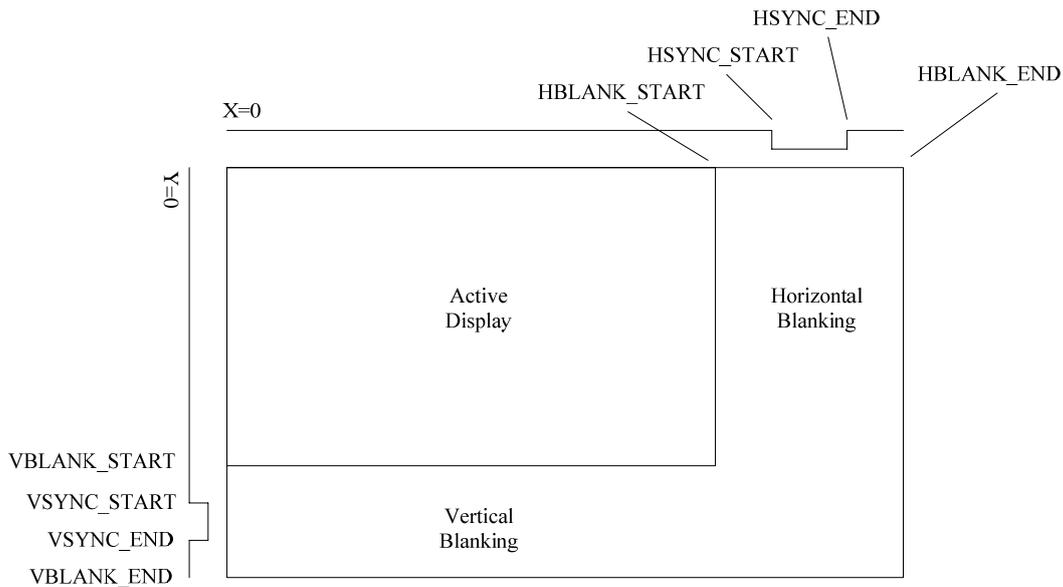


图 10-1 显示控制器输出定序

### 10.2.15 伽马校正目录寄存器

表 10-13 伽马校正目录寄存器

寄存器名	地址	R/W	描述	复位值
GammaIndex	0xbc3014e0	R/W	伽马校正目录寄存器	32' h0
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

### 10.2.16 伽马校正寄存器

表 10-14 伽马校正寄存器

寄存器名	地址	R/W	描述	复位值
GammaData	0xbc301500	R/W	伽马校正寄存器	32'h0
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

### 10.2.17 伽马校正说明

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……

# 11 MAC 控制器

## 11.1 DMA 寄存器描述

龙芯 1C 集成了 1 个 MAC 控制器。MAC 内部集成独有的 DMA 控制器，专门配合 MAC 数据传输；该 DMA 控制器不能被其他模块使用，MAC 也无法使用外部其他 DMA。MAC 控制器寄存器包括 MAC 寄存器部分和 DMA 寄存器部分。MAC 寄存器的起始地址是 0x1fe1\_0000；DMA 寄存器的起始地址是 0x1fe1\_1000。

下面分别介绍 DMA 寄存器和 MAC 寄存器的意义。

### 11.1.1 Register0 (Bus Mode Register)

参数名称	位	缺省值	描述
Register0 (Bus Mode Register) Offset: 0x00			
Reserved 保留	31:27	0x0	保留，只读
MB: Mixed Burst 混合突发访问	26	0x0	当此位为高，FB 位为低时，AXI master 在突发访问长度大于 16 时采用 INCR 访问模式，当突发访问长度为 16 或者小于 16 时采用 FIX 访问模式。用户不用关心此位设置。
AAL:Address-Align ed Beats 地址对齐节拍	25	0x0	当此位和 FB 位同时为高时，AXI 接口的所有访问将对齐到起始地址的 LS 位。如果 FB 位为 0，首次访问地址访问不对齐，剩余的访问地址对齐。用户不用关心此位设置。
8XPBL Mode 是否使能 PBLX8 模式	24	0x0	此位为高时，MAC DMA 的最大突发数据传输长度为 8,16,32,64,128 或者 256。最大突发长度取决于 PBL。用户不用关心此位设置。
USP:Use Separate PBL 使用分离的 PBL 值	23	0x0	此位为高时，PBL 值只应用于 TxDMA。此位为低时，PBL 值应用于 TxDMA 和 RxDMA。用户不用关心此位设置。
RPBL: RxDMA PBL RxDMA 突发传输长度	22:17	0x01	表示一次 RxDMA 传输的最大突发传输长度。只能为 1,2,4,8,16 和 32，其它值无效。
FB: Fixed Burst 定长突发传输长度	16	0x0	指定 AXI Master 接口是否采用 FIX 突发传输模式。用户不用关心此位设置。

使能			
PR: Rx:Tx priority ratio RxDMA 与 TxDMA 优先级比例	15:14	0x0	在 DA 位为 0 时起作用。 00: 1: 1 01: 2: 1 10: 3: 1 11: 4: 1
PBL: Programmable Burst Length 可编程突发传输长度	13:8	0x1	用户不用关心此设置。
ATDS: Alternate Descriptor size 是否使用 32 字节大小描述符	7	0x0	此位为 1 时使用 32 字节大小的描述符 此位为 0 时使用 16 字节大小的描述符
DSL: Descriptor Skip Length 描述符间隔距离	6:2	0x00	设置 2 个描述符间的距离。但此值为 0 时，默认为 DMA 描述符大小。
DA: DMA Arbitration scheme DMA 传输仲裁策略	1	0x0	0: 在 RxDMA 和 TxDMA 间采用轮转仲裁机制 1: RxDMA 优先级高于 TxDMA 优先级。具体比值见 PR 值。
SWR: Software Reset 软件复位	0	0x1	此位置高 DMA 控制器将复位 MAC 内部寄存器和逻辑。当复位结束时该位自动清零。

### 11.1.2 Register1 (Transmit Poll Demand Register)

Register1 (Transmit Poll Demand Register) Offset: 0x04			
TPD: Transmit Poll Demand 传输轮询使能	31:0	0x0	向此值写入任意值，发送 DMA 控制器将会读取寄存器 18 对应的描述符。如果该描述符无效，DMA 传输将会停止。如果该描述符有效，DMA 传输将会继续。

### 11.1.3 Register2 (Receive Poll Demand Register)

Register2 (Receive Poll Demand Register) Offset: 0x08			
RPD: Receive Poll Demand	31:0	0x0	向此值写入任意值，接收 DMA 控制器将会读取寄存器 18 对应的描述符。如果该描述符无效，DMA 传输将会停止。如果该描述符有效，DMA 传输将会继续。

接收轮询使能			续。
--------	--	--	----

#### 11.1.4 Register3 (Receive Descriptor List Address Register)

Register3 (Receive Descriptor List Address Register) Offset: 0x0C			
Start of Receive List 接收描述符起始地址	31:0	0x0	指向接收描述符首地址。

#### 11.1.5 Register4 (Transmit Descriptor List Address Register)

Register4 (Transmit Descriptor List Address Register) Offset: 0x10			
Start of Transmit List 发送描述符起始地址	31:0	0x0	指向发送描述符首地址

#### 11.1.6 Register5 (Status Register)

Register5 (Status Register) Offset: 0x14			
Reserved	31:30		保留，只读
TTI: Time-Stamp Trigger Interrupt 时间戳触发中断	29	0x0	时间戳模块触发中断。只读。
GPI:MAC PMT Interrupt 电源管理模块触发中断	28	0x0	电源管理模块触发中断。只读。
GMI:MAC MMC Interrupt MMC 模块触发中断	27	0x0	MMC 模块触发中断。只读。
GLI:MAC Line interface Interrupt MAC 模块线路触发中断	26	0x0	MAC 模块的 PCS 或者 RGMII 模块触发中断。只读。
EB: Error Bits 错误位	25:23	0x0	23: 1'b1 TxDMA 数据传输过程中发生错误 1'b0 RxDMA 数据传输过程中发生错误

			<p>24: 1'b1 读传输错误 1'b0 写传输错误</p> <p>25: 1'b1 描述符访问错误 1'b0 数据缓存访问错误</p>
<p>TS:Transmit Process State 传输过程状态</p>	22:20	0x0	<p>3'b000:传输停止；复位或者停止命令发送</p> <p>3'b001:正在进行；获取传输描述符</p> <p>3'b010:正在进行；等待传输状态</p> <p>3'b011:正在进行；从发送缓存读取数据并发送到传输 FIFO(TxFIFO)</p> <p>3'b100:写入时间戳状态</p> <p>3'b101:保留</p> <p>3'b110:挂起；传输描述符不可用或者传输缓存下溢。</p> <p>3'b111:运行；关闭传输描述符。</p>
<p>RS:Receive Process State 接收过程状态</p>	19:17	0x0	<p>3'b000:停止；复位或者接收到停止命令</p> <p>3'b001:运行；获取接收描述符。</p> <p>3'b010:保留；</p> <p>3'b011:运行；等待接收包。</p> <p>3'b100:暂停；接收描述符不可用。</p> <p>3'b101:运行；关闭接收描述符。</p> <p>3'b110:时间戳写状态。</p> <p>3'b111:运行；将包内容从接收缓存传输到系统内存。</p>
<p>NIS:Normal Interrupt Summary 正常中断汇总</p>	16	0x0	提示系统是否存在正常中断。
<p>AIS:Abnormal Interrupt Summary 异常中断汇总</p>	15	0x0	提示系统是否存在异常中断。
<p>ERI:Early Receive Interrupt 提前接收中断</p>	14	0x0	提示 DMA 控制器已经把包的第一个数据写入接收缓存
<p>FBI:Fatal Bus Error Interrupt 总线错误中断</p>	13	0x0	提示总线错误，具体信息见[25:23]。当此位设置后 DMA 引擎停止总线访问操作。

Reserved	12:11	0x0	保留
ETI:Early Transmit Interrupt 提前发送中断	10	0x0	提示需要传输的以太网帧已经完全传输到 MTL 模块中的传输 FIFO
RWT:Receive Watchdog Timeout 接收看门狗超时	9	0x0	提示接收到一个大小超过 2048 字节的以太网帧。(当巨帧使能时, 提示接收到大小超过 10240 字节的以太网帧)
RPS:Receive Process Stopped 接收过程停止	8	0x0	指示接收过程停止
RU:Receive Buffer Unavailable 接收缓存不可用	7	0x0	指示接收缓存不可用
RI:Receive Interrupt 接收中断	6	0x0	指示帧接收完成。帧接收的状态信息已经写入接收描述符。接收处于运行状态。
UNF:Transmit Underflow 传输缓存下溢	5	0x0	指示帧发送过程中产生接收缓存下溢。
OVF:Receive Overflow 接收缓存上溢	4	0x0	指示帧接收过程中接收缓存上溢。
TJT:Transmit Jabber Timeout	3	0x0	
TU:Transmit Buffer Unavailable 传输缓存不可用	2	0x0	提示传输列表中的下一个描述符不能被 DMA 控制器访问。
TPS:Transmit Process Stopped 传输过程停止	1	0x0	提示传输过程停止
TI:Transmit Interrupt 传输完成中断	0	0x0	提示帧传输完成并且第一个描述符的 31 位置位。

### 11.1.7 Register6 (Operation Mode Register)

Register6 (Operation Mode Register) Offset: 0x18			
Reserved 保留	31:27	0x0	保留
DT 关闭丢弃 TCP/IP Checksum 错误以 以太网帧的功能	26	0x0	此位为 1 时 MAC 将不丢弃 checksum 错误的以太网帧。
RSF:Receive Store and Forward 接收存储转发	25	0x0	此位为 1 时 MTL 模块只接收已经全部存储在接收 FIFO 中的以太网帧。
DFF:Disable Flushing of Received Frames 关闭冲刷接收的以 以太网帧的功能	24	0x0	此位为 1 时, 接收 DMA 在接收描述符或者接收缓存不可用时不冲刷任何以太网帧。
RFA[2]:MSB of Threshold for Activating Flow Control 激活流控阈值	23	0x0	100: 最大值减去 5KB 101: 最大值减去 6KB 110: 最大值减去 7KB 111: 保留 (注: 最大值为 8KB)
RFD[2]:MSB of Threshold for Deactivating Flow Control 关闭流控阈值	22	0x0	100: 最大值减去 5KB 101: 最大值减去 6KB 110: 最大值减去 7KB 111: 保留 (注: 最大值为 8KB)
TSF:Transmit Store and Forward 发送存储转发	21	0x0	此位为 1 时, 帧的发送只在帧的内容已经全部进入 MTL 的传输 FIFO 中。
FTF:Flush Transmit FIFO 冲刷传输 FIFO	20	0x0	此位为 1 时, 传输控制逻辑复位为默认值, 并且会导致发送 FIFO 里面的数据全部丢失。
Reserved	19:17	0x0	保留
TTC:Transmit	16:14	0x0	当帧大小超过此值时 MTL 将会传输该帧。

Threshold Control 传输阈值控制			000: 64 字节 001: 128 字节 010: 192 字节 011: 256 字节 100: 40 字节 101: 32 字节 110: 24 字节 111: 16 字节
ST:Start/Stop Transmission Command 开始/停止传输命令	13	0x0	此位为 1，传输进入运行状态。 此位为 0，传输进入停止状态。
RFD:Threshold for deactivating flow control 关闭流控阈值	12:11	0x0	00: 最大值减去 1KB 01: 最大值减去 2KB 10: 最大值减去 3KB 11: 最大值减去 4KB (最大值为 8KB)
RFA:Threshold for Activating flow control 激活流控阈值	10:9	0x0	00: 最大值减去 1KB 01: 最大值减去 2KB 10: 最大值减去 3KB 11: 最大值减去 4KB (最大值为 8KB)
EFC:Enable HW flow control 使能硬件流控	8	0x0	此位为 1 时，基于接收 FIFO 利用率的硬件流控电路生效。
FEF:Forward Error Frames 传输错误帧	7	0x0	此位为 1 时，接收错误帧（错误帧包括：CRC 错误，冲突错误，巨帧，看门狗超时，溢出等）
FUF:Forward Undersized Good Frames 接收无错误的小帧	6	0x0	此位为 1 时，接收 FIFO 将会接收没有错误但小于 64 字节的以太网帧。
Reserved	5	0x0	保留

RTC:Receive Threshold Control 接收阈值控制	4:3	0x0	MTL 传输接收 FIFO 中帧内容已经超过此项设置大小。 00: 64 字节 01:32 字节 10: 96 字节 11: 128 字节
OSF:Operation Second Frame 是否操作第二个以太网帧	2	0x0	此位为高时，DMA 在第一个以太网帧的状态尚未写回时即可以开始处理第二个以太网帧。
SR:Start/Stop Receive 开始/停止接收	1	0x0	此位设置为高时，接收进入运行状态。 此位设置为低时，接收进入停止状态。
Reserved 保留	0	0x0	保留

### 11.1.8 Register7 (Interrupt Enable Register)

Register7 (Interrupt Enable Register) Offset: 0x1C			
Reserved 保留	31:17	0x0	保留
NIE:Normal Interrupt Summary Enable 正常中断汇总使能	16	0x0	此位为 1 时：正常中断使能 此位为 0 时：正常中断不使能
AIE:Abnormal Interrupt Summary Enable 非正常中断汇总使能	15	0x0	此位为 1 时：非正常中断使能。 此位为 0 时：非正常中断不使能。
ERE: Early Receive Interrupt Enable 早期接收中断使能	14	0x0	此位为高时：早期接收中断使能
FBE:Fatal Bus Error Enable 总线致命错误中断	13	0x0	此位为高时：总线致命错误中断使能。

使能			
Reserved 保留	12:11	0x0	保留
ETE:Early Transmit Interrupt Enable 早期传输中断使能	10	0x0	此位为高时：使能早期传输中断
RWE:Receive Watchdog Timeout Enable 接收看门狗超时中 断使能	9	0x0	此位为高时：使能接收看门狗超时中断
RSE:Receive Stopped Enable 接收停止中断使能	8	0x0	此位为高时：使能接收停止中断。
RUE:Receive Buffer Unavailable Enable 接收缓冲区不可用 中断使能	7	0x0	此位为高时：使能接收缓冲区不可用中断。
RIE:Receive Interrupt Enable 接收中断使能	6	0x0	此位为高时：使能接收完成中断
UNE:Underflow Interrupt Enable 传输 FIFO 下溢中断 使能	5	0x0	此位为高时：使能传输 FIFO 下溢中断
OVE:Overflow Interrupt Enable 接收 FIFO 上溢中断 使能	4	0x0	此位为高时：使能接收 FIFO 上溢中断。
TJE:Transmit Jabber Timeout Enable 传输 Jabber 超时中	3	0x0	此位为高时：使能 Jabber 超时中断。

断使能			
TUE:Transmit Buffer Unavailable Enable 传输缓存不可用中断使能	2	0x0	此位为高时：使能传输缓存不可用中断。
TSE:Transmit Stopped Enable 传输停止中断使能	1	0x0	此位为高时：使能传输停止中断。
TIE:Transmit Interrupt Enable 传输完成中断使能	0	0x0	此位为高时：使能传输完成中断。

### 11.1.9 Register8 (Missed Frame and Buffer Overflow Counter Register)

Register8 (Missed Frame and Buffer Overflow Counter Register) Offset: 0x20			
Reserved 保留	31:29	0x0	保留
Overflow bit for FIFO Overflow Counter FIFO 溢出指示位	28	0x0	FIFO 溢出指示位
Indicates the number of frames missed by the application 应用程序丢失的帧个数	27:17	0x0	指示应用程序丢失帧的个数
Overflow bit for Missed Frame Counter 丢失帧个数溢出指示	16	0x0	提示丢失帧个数已经超过计数的最大值。
Indicates the number of frames missed by the	15:0	0x0	指示因为主机接收缓存不可用导致帧丢失个数的计数。

controller due to the Host Receive Buffer being unavailable 因为主机接收缓存不可用导致帧丢失的个数			
--	--	--	--

### 11.1.10 Register9(Receive Interrupt Watchdog Timer Register)

Register9(Receive Interrupt Watchdog Timer Register)    Offset: 0x24			
Reserved 保留	31:8	0x0	
RIWT: RI Watchdog Timer count 接收看门狗时间计数	7:0	0x0	当看门狗设置后，表示以时钟周期 x256 的时间为单位计时。当 DMA 接收到数据包且 status 寄存器中 RI 位为 0 时开始计时，当看门狗计数超时时 RI 位置 1。当 RI 位为 1 后该域复位

### 11.1.11 Register10 (AXI Bus Mode Register)

Register10 (AXI Bus Mode Register)    Offset: 0x28			
EN_LPI: Enable LPI (Low Power Interface) 启用低功耗接口	31	0x0	该位为 1 时，启用低功耗接口(Low Power Interface)，系统接受来自 AXI 系统时钟控制器的 LPI 请求。该位为 0 时，不启用低功耗接口，系统永远拒绝 LPI 请求
UNLCK_ON_MGK_RWK: Unlock on Magic Packet or Remote Wake Up 解锁魔法包或远程唤醒包	30	0x0	当该位为 1 时，MAC 处于低功耗(Low Power)状态下时只能通过魔法包或者远程唤醒包来重新回到工作状态；当该位为 0 时，MAC 处于低功耗(Low Power)状态下时只能通过任意包重新回到工作状态
Reserved 保留	29:23	0x0	
WR_OSR_LMT: AXI Maximum Write Outstanding Request Limit	22:20	0x0	该位设置了 AXI 接口写操作发出最大的 outstanding 请求数

AXI 最大 outstanding 写请求			
Reserved 保留	19	0x0	
WR_OSR_LMT: AXI Maximum Read Outstanding Request Limit AXI 最大 outstanding 读请求	18:16	0x0	该位设置了 AXI 接口读操作发出最大的 outstanding 请求数
Reserved 保留	15:13	0x0	
AXI_AAL: Address-Aligned Beats AXI 地址对齐	12	0x0	这位是一个只读位，与 AAL 位 0 相同
Reserved 保留	11:8	0x0	
BLEN256: AXI Burst Length 256 AXI Burst 长度 256	7	0x0	该位只有当 AXI_BURST_LENGTH 配置 256 时有效，为 1 时表示 AXI 支持 Burst 长度为 256 的请求；否则保留
BLEN256: AXI Burst Length 128 AXI Burst 长度 128	6	0x0	该位只有当 AXI_BURST_LENGTH 配置 128 时有效，为 1 时表示 AXI 支持 Burst 长度为 128 的请求；否则保留
BLEN256: AXI Burst Length 64 AXI Burst 长度 64	5	0x0	该位只有当 AXI_BURST_LENGTH 配置 64 时有效，为 1 时表示 AXI 支持 Burst 长度为 64 的请求；否则保留
BLEN256: AXI Burst Length 32 AXI Burst 长度 32	4	0x0	该位只有当 AXI_BURST_LENGTH 配置 32 时有效，为 1 时表示 AXI 支持 Burst 长度为 32 的请求；否则保留
BLEN256: AXI Burst Length 16 AXI Burst 长度 16	3	0x0	该位只有当 AXI_BURST_LENGTH 配置 16 时有效，为 1 时表示 AXI 支持 Burst 长度为 16 的请求；否则保留
BLEN256: AXI Burst	2	0x0	该位只有当 AXI_BURST_LENGTH 配置 8 时有效，

Length 8 AXI Burst 长度 8			为 1 时表示 AXI 支持 Burst 长度为 8 的请求；否则保留
BLEN256: AXI Burst Length 4 AXI Burst 长度 4	1	0x0	该位只有当 AXI_BURST_LENGTH 配置 4 时有效，为 1 时表示 AXI 支持 Burst 长度为 4 的请求；否则保留
UNDEF: AXI Undefined Burst Length 为定义 AXI Burst 长度	0	0x0	该位为 1 时表示 AXI 接口可以发出任意 Burst 长度的请求，为 0 时表示 AXI 支持以上 bit7-1 定义的长度为 256/128/64/32/16/8/4 或者 1 的请求

### 11.1.12 Register11 (AXI Status Register)

Register11 (AXI Status Register)		Offset: 0x2C	
Reserved 保留	31:2	0x0	
	1	0x0	该为为 1 时表示当前 AXI 正在发送读请求
	0	0x0	该为为 1 时表示当前 AXI 正在发送写请求

Register12- Register18: **Reserved**

### 11.1.13 Register18 (Current Host Transmit Descriptor Register)

Register18 (Current Host Transmit Descriptor Register)		Offset: 0x48	
Host Transmit Descriptor Address Pointer 当前发送描述符主机地址指针	31:0	0x0	只读

### 11.1.14 Register19 (Current Host Receive Descriptor Register)

Register19 (Current Host Receive Descriptor Register)		Offset: 0x4C	
Host Receive Descriptor Address Pointer 当前接收描述符主机地址指针	31:0	0x0	只读

### 11.1.15 Register20 (Current Host Transmit Buffer Address Register)

Register20 (Current Host Transmit Buffer Address Register) Offset: 0x50				
Host Transmit Buffer Address Pointer	31:0	0x0	只读	
当前传输缓冲区主机地址指针				

### 11.1.16 Register21 (Current Host Receive Buffer Address Register)

Register21 (Current Host Receive Buffer Address Register) Offset: 0x54				
Host Receive Buffer Address Pointer	31:0	0x0	只读	
当前接收缓冲区主机地址指针				

## 11.2 MAC 控制器寄存器描述

### 11.2.1 Register0 (MAC Configuration Register)

参数名称	位	缺省值	范围	描述
Register0 (MAC Configuration Register) Offset: 0x0000				
Reserved 保留	31:26	0x0	保留	
TC: Transmit Configuration in RGMII 使能 RGMII 链路信息传输	24	0x0		此位为高时，将会把双工模式，链路速度，链路以及链路连接/断开等信息通过 RGMII 接口传输给 PHY。
WD: Watchdog Disable 关闭看门狗	23	0x0		此位为高时，MAC 将关闭接收端的看门狗定时器，可以接收最大 16384 字节的以太网帧。
JD: Jabber Disable 关闭 Jabber 定时器	22	0x0		此位为高时，MAC 关闭发送过程中的 Jabber 定时器，可以发送最大 16384 字节的以太网帧。
BE: Frame Burst Enable	21	0x0		此位为高时，MAC 使能传输过程中的帧突发传输模式。

帧突发传输使能			
JE: Jumbo Frame Enable 巨帧使能	20	0x0	此位为高时，MAC 使能巨帧（最大 9018 字节）的接收。
IFG: Inter-Frame Gap 最小帧间距	19:17	0x0	设置传输过程中的最小帧间距。 000: 96 位时间 001: 88 位时间 010: 80 位时间 ... 111: 40 位时间
DCRS: Disable Carrier Sense During Transmission 传输过程中关闭载波冲突检测	16	0x0	此位为高时，MAC 忽略半双工模式下 CRS 信号的检测。
PS: Port Select 端口选择	15	0x0	0: GMII (1000Mbps) 1: MII (10/100Mbps)
FES: Speed 快速以太网速度提示	14	0x0	0: 10Mbps 1: 100Mbps
DO: Disable Receive Own 关闭接收自己发出的以太网帧	13	0x0	此位为高时，MAC 不接收半双工模式下 gmii_txen_o 有效的以太网帧。
LM: Loopback Mode 使能环回模式	12	0x0	此位为高时，GMII/MII 工作在环回模式下。
DM: Duplex Mode 使能全双工模式	11	0x0	此位为高时，MAC 工作在全双工模式下，在全双工模式下可以同时发送和接收以太网帧。
IPC: Checksum Offload 校验和卸载使能	10	0x0	此位为高时，MAC 硬件计算接收到以太网帧的负载 (payload)。还检查 IPV4 头的校验和是否正确。

DR: Disable Retry 关闭重传	9	0x0	此位为高时，MAC 在遇到冲突时不重传发送冲突的以太网帧，而只报告冲突错误。
LUD: Link Up/Down 链路连接/链路断开	8	0x0	0: 链路断开 1: 链路连接
ACS: Automatic Pad/CRC Stripping 以太网帧 Pad/CRC 自动去除	7	0x0	此位为 1 时，MAC 中去除接收到的以太网帧的 Pad 和 FCS。
BL: Back-Off Limit 回退限制	6:5	0x0	回退限制决定基于 slot 的延迟时间。 00: $k=\min(n,10)$ 01: $k=\min(n,8)$ 10: $k=\min(n,4)$ 11: $k=\min(n,1)$
DC: Deferral Check Deferral 检查	4	0x0	此位为 1 时，使能 deferral 检测功能。
TE: Transmitter Enable 传输使能	3	0x0	此位为 1 时，使能 MAC 传输功能。
RE: Receiver Enable 接收使能	2	0x0	此位为 1 时，使能 MAC 接收功能。
Reserved	1:0	0x0	保留。

### 11.2.2 Register1 (MAC Frame Filter)

Register1 (MAC Frame Filter) Offset: 0x0004			
RA: Receive All 接收全部	31	0x0	此位为 1 时，MAC 接收模块把接收到的所有帧都发给应用程序，忽略源地址/目标地址过滤机制。
Reserved 保留	30:11	0x0	保留
HPF: Hash or Perfect Filter 哈希或者完全过滤	10	0x0	此位为 1 时，在哈希/完全过滤机制中匹配的以太网帧发送给应用。 此位为 0 时，只有在哈希过滤机制中匹配的以太网帧才发送给应用。
SAF: Source	9	0x0	MAC CORE 比较接收到的以太网帧的源地址域和

Address Filter Enable 源地址过滤使能			在 SA 寄存器中的值，如果匹配，接收状态寄存器中的 SAMatch 位设置为高。如果此位为 1，源地址匹配失败，MAC CORE 将丢弃该以太网帧。 如果此位为 0，不管源地址匹配结果 MAC CORE 都接收此帧，而匹配结果写入接收状态寄存器。
SAIF: SA Inverse Filtering 源地址反转过滤	8	0x0	此位为 1 时，和 SA 寄存器中源地址匹配的以太网帧将会标记为源地址匹配失败。 此位为 0 时，和 SA 寄存器中源地址不匹配的以太网帧将会标记为源地址匹配失败。
PCF: Pass Control Frames 接收控制帧	7:6	0x0	00: MAC 过滤所有控制帧 01: MAC 接收除了 pause 帧以外的所有控制帧。 10: MAC 接收所有控制帧。 11: MAC 根据地址过滤情况接收控制帧
DBF: Disable Broadcast Frames 关闭广播帧	5	0x0	此位为 1 时，过滤所有接收的广播帧。 此位为 0 时，接收所有广播帧。
PM: Pass All Multicast 接收所有多播帧	4	0x0	此位为 1 时，接收所有多播帧。 此位为 0 时，过滤所有多播帧。
DAIF: DA Inverse Filtering 目标地址反转过滤	3	0x0	此位为 1 时，对单播和多播帧进行反向目标地址匹配。 此位为 0 时，对单播和多播帧进行正常目标地址匹配。
HMC: Hash Multicast 哈希多播过滤	2	0x0	此位为 1 时，对接收到的多播帧根据哈希表的内容进行目标地址过滤。
HUC: Hash Unicast 哈希单播过滤	1	0x0	此位为 1 时，对接收到的单播帧根据哈希表的内容进行目标地址过滤。
PR: Promiscuous Mode 混杂模式	0	0x0	接收所有以太网帧。

### 11.2.3 Register2 (Hash Table High Register)

Register2 (Hash Table High Register) Offset: 0x0008

HTH: Hash Table High 哈希表高位	31:0	0x0	哈希表的高 32 位。
----------------------------------	------	-----	-------------

#### 11.2.4 Register3 (Hash Table Low Register)

Register3 (Hash Table Low Register) Offset: 0x000C			
HTL: Hash Table Low 哈希表低位	31:0	0x0	哈希表的低 32 位。

#### 11.2.5 Register4 (GMII Address Register)

Register4 (GMII Address Register) Offset: 0x0010			
Reserved 保留	31:16	0x0	保留
PA: Physical Layer Address PHY 地址	15:11	0x0	此域选择需要访问 32 个 PHY 中的哪个。
GR: GMII Register 需要访问的 PHY 设备中的寄存器	10:6	0x0	此域选择需要访问的的 PHY 的哪个 GMII 配置寄存器。
Reserved 保留	5	0x0	保留
CR: CSR Clock Range CSR 时钟范围	4:2	0x0	此域决定 MDC 时钟是 clk_csr_i 时钟频率比例。 0000 clk_csr_i/42 0001 clk_csr_i/62 0010 clk_csr_i/16 0011 clk_csr_i/26 0100 clk_csr_i/102 0101 clk_csr_i/124 0110, 0111 Reserved
GW: GMII Write GMII 写	1	0x0	此位为 1 时, 通过 GMII 数据寄存器对 PHY 进行写操作 此位为 0 时, 通过 GMII 数据寄存器对 PHY 进行读操作。

GB: GMII Busy GMII 忙	0	0x0	对寄存器 4 和寄存器 5 写之前，此位应为 0。在写寄存器 4 之前此位必须先置 0。在访问 PHY 的寄存器时，应用程序需要将此位设置为 1，表示 GMII 接口上有写或者读操作正在进行。
-------------------------	---	-----	--

### 11.2.6 Register5 (GMII Data Register)

Register5 (GMII Data Register) Offset: 0x0014			
Reserved 保留	31:16	0x0	保留
GD: GMII Data GMII 数据	15:0	0x0	此域保存了对 PHY 进行管理读访问操作的 16 位数据，或者对 PHY 进行管理写访问的 16 位数据。

### 11.2.7 Register6 (Flow Control Register)

Register6 (Flow Control Register) Offset: 0x0018			
PT: Pause Time 暂停时间	31:16	0x0	此域保存了需要填入传输控制帧中的暂停时间域。
Reserved 保留	15:8	0x0	保留
DZPQ: Disable Zero-Quanta Pause 禁止零时间片暂停帧	7	0x0	此位为 1 时，禁止自动零时间片的暂停控制帧的产生。
Reserved 保留	6	0x0	保留
PLT: Pause Low Threshold 暂停帧的低阈值	5:4	0x0	此域用于设置暂停时间的阈值。 00: 暂停时间减少 4 个时间槽 01: 暂停时间减少 28 个时间槽 10: 暂停时间减少 144 个时间槽 11: 暂停时间减少 256 个时间槽 (一个时间槽为在 GMII/MII 接口上传输 512 比特或者 64 字节的时间)
UP: Unicast Pause Frame Detect 单播的暂停帧探测	3	0x0	此位为 1 时，MAC 将会根据 MAC 地址 0 指定的本站单播地址来探测暂停帧。
RFE: Receive Flow	2	0x0	此位为 1 时，MAC 将会解析接收到的暂停帧，并且

Control Enable 接收流控使能			按照暂停帧指定的时间暂停帧的发送。
TEF: Transmit Flow Control Enable 发送流控使能	1	0x0	在全双工模式下，此位为 1 时，MAC 使能暂停帧的发送。 在半双工模式下，此位为 1 时，MAC 使能反压操作。
FCB/BPA: Flow Control Busy/Backpressure Activate 流控忙/反压激活	0	0x0	此位为 1 时，在全双工模式下发起暂停控制帧的发送或在半双工模式下启动反压操作。

### 11.2.8 Register7 (VLAN Tag Register)

Register7 (VLAN Tag Register) Offset: 0x001C			
Reserved 保留	31:17	0x0	保留
ETV: Enable 12-Bit VLAN Tag Comparison 使能 12 位 VLAN Tag 比较	16	0x0	此位为 1 时，使用 12 位 VLAN Tag 而不是使用 16 位 VLAN Tag 用于以太网帧比较和过滤。
VL: VLAN Tag Identifier for Receive Frames 帧接收的 VLAN Tag 标识	15:0	0x0	此域保存 802.1Q 格式的 VLAN Tag，用于比较接收到的以太网帧的位于第 15 和第 16 个字节的 VLAN Tag。

### 11.2.9 Register8 (Version Register)

Register8 (Version Register) Offset: 0x0020			
Reserved 保留	15:8	0x0	保留
Version 版本号	7:0	0x0	0X35

### 11.2.10 Register14 (Interrupt Status Register)

Register14 (Interrupt Status Register) Offset: 0x0038			
Reserved	15:8	0x0	保留

保留			
MMC Receive Checksum Offload Interrupt Status MMC 接收校验和卸 载状态中断	7	0x0	MMC 校验和卸载寄存器产生任何中断产生时，此位 设置为 1。
MMC Transmit Interrupt Status MMC 传输中断	6	0x0	MMC 传输中断寄存器产生任何中断时，此位设置为 1。
MMC Receive Interrupt Status MMC 接收中断状态	5	0x0	MMC 接收中断寄存器产生任何中断时，此位设置为 1。
MMC Interrupt Status MMC 中断状态	4	0x0	7:5 的任何位为高时，此位设置为 1。
PMT Interrupt Status 电源管理中断状态	3	0x0	在 Power Down 状态下，收到 magic 帧或在 Wake-on-LAN 帧时，此位设置为 1。
PCS Auto-Negotiation Complete PCS 自动协商完成	2	0x0	RGMII PHY 接口自动协商完成时，此位设置为 1。
PCS Link Status Changed PCS 链路状态变化	1	0x0	RGMII PHY 接口的链路状态发生任何变化时，此位 设置为 1。
RGMII Interrupt Status RGMII 中断状态	0	0x0	RGMII 接口的链路状态发生任何变化时，此位设置 为 1。

### 11.2.11 Register15 (Interrupt Mask Register)

Register15 (Interrupt Mask Register) Offset: 0x003C			
Reserved 保留	15:10	0x0	保留
Time Stamp	9	0x0	此位为 1 时，禁止时间戳发生的中断

Interrupt Mask 时间戳中断使能			
Reserved 保留	8:4	0x0	保留
PMT Interrupt Mask 电源管理中断使能	3	0x0	此位为 1 时，禁止电源管理引起的中断。
PCS AN Completion Interrupt Mask PCS 自动协商完成 中断使能	2	0x0	此位为 1 时，禁止 PCS 自动协商完成中断。
PCS Link Status Interrupt Mask PCS 链路状态中断 使能	1	0x0	此位为 1 时，禁止由于 PCS 链路状态变化引起的中 断。
RGMII Interrupt Mask RGMII 中断使能	0	0x0	此位为 1 时，禁止 RGMII 引起的中断。

### 11.2.12 Register16 (MAC Address0 High Register)

Register16 (MAC Address0 High Register) Offset: 0x0040			
MO: Always 1 保留	31	0x0	保留
Reserved 保留	30:16	0x0	保留
MAC Address0[47:32] MAC 地址高 16 位	15:0	0x0	存放用于接收地址过滤和传输流控帧的 MAC 地址。

### 11.2.13 Register17 (MAC Address0 Low Register)

Register17 (MAC Address0 Low Register) Offset: 0x0044			
MAC Address0[31:0] MAC 地址低 32 位	31:0	0x0	存放用于接收地址过滤和传输流控帧的 MAC 地址。

### 11.2.14 Register18 (MAC Address1 High Register)

Register18 (MAC Address1 High Register) Offset: 0x0048			
AE: Address Enable 地址使能	31	0x0	此位为 1 时，地址过滤模块使用第 2 个 MAC 地址用于完全地址过滤。此位为 0 时，地址过滤模块不使用第 2 个 MAC 地址用于地址过滤。
SA: Source Address 源 MAC 地址	30	0x0	此位为 1 时，MAC 地址 1 用于比较接收帧的源 MAC 地址。 此位为 0 时，MAC 地址 1 用于比较接收帧的目标 MAC 地址。
MBC: Mask Byte Control 掩模字节控制	29:24	0x0	此域用于比较每个 MAC 地址的字节掩模控制位。比如第 29 位用于掩码寄存器 18 的[15:8]这个字节。
Reserved 保留	23:16	0x0	保留。
MAC Address1[47:32] 第 2 个 MAC 地址的高 16 位	15:0	0xFFFF	

### 11.2.15 Register19 (MAC Address1 Low Register)

Register19 (MAC Address1 Low Register) Offset: 0x004C			
MAC Address1[31:0] 第 2 个 MAC 地址的低 32 位	31:0	0x0	

### 11.2.16 Register48 (AN Control Register)

Register48 (AN Control Register) Offset: 0x00C0			
Reserved 保留	31:19	0x0	保留
SGMII RAL Control 保留	18	0x0	保留
LR: Lock to Reference 锁定到参考时钟	17	0x0	此位为 1 时，PHY 将其锁相环锁定到 125MHz 的参考时钟。

ECD: Enable Comma Detect 使能停顿探测	16	0x0	此位为 1 时，使能 PHY 的停顿探测和字重同步。
Reserved 保留	15	0x0	保留
ELE: External Loopback Enable 外部环回使能	14	0x0	此位为 1 时，使能 PHY 进入环回模式。
Reserved 保留	13	0x0	保留
ANE: Auto-Negotiation Enable 自动协商使能	12	0x0	此位为 1 时，MAC 将会和链路对方进行自动协商。
Reserved 保留	11:10	0x0	保留
RAN: Restart Auto-Negotiation 重新进行自动协商	9	0x0	此位为 1 时，重新进行自动协商。
Reserved 保留	8:0	0x0	保留

### 11.2.17 Register49 (AN Status Register)

Register49 (AN Status Register) Offset: 0x00C4			
Reserved 保留	31:9	0x0	保留
ES: Extended Status 扩展状态	8	0x0	只读，因为 MAC 支持扩展状态信息。
Reserved 保留	7:6	0x0	保留
ANC: Auto-Negotiation Complete	5	0x0	只读，指示自动协商完成。

自动协商完成			
Reserved 保留	4	0x0	保留
ANA: Auto-Negotiation Ability 自动协商能力	3	0x0	只读，因为 MAC 支持自动协商。
LS: Link Status 链路状态	2	0x0	此位为 1 时，指示链路连接上。 此位为 0 时，指示链接未连接。
Reserved 保留	1:0	0x0	保留。

### 11.2.18 Register50 (Auto-Negotiation Advertisement Register)

Register50 (Auto-Negotiation Advertisement Register) Offset: 0x00C8			
Reserved 保留	31:16	0x0	保留
NP: Next Page Support 下一页面支持	15	0x0	只读为 0，因为 MAC 不支持下一页面。
Reserved 保留	14	0x0	保留
RFE: Remote Fault Encoding 远端错误编码	13:12	0x0	此 2 位指示链路对端发生错误，具体编码将 IEEE 802.3z 第 37.2.1.5 小节。
Reserved 保留	11:9	0x0	保留
PSE: Pause Encoding Pause 位编码	8:7	0x0	见 IEEE 802.3z 第 37.2.1.4 小节
HD: Half-Duplex 半双工	6	0x0	此位为 1 时，指示 MAC 支持半双工。
FD: Full-Duplex 全双工	5	0x0	此位为 1 时，指示 MAC 支持全双工。

Reserved 保留	4:0	0x0	保留
----------------	-----	-----	----

### 11.2.19 Register51 (Auto-Negotiation Link Partner Ability Register)

Register51 (Auto-Negotiation Link Partner Ability Register) Offset: 0x00CC			
Reserved 保留	31:16	0x0	保留
NP: Next Page Support 下一页面支持	15	0x0	此位为 1 时，指示有更多下一页面信息可用 此位为 0 时，指示下一页面交换不可用。
ACK: Acknowledge 确认	14	0x0	指示在自动协商中，链路对端成功接收到 MAC 的基本页面。
RFE: Remote Fault Encoding 远端错误编码	13:12	0x0	见 IEEE 802.3z 第 37.2.1.5 小节。
Reserved 保留	11:9	0x0	保留
PSE: Pause Encoding 对端 pause 状态编码	8:7	0x0	见 IEEE 802.3z 第 37.2.14 小节。
HD: Half-Duplex 半双工	6	0x0	指示对端可以运行在半双工模式。
FD: Full-Duplex 全双工	5	0x0	指示对端可以运行在全双工模式。
Reserved 保留	4:0	0x0	保留

### 11.2.20 Register52 (Auto-Negotiation Expansion Register)

Register52 (Auto-Negotiation Expansion Register) Offset: 0x00D0			
Reserved 保留	31:3	0x0	保留
NPA: Next Page Ability	2	0x0	只读为 0，因为 MAC 不支持下一页面。

下一页面能力			
NPR: New Page Received 接收到新页面	1	0x0	此位为 1 时, 指示 MAC 接收到新页面。
Reserved 保留	0	0x0	保留

### 11.2.21 Register54 (SGMII/RGMII Status Register)

Register54 (SGMII/RGMII Status Register) Offset: 0x00D8			
Reserved 保留	31:4	0x0	保留
Link Status 链路状态	3	0x0	此位为 1 时, 指示链路连接上。 此位为 0 时, 指示链路未连接上。
Link Speed 链路速度	2:1	0x0	指示链路当前速度 00: 2.5MHz 01: 25MHz 10: 125MHz
Link Mode 链路模式	0	0x0	0: 半双工 1: 全双工

### IEEE1588 寄存器:

### 11.2.22 Register448 (Time Stamp Control Register)

Register448 (Time Stamp Control Register) Offset: 0x0700			
Reserved 保留	31:20	0x0	保留
ATSFC: Auxiliary Snapshot FIFO Clear Auxiliary 快照 FIFO 清除	19	0x0	该位为 1 时重置 Auxiliary Snapshot FIFO 的指针, FIFO 清空, 完成后为 0
TSENMACADDR: Enable MAC address for PTP frame filtering 启用 PTP 帧的 MAC	18	0x0	此位为 1 时, 接收到的 PTP 包会经过 MAC 的 MAC 目的地址过滤器(DA Fiter), 不匹配的 PTP 包会被丢弃

地址过滤			
TSCLKTYPE: Select the type of clock node 选择时间节点类型	17:16	0x0	该域按照以下类型编码： 00: 普通时钟(Ordinary clock) 01: 边界时钟(Boundary clock) 10: 端到端的透明时钟(End-to-End Transparent clock) 11: 点到点透明时钟(Peer-to-peer Transparent clock)
TSMSTRENA: Enable Snapshot for Messages Relevant to Master 启用主机相关信息快照	15	0x0	该位为 1 时快照(Snapshot)记录主机相关信息(Relevant to Master); 为 0 时快照记录从机相关信息(Relevant to Slave)
TSEVNTENA: Enable Time Stamp Snapshot for Event Messages 启用事件信息时间戳快照	14	0x0	该位为 1 时快照(Snapshot)只记录事件相关信息(Event Message); 为 0 时快照记录所有信息
TSIPV4ENA: Enable Time Stamp Snapshot for IPv4 frames 启用 IPv4 时间戳快照	13	0x0	该位为 1 时, IPv4 帧的时间戳快照(Time Stamp Snapshot)启用
TSIPV6ENA: Enable Time Stamp Snapshot for IPv6 frames 启用 IPv6 时间戳快照	12	0x0	该位为 1 时, IPv6 帧的时间戳快照(Time Stamp Snapshot)启用
TSIPENA: Enable Time Stamp Snapshot for PTP	11	0x0	该位为 1 时, 以太网格式的 PTP 包(PTP over Ethernet)启用时间戳快照(Time Stamp Snapshot); 为 0 时 UDP-IP-Ethernet 格式的 PTP 包启用时间戳

over Ethernet frames 启用以太网 PTP 包时间戳快照			快照
TSVER2ENA: Enable PTP packet snooping for version 2 format 基于第二版本格式启用 PTP 包快照	10	0x0	该位为 1 时，以太网格式的 PTP 包快照为第二版本格式(version 2 format); 为 0 时为第一版本格式(version 1 format)
TSCTRLSSR: Time Stamp Digital or Binary rollover control 时间戳数字或二进制更新控制	9	0x0	该位为 1 时，时间戳低位寄存器(Time Stamp Low register)在值超过 0x3B9A_C9FF 后更新且增加时间戳高位寄存器的秒; 该为为 0 时，时间戳低位寄存器复位为 0x7FFF_FFFF
TSEALL: Enable Time Stamp for All Frames 对所有帧启用时间戳	8	0x0	该为为 1 时，表示对所有 MAC 接收到的帧启用时间戳
Reserved 保留	7:6	0x0	保留
TSADDREG: Addend Reg Update 被加数寄存器	5	0x0	该位为 1 时，时间戳被加数寄存器(Time Stamp Addend register)的值被用来更新 PTP 块来做 fine correction. 当更新结束时该位为 0
TSTRIG: Time Stamp Interrupt Trigger Enable 启用时间戳中断触发	4	0x0	该位为 1 时，当系统时间大于目标时间寄存器(Target Time register)中的值时，产生一个时间戳中断; 中断产生后该位重置为 0
TSUPDT: Time Stamp Update 时间戳更新	3	0x0	该位为 1 时，系统根据时间戳更新高位寄存器和时间戳低位寄存器的值(Time Stamp High Update and Time Stamp Low Update registers)更新当前系统时间，完成后该位为 0
TSINIT: Time Stamp Initialize	2	0x0	该位为 1 时，系统根据时间戳更新高位寄存器和时间戳低位寄存器的值(Time Stamp High Update and

时间戳初始化			Time Stamp Low Update registers)初始化当前系统时间，完成后该位为 0
TSCFUPDT: Time Stamp Fine or Coarse Update 时间戳 Fine or Coarse 更新	1	0x0	该位为 1 时，表示时间戳更新将会用 Fine Update 的方式，为 0 时表示时间戳更新将会用 Coarse 的方式
TSENA: Time Stamp Enable 时间戳启用	1	0x0	该位为 1 时，表示当前对于接收和发送帧系统将会启用时间戳记录；为 0 时表示当前停止向接收和发送帧进行时间戳记录，并且暂停时间戳发生器；该为置 1 后应该接着进行时间戳初始化(bit 2)

### 11.2.23 Register449 (Sub-Second Increment Register)

Register449 (Sub-Second Increment Register) Offset: 0x0704			
Reserved 保留	31:8	0x0	保留
SSINC: Sub-second increment value 亚秒增加值	7:0	0x0	此域中的值将会被加到亚秒寄存器(sub-second register)

### 11.2.24 Register 450 (System Time - Seconds Register)

Register 450 (System Time - Seconds Register) Offset: 0x0708			
TSS: Time Stamp Second 时间戳秒	31:0	0x0	该域的值表示当前系统时间的秒级计数

### 11.2.25 Register 451 (System Time - Nanoseconds Register)

Register 451 (System Time - Nanoseconds Register) Offset: 0x070C			
PSNT: Positive or Negative Time 正或负时间	31	0x0	该域表示正或负的时间，当为 1 时表示时间为正数，为 0 时表示为负数
TSSS: Time Stamp Sub Seconds 时间戳亚秒	30:0	0x0	该域的值表示当前系统时间的亚秒级计数，默认时间精度为 0.46 纳秒(当 TSCTRLSSR 置起时精度为 1ns，最大值为 0x3B9A_C9FF)

### 11.2.26 Register 452 (System Time - Seconds Update Register)

Register 452 (System Time - Seconds Update Register) Offset: 0x0710			
TSS: Time Stamp	31:0	0x0	该域的值在系统初始化时的初值，或是更新时的增加

Second 时间戳秒			值
----------------	--	--	---

### 11.2.27 Register 453 (System Time - Nanoseconds Update Register)

Register 453 (System Time - Nanoseconds Update Register) Offset: 0x0714			
ADDSUB: Add or subtract time 加或减时间	31	0x0	该域表示加或减时间, 当为 1 时表示更新时减去更新寄存器的值, 为 0 时表示增加更新寄存器的值
TSSS: Time Stamp Sub Seconds 时间戳亚秒	30:0	0x0	该域的值表示当前系统时间更新时增加或减少的亚秒级计数, 默认时间精度为 0.46 纳秒(当 TSCTRLSSR 置起时精度为 1ns, 最大值为 0x3B9A_C9FF)

### 11.2.28 Register 454 (Time Stamp Addend Register)

Register 454 (Time Stamp Addend Register) Offset: 0x0718			
TSAR: Time Stamp Addend Register 时间戳加数寄存器	31:0	0x0	该域的值表示在系统时间同步时需要被增加的 32 位时间值

### 11.2.29 Register 455 (Target Time Seconds Register)

Register 455 (Target Time Seconds Register) Offset: 0x071C			
TSTR: Target Time Seconds Register 目标时间秒寄存器	31:0	0x0	该域保存了以秒为单位的时间, 当时间戳时间匹配或超过了该域的值及低位寄存器(reg 456)的值且包 MAC 地址匹配, 系统会产生一个中断(中断必须事先启用)

### 11.2.30 Register 456 (Target Time Nanoseconds Register)

Register 456 (Target Time Nanoseconds Register) Offset: 0x0720			
Reserve 保留	31	0x0	
TSTR: Target Time Stamp Low Register 目标时间戳低寄存器	30:0	0x0	该域保存了以纳秒为单位的时间, 当时间戳时间匹配或超过了该域的值及目标时间寄存器(reg 455)的值且包 MAC 地址匹配, 系统会产生一个中断(中断必须事先启用)

### 11.2.31 Register 457 (System Time - Higher Word Seconds Register)

Register 457 (System Time - Higher Word Seconds Register) Offset: 0x0724			
Reserve 保留	31:16	0x0	

TSHWR: Time Stamp Higher Word Register 时间戳高位寄存器	15:0	0x0	该域保存高 16 位(most significant bit)的时间戳值
--	------	-----	---------------------------------------

### 11.2.32 Register 458 (Time Stamp Status Register)

Register 458 (Time Stamp Status Register) Offset: 0x0728			
Reserve 保留	31:28	0x0	
ATSNS: Auxiliary Time Stamp Number of Snapshots 附属时间戳快照个数	27:25	0x0	该域表示了 FIFO 中快照的个数, 4(3'b100)表示 FIFO 满, 0(3'000)表示 FIFO 空
ATSSTM: Auxiliary Time Stamp Snapshot Trigger Missed 附属时间戳快照触发缺失	24	0x0	该位为 1 时表示当前 FIFO 满且设置了外部触发, 表示最后一个快照因为 FIFO 满被丢失了; 否则为 0
Reserve 保留	23:3	0x0	
Auxiliary Time Stamp Snapshot Trigger 附属时间戳快照触发	2	0x0	该位为 1 时表示有快照(snapshot)写入了 FIFO
TSTARGET: Time Stamp Target Time Reached 达到时间戳目标时间	1	0x0	该位为 1 时表示系统时间已经大于或等于目标时间戳寄存器中的值
TSSOVF: Time Stamp Seconds Overflow 时间戳秒溢出	0	0x0	该位为 1 时表示当前时间戳的时间已经溢出(大于 0xFFFF_FFFF); 反之为 0

### 11.2.33 Register 459 (PPS Control Register)

Register 459 (PPS Control Register) Offset: 0x072C			
--	--	--	--

Reserve 保留	31:4	0x0	
PPSCTRL: Control the duration between 2-pulses of PPS signal output 2 个 PPS 信号输出的时间间隔控制	15:0	0x0	该域设置 2 个 PPS 信号输出的时间间隔, 编码如下: 0000: 1.0 秒(Binary & Digital Rollover) 0001: 0.5 秒(Binary Rollover), 0.536 秒(Digital Rollover) 0010: 0.25 秒(Binary Rollover), 0.26 秒(Digital Rollover) 0011: 0.125 秒(Binary Rollover), 0.13 秒(Digital Rollover) ... 1111: 15.28 μ 秒(Binary Rollover), 32.77 μ 秒(Digital Rollover)

#### 11.2.34 Register 460 (PPS Auxiliary Time Stamp - Nanoseconds Register)

Register 460 (PPS Auxiliary Time Stamp - Nanoseconds Register) Offset: 0x0730			
	31:0	0x0	包含了附属时间戳(Auxiliary Time Stamp)的低(纳秒)32 位值

#### 11.2.35 Register 461 (PPS Auxiliary Time Stamp - Seconds Register)

Register 460 (PPS Auxiliary Time Stamp - Seconds Register) Offset: 0x0730			
	31:0	0x0	包含了附属时间戳(Auxiliary Time Stamp)的高(秒)32 位值

### 11.3 DMA 描述符

DMA 描述符是 MAC 驱动和硬件的交互接口, 记录了数据包的内存地址和传输状态。在此分别定义了发送描述符(Tx Descriptor)和接收描述符(Rx Descriptor)两种数据结构。两种描述符可以自由选择分别以环式(ring mode)或者链式(chain mode)相连, 以供 MAC 使用。

#### 11.3.1 DMA 描述符的基本格式

每一个 DMA 描述符包含两个数据 buffer、两个字节计数 buffer 和两个指向数据 buffer 地址的指针。需要注意的是描述符的地址必须保证按照所连接的系统总线位宽对齐, 同时保证与系统字节序相同(默认小尾端)。

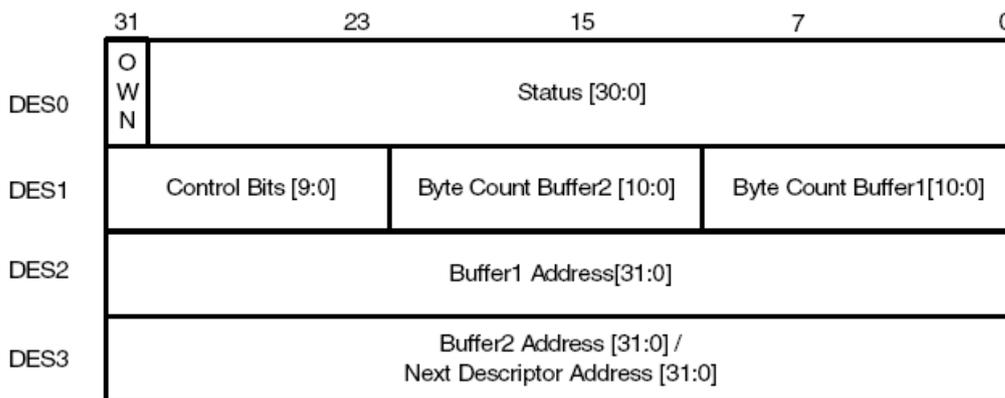


图 11-1 DMA 描述符的基本格式（小尾端 32 位总线）

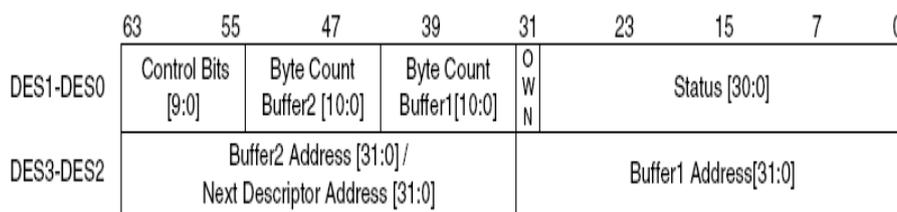


图 11-2 DMA 描述符的基本格式（小尾端 64 位总线）

### 11.3.2 DMA 接收描述符

GMAC 子系统在工作模式下需要至少两个接收描述符才能够正常的接收一个网络数据包。其内部的接收模块在处理一个网络数据包时，总是在同时尝试获取下一个接收描述符。每一个网络数据包被称为一个帧(frame)。

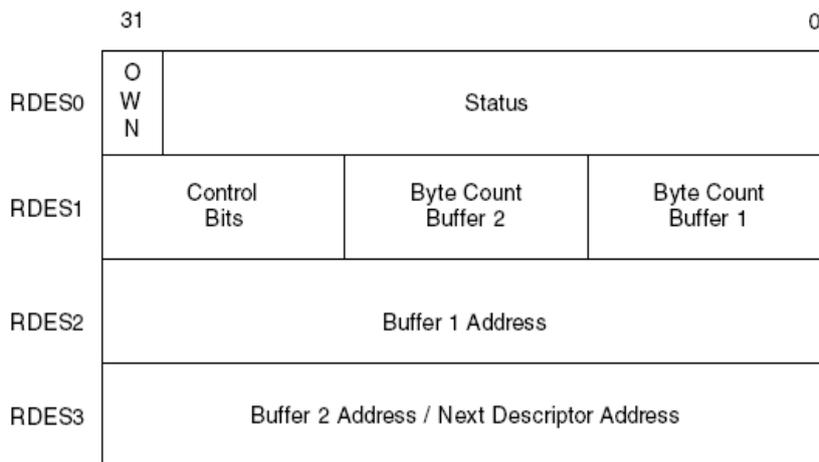


图 11-3 DMA 接收描述符的基本格式(小尾端 32 位总线)

#### RDES0

RDES0 包括了当前接收帧状态、长度以及该描述符的所有情况(主机或 DMA

拥有)。RDES0 的具体细节参见下表。

RDES0	位	
OWN 所有模式	31	该位为 1 时表示描述符当前属于 DMA 控制，0 表示属于主机控制。当 DMA 模块完成一次传输时，会将该位主动清 0
AFM: Destination Address Filter Fail 目标地址过滤错误	30	当该位为 1 时，表示当前数据帧目标地址不符合 GMAC 内部的帧目标地址过滤器
FR: Frame length 帧长度	29:16	表示接收当前帧的长度，当 ES 位为 0 时有效
ES: Error Summary 总体错误信息	15	指示当前帧是否出错，其值为 RDES[0]、RDES[1]、RDES[3]、RDES[4]、RDES[6]、RDES[7]、RDES[11]、RDES[14] 各位作或运算(OR)的结果
DE: Descriptor Error 描述符错误	14	当该位为 1 时表示，当前描述符所指向的 buffer 与帧不相符或者 OWN 为 0(主机控制)
SAF: Source Address Filter Fail 源地址过滤错误	13	当该位为 1 时，表示当前数据帧的源地址不符合 GMAC 内部的帧源地址过滤器
LE: Length Error 长度错误	12	当该位为 1 时，表示当前接收帧长度与默认长度不符。当 Frame Type 位为 1 且 CRC Error 位为 0 时有效
OE: Over Flow Error 溢出错误	11	当该位为 1 时，表示接收该帧时 GMAC 内部 Rx FIFO 溢出
VLAN: VLAN Tag VLAN 标志	10	当该位为 1 时，表示该帧的类型为 VLAN
FS: First Descriptor 第一个描述符	9	当该位为 1 时，表示当前描述符所指向的 buffer 为当前接收帧的第一个保存 buffer
LS: Last Descriptor 最后一个描述符	8	当该位为 1 时，表示当前描述符所指向的 buffer 为当前接收帧的最后一个保存 buffer
IPC Checksum Error/Giant Frame	7	当该位为 1 时，如果 IPC 校验功能启用则表示当前帧的 IPv4 头校验值与帧内部校验域的值不相符。如果未启用则表示当

校验错误/超长帧		前帧为一个超长帧(长度大于 1518 字节)
LC: late collision 后期冲突	6	当该位为 1 时, 表示在半双工模式下, 当前帧接收时发生了一个后期冲突
FT: Frame Type 帧类型	5	当该位为 1 时, 表示当前帧为一个以太网格式帧, 为 0 时表示当前帧为一个 IEEE802.3 格式帧
RWT: Receive Watchdog Timeout	4	当该位为 1 时, 表示当前时钟值超过了接收模块看门狗电路时钟的值, 既接收帧超时
RE: Receive Error 接收错误	3	当该位为 1 时, 表示接收当前帧时内部模块出错。内部信号 rxer 置 1 且 rxdv 置 1
DE: Dribble bit Error 奇数位错误	2	当该位为 1 时, 表示接收帧长度不是整数, 即总长度为奇数位, 该位只有在 mii 模式下有效
CE: CRC Error 接收 CRC 校验错误	1	当该位为 1 时, 表示接收当前帧时内部 CRC 校验出错。该位只有在 last descriptor(RDES0[8])为 1 时有效
RX MAC: Checksum/payload Checksum Error 接受校验/负载校验 错误	0	当该位为 1 时, 表示接收当前帧时内部 RX MAC 寄存器组 1-15 中存在一个匹配当前帧目的地址。为 0 时表示 RX MAC 寄存器组 0 匹配接受帧目的地址。如果 Full Checksum Offload Engine 启用时, 为 1 表示该帧 TCP/UDP/ICMP 校验错误。该位为 1 时也可能表示当前帧实际接受长度与帧内部记载长度不相符。

## RDES1

RDES1 记录了描述符所指向的 buffer 大小, 以及描述符的组织格式(环形或链型)。

RDES1	位	
Disable Intr in Completion 禁止完成后发中断	31	该位为 1 时表示该帧接收完成后将不会置起 STATUS 寄存器中 RI 位 (CSR5[6]), 这将会使得主机无法检测到该中断
Reserved 保留	30:26	
RER: Receive End of Ring 环型描述符结尾	25	该位为 1 时表示该描述符为环型描述符链表的最后一个, 下一个描述符的地址为接收描述符链的基址
RCH: Second	24	该位为 1 时表示描述符中的第二个 buffer 地址指向的是下一

Address Chained 第二个buffer地址指向下一个链式描述符		个描述符的地址，为0时表示该地址指向第二个buffer地址 当该位为1时，RDES1[21-11]的值将没有意义，RDES1[25]比RDES1[24]具有更高优先级(代表环型而不是链型)
Reserved 保留	23:22	
RBS2: Receive Buffer Size 2 接收buffer2大小	21:11	该域表示数据buffer2的大小。根据系统总线的宽度32/64/128，Buffer2的大小应该为4/8/16的整数倍。如果不满足则会导致未知的结果。该域在RDES1[24]为0时有效
RBS2: Receive Buffer Size 1 接收buffer1大小	10:0	该域表示数据buffer1的大小。根据系统总线的宽度32/64/128，Buffer1的大小应该为4/8/16的整数倍。如果不满足则会导致未知的结果。该域一直有效。如果该域值为0，DMA则会访问buffer2或者下一个接收描述符

## RDES2

该域记录了数据接收buffer1的地址。

RDES2		位
Buffer1 Address Pointer 接收buffer1地址	31:0	该域记录了数据接收buffer1的32位物理地址。该物理地址没有默认的对齐要求。当GMAC DMA内部实现了总线数据32/64/128位对齐，则该地址的低2/3/4位会被忽略

## RDES3

该域记录了数据接收buffer2的地址。

RDES3		位
Buffer2 Address Pointer 接收buffer2地址	31:0	该域记录了数据接收buffer2的32位物理地址。该物理地址没有默认的对齐要求。当GMAC DMA内部实现了总线数据32/64/128位对齐，则该地址的低2/3/4位会被忽略。如果描述符是以链式连接，则该域记录的是下一个描述符的地址

### 11.3.3 DMA 发送描述符

发送描述符与接收描述符的格式基本相同。每个描述符的地址需要按照总线宽度(32/64/126位)对齐。

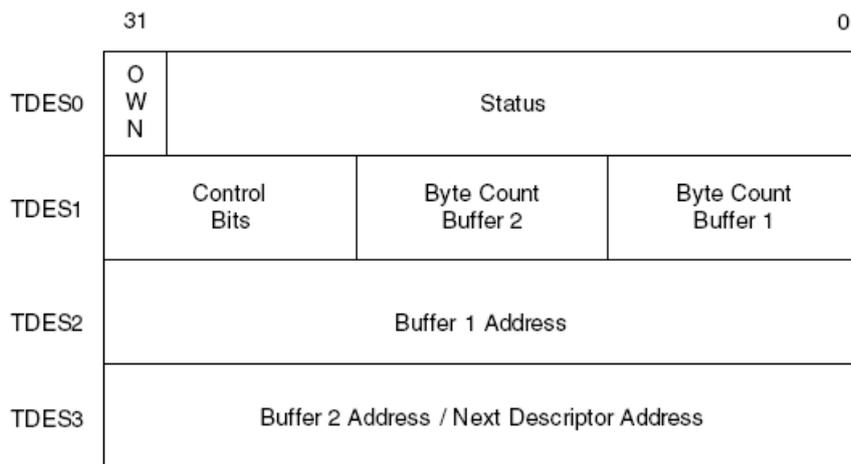


图 11-4 DMA 发送描述符的基本格式(小尾端 32 位总线)

### TDES0

TDES0 包含了发送帧的状态和发送描述符的所属信息。

TDES0	位	
OWN 所属模式	31	该位为 1 时表示描述符当前属于 DMA 控制，0 表示属于主机控制。当 DMA 模块完成一次传输时，会将该位主动清 0
Reserved 保留	30:18	
TTSS: Tx Time Stamp Status 发送时间戳状态	17	当 IEEE1588 功能启用时，该位为 1 表示 TDES2 和 TDES3 中保存了该发送帧的时间戳信息。否则该位保留
IHE: IP Header Error IP 头错误	16	该位为 1 时表示内部校验模块发现该发送帧的 IP 头出错，并且不会对该域做任何修改
ES: Error Summary 总体错误信息	15	指示当前帧是否出错，其值为 TDES[1]、TDES[2]、TDES[8]、TDES[9]、TDES[10]、TDES[11]、TDES[13]、TDES[14] 各位作或运算(OR)的结果
JT: Jabber Timeout Jabber 超时	14	该位为 1 时表示 GMAC 发送模块遇到了 Jabber 超时
FF: Frame Flushed 帧刷新	13	该位为 1 时表示软件发出了一个刷新命令导致 DMA/MTL 将其内部的帧刷新掉
PCE: Payload Checksum Error	12	该位为 1 时表示内部负载校验模块再向发送帧中插入校验数据时出错。当负载校验模块启用时，该位有效

负载校验错误		
LC: Loss of Carrier 载波丢失	11	该位为 1 时表示在发送该帧过程中载波丢失(gmii_crs 信号多个周期未置起)
NC: No Carrier 载波无效	10	该位为 1 时表示在发送过程中，PHY 的载波信号一直未置起
LC: Late Collision 后期冲突	9	当该位为 1 时表示在半双工模式下，当前帧接收时发生了一个后期冲突
EC: Excessive Collision 连续冲突	8	当该位为 1 时表示在发送当前帧的时候连续出现了 16 次冲突
VF: VLAN Frame VLAN 帧	7	该位为 1 时表示当前发送帧为一个 VLAN 帧
CC: Collision Count 冲突计数	6:3	该域表示当前帧在成功发送之前所遇到冲突次数的总数
ED: Excessive Deferral 连续 Deferral	2	该位为 1 时表示当前帧传输结束
UF: Underflow Error 溢出错误	1	该位为 1 时表示当前帧传输时发生了溢出错误，即数据传输 buffer 过小或不可用
DB: Defered Bit 帧刷新	0	该位为 1 时表示此次发送被延迟，只有在半双工模式下有效

## TDES1

TDES1 包含了 buffer 大小以及其他一些控制描述符环型/链型连接的控制和状态位。

TDES1	位	
IC: Interrption on Complete 完成时中断	31	该位为 1 时表示该帧接发送完成后将会置起 STATUS 寄存器中 TI 位(CSR5[0])
LS: Last Segment 最后段	30	该位为 1 时表示当前 buffer 包含的是一帧数据的最后一段(如果帧分为多个段)
FS: First Segment	29	该位为 1 时表示当前 buffer 包含的是一帧数据的第一段(如果

第一段		帧分为多个段)
CIC: Checksum Insertion Control 校验数据填充控制	28:27	该域控制内部模块是否在发送帧中填充校验数据。 值：2'b00: 不填充校验数据 2'b01: 填充 IPV4 头校验数据 2'b10: 在伪头数据(pseudo-header)存在的情况下，填充 TCP/UDP/ICMP 全校验数据 2'b11: 总是填充 TCP/UDP/ICMP 的全校验数据
DC: Disable CRC 禁止 CRC 校验	26	该位为 1 时 GMAC 硬件不在每个发送帧的结尾添加 CRC 校验数据
TER: Transmit End of Ring 环形描述符结尾	25	该位为 1 时表示该描述符为环型描述符链表的最后一个，下一个描述符的地址为发送描述符链的基址
TCH: Second Address Chained 第二个 buffer 地址指向下个链式描述符	24	该位为 1 时表示描述符中的第二个 buffer 地址指向的是下一个描述符的地址，为 0 时表示该地址指向第二个 buffer 地址 当该位为 1 时，TDES1[21-11]的值将没有意义，TDES1[25]比 TDES1[24]具有更高优先级(代表环型而不是链型)
DP: Dissable Pading 禁止填充	23	该位为 1 时表示 GMAC 将不会对长度小于 64 字节的数据包进行空数据填充
TTSE: Transmit Time Stamp Enable 启用发送时间戳	22	该位为 1 时表示将启用内部模块计算 IEEE1588 硬件时间戳计算，在 TDES1[29]为 1 时有效
TBS2: Transmit Buffer Size 2 发送 buffer2 大小	21:11	该域表示数据 buffer2 的大小。当 TDES1[24]为 1 时，该域无效
TBS1: Transmit Buffer Size 1 发送 buffer1 大小	10:0	该域表示数据 buffer1 的大小。该域一直有效。如果该域值为 0，DMA 则会自动访问 buffer2 或者下一个接收描述符

## TDES2

该域记录了数据发送 buffer1 的地址。

TDES2	位	
Buffer1 Address	31:0	该域记录了数据接收 buffer1 的 32 位物理地址。该物理地址没有默认的对齐要求。当 GMAC DMA 内部实现了总线数据

Pointer 发送 buffer1 地址		32/64/128 位对齐，则该地址的低 2/3/4 位会被忽略
--------------------------	--	----------------------------------

### TDES3

该域记录了数据发送 buffer2 的地址。

TDES3 位		
Buffer2 Address Pointer 发送 buffer2 地址	31:0	该域记录了数据接收 buffer2 的 32 位物理地址。该物理地址没有默认的对齐要求。当 GMAC DMA 内部实现了总线数据 32/64/128 位对齐，则该地址的低 2/3/4 位会被忽略。如果描述符是以链式连接，则该域记录的是下一个描述符的地址

## 11.4 软件编程向导

### DMA 初始化:

1. 软件重置(reset)MAC
2. 等待重置完成(查询 DMA reg0[0])
3. 对 DMA reg0 的以下域进行编程
  - a. MIX-BURST 和 AAL(DMA reg0[26]、[25])
  - b. Fixed-burst 或者 undefined-burst(DMA reg0[16])
  - c. Burst-length 和 Burst-mode
  - d. Descriptor Length(只有当环形格式时有效)
  - e. Tx 和 Rx 仲裁调度
4. 对 AXI Bus Mode Reg 进行编程
  - a. 如果选择了 Fixed-burst, 则需要在该寄存器内设置最大 burst length
5. 分别创建发送、接收描述符链, 可以分别选择环形模式或者链型模式进行连接, 并将接收描述符的 OWN 位设为 1(DMA 拥有)
6. 在软件启用 DMA 描述符之前, 必须保证至少发送/接收描述符链中有三个描述符
7. 将发送、接收描述符链表的首地址写入 DMA reg3、4
8. 对 DMA reg6(DMA mode operation)中的以下位进行配置
  - a. 接收/发送的 Store and Forward
  - b. 接收/发送的阈值因子(Threshold Control)
  - c. 启用流控制(hardware flow control enable)
  - d. 错误帧和未识别的正确帧略过(forwarding enable)
  - e. OSF 模式
9. 向 DMA reg6(Status reg)写 1.清除所有中断请求
- 10.向 DMA reg7(interrupt enable reg)写 1, 启用所有中断
- 11.向 DMA reg6[1]、[13]中写 1, 启用发送和接收 DMA

### MAC 初始化:

1. 正确配置配套 PHY 芯片
2. 对 MAC reg4(GMII Address Register)进行正确配置,使其能够正常访问 PHY 相关寄存器
3. 读取 MAC reg5(GMII Data Register)获取当前 PHY 的链接(link)、速度(speed)、模式(双工)等信息
4. 配置 MAC 地址
5. 如果启用了 hash filtering, 则需要对 hash filtering 进行配置
6. 对 MAC reg1(Mac Frame filter)以下域进行配置, 来进行帧过滤
  - a. 接收所有

- b. 混杂模式(promiscuous mode)
  - c. 哈希或完美过滤(hash or perfect filter)
  - d. 组播、多播过滤设置等等
7. 对 MAC reg6(Flow control register)以下域进行配置
    - a. 暂停时间和其他暂停控制位
    - b. 接收和发送流控制位
    - c. 流控制忙/后压力启用
  8. 对中断掩码寄存器(Mac reg15)进行配置
  9. 基于之前得到的线路信息(link,speed,mode)对 MAC reg0 进行正确的配置
  10. 设置 MAC reg0[2]、[3]来启用 MAC 中的发送、接收模块

#### 发送和接收的一般过程:

1. 检测到发送或接收中断后, 查寻相应描述符来判断其是否属于主机, 并读取描述符中的数据
2. 完成对描述符中数据的读取后, 将描述符各位清 0 并设置其 OWN 位, 使其继续发送/接收数据
3. 如果当前发送或接收描述符不属于 DMA(OWN=0), 则 DMA 模块会进入挂起状态。当有数据需要被发送或接收时, 向 DMA Tx/Rx POLL 寄存器写 1 重新使能 DMA 模块。需要注意的是接收描述符在空闲时应该总是属于 DMA(OWN=1)
4. 发送和接收描述符及对应 buffer 地址的实时信息可以通过查寻 DMA reg18、19、20、21 获得

## 11.5 IEEE 1588 支持

MAC 控制器支持 IEEE 1588 时钟同步协议。IEEE 1588 定义了利用本地局域网、分布式对象等技术实现精确时钟同步校对、测量、控制系统的协议。IEEE1588 协议广泛应用于支持各种不同时钟特性的异构系统当中, 提供全系统的亚微秒级精度同步。下面两个图给出了典型的基于 IEEE1588 协议的两个不同时钟系统同步的过程。其中图 1 为主机周期向网络中所有从机发送同步信息的过程, 图 2 为从机主动向主机同步的过程。



3. 根据 ptp 时钟频率来设置亚秒增量寄存器(Register 449 Sub-Second Increment Register)。
4. 如果你使用了 Fine Correction approach, 设置(Register 454 Time Stamp Addend register)寄存器并将时间戳控制寄存器(reg 448)的 bit 5 设为 1 (addend reg update)。
5. 轮询时间戳控制寄存器的 bit5, 至该位为 0。
6. 通过设置时间戳控制寄存器的 bit1 为 1, 来启用 Fine Update method (如果有必要)。
7. 将时间戳高位更新寄存器(Time Stamp High Update)和时间戳低位寄存器(Time Stamp Low Update)设置为相应的值。
8. 将时间戳控制寄存器 bit2 设为 1(时间戳初始化)。
9. 当向时间戳更新寄存器(registers 452 and 453 Time Stamp Update register)中写入初始值后, 时间戳计数器开始工作。
10. 启用 MAC 的发送和接收模块来正确运行时间戳功能。

注意: 当 IEEE1588 功能中途关闭后, 重新启用时必须重复执行以上过程

系统时钟校正:

在一个进程当中同步或更新系统时间(coarse correction method), 进行以下步骤:

1. 将偏移(offset 正或负)写入时间戳更新寄存器(registers 452 and 453 Time Stamp Update registers)。
2. 将时间戳控制寄存器(reg 448)的 bit3 (TSUPDT)设为 1。
3. 当 TSUPDT 位(reg448 bit3)变为 0 时, 系统时钟加或者减去了步骤 1 中时间戳更新寄存器中的偏移值。

# 12 USB 控制器

## 12.1 总体概述

1C 的 USB 主机端口特性如下：

- 兼容 USB Rev 1.1 、 USB Rev 2.0 协议
- 兼容 OHCI Rev 1.0 、 EHCI Rev 1.0 协议
- 支持 LS (Low Speed)、FS (Full Speed) 和 HS (HighSpeed) 的 USB 设备
- 支持六个端口，每个端口都可挂 LS、FS 或 HS 设备

USB 主机控制器模块包括一个支持高速设备的 EHCI 控制器，一个支持全速与低速设备的 OHCI 控制器。其中 EHCI 控制器处于主控地位，只有当挂上的设备是全速或低速设备时，才将控制权转交给 OHCI 控制器；当全速或低速设备拔掉时，控制权返回 EHCI 控制器。

同时 USB 控制器内部集成了 AHB 总线接口（与 AMBA Specification Revision 2.0 兼容），用来和内存/应用程序之间通信。USB 控制器与外部的互联结构图如图 12-1 所示：

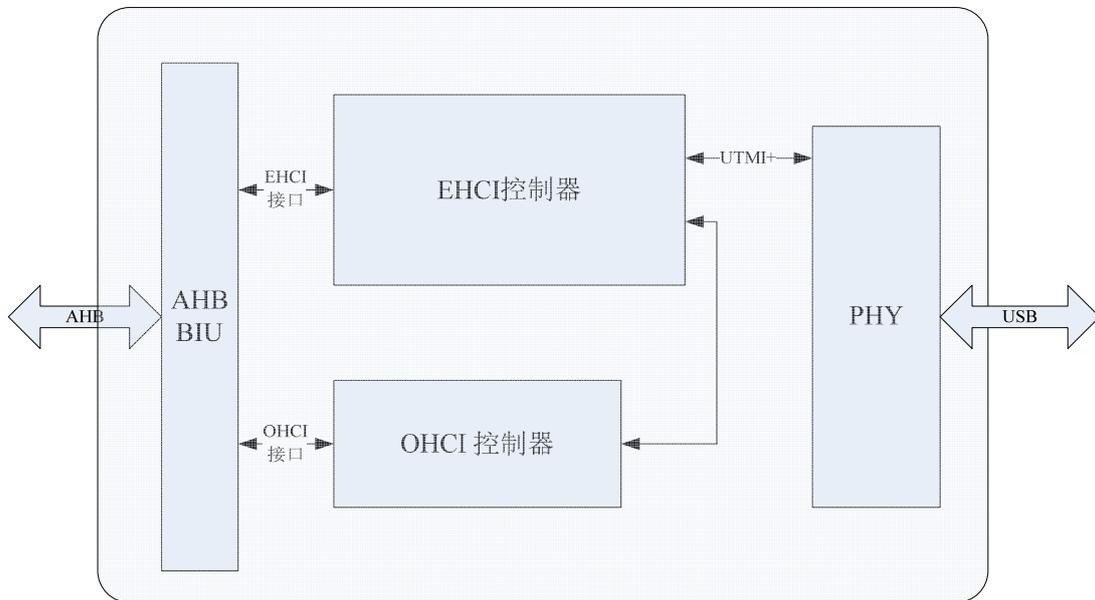


图 12-1 USB 主机控制器模块图

## 12.2 USB 主机控制器寄存器

### 12.2.1 EHCI 相关寄存器

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。1C 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。

EHCI 寄存器的基址为，以下寄存器的地址为基址加偏移。

### 12.2.2 Capability 寄存器

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

(注：USBBase 固定为 EHCI slave 的起始地址 0xbfe0\_0000)

### 12.2.3 Operational 寄存器

名称	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 slave 的起始地址+`h10)

#### 12.2.4 EHCI 实现相关寄存器

EHCI 实现相关寄存器的详细描述如下。

名称	地址	宽度	访问	说明
INSNREG00	0x90	32	R/W	帧的长度配置寄存器
INSNREG01	0x94	32	R/W	数据包缓冲区 OUT/IN 阈值寄存器
INSNREG02	0x98	32	RO	数据包缓冲深度寄存器
INSNREG03	0x9c	32	RO,R/W	参照寄存器详细描述
INSNREG04	0xa0	32	R/W	用于 Debug
INSNREG05	0xa4	32	RO,R/W	UTMI 配置 (默认配置), 控制和状态寄存器
INSNREG06	0xa8	32	RO	AHB 错误状态寄存器
INSNREG07	0xac	32	RO	AHB Master 错误地址寄存器
INSNREG08	0xb0	32	RO	HSIC 使能寄存器

#### INSNREG00 寄存器 (disable)

#### INSNREG01 寄存器

位域	访问	复位值	说明
31:16	R/W	16'h0020	OUT 阈值 (单位是 4 bytes), 一旦从系统内存中取出的数据量达到 OUT 阈值, 就开始 USB 传输, 最小为 16bytes
15:0	R/W	16'h0020	IN 阈值 (单位是 4 bytes), 一旦 Packet Buffer 里的数据量达到 IN 阈值, 就开始向内存传输, 最小为 16bytes

#### INSNREG02 寄存器

位域	访问	复位值	说明
31:12	Reserved	20'h0	保留
11:0	RO	12'h0020	数据包缓冲深度 (单位是 4 bytes)

#### INSNREG03 寄存器

位域	访问	复位值	说明

31:13	Reserved	19'h0	保留
12:10	RO	3'h0	这个字段指定 phy_clks 的额外延时，这个延时被添加到“Tx-Tx turnaround Delay”中。
9	RO	1'h0	置 1：将迫使主机控制器在一帧的每一微帧中获取周期数据帧表， 置 0：主机控制器在一帧的微帧 0 中获取周期数据帧表
8:1	R/W	8'h0	时间可容忍偏移，这个字段用来指明为了容忍计算可用时间而要附加的字节数。计算可用时间为以后的传输弹性增加的，用户程序默认不需要修改这个字段。
0	RO	1'h0	Break Memory Transaction 模式 置 1：使能此功能 置 0：禁止此功能

**INSNRE04 寄存器（仅用于调试，软件不必更改此寄存器）**

位域	访问	复位值	说明
31:6	Reserved	26'h0	保留
5	R/W	1'h0	置 1：禁止 automatic 功能，即当软件清除 Run/Stop 位时，USB 主机控制器会把挂起（Suspend）的端口唤醒 置 0：启用 automatic 功能，当 reset Run/Stop 位时，Suspend 信号会置为 1
4	R/W	1'h0	置 1：禁止 NAK reload 修复 置 0：启用 NAK reload 修复
3	Reserved	1'h0	保留
2	R/W	1'h0	置 1：缩短端口枚举（enumeration）时间（仿真）
1	R/W	1'h0	置 1： HCCPARAMS 寄存器的第 17、15:4、2:0 位均可写
0	R/W	1'h0	置 1： HCSPARAMS 寄存器可写

**INSNRE05 寄存器**

位域	访问	复位值	说明
31:18	Reserved	14'h0	保留
17	RO	1'h0	置 1：表示对这个寄存器进行了一个写操作，硬件正在执行 置 0：表示硬件已经执行完操作
16:13	R/W	5'h0	端口号

12	R/W	4'h1	VControlLoadM 置 1: NOP 置 0: Load
11:8	R/W	4'h0	VControl
7:0	RO	4'h0	VStatus

### INSNREG06 寄存器

位域	访问	复位值	说明
31	R/W	1'h0	一旦 AHB 出错即被捕获并置 1，写 0 清除该字段
30:12	Reserved	19'h0	保留
11:9	RO	3'h0	AHB 出错时控制段 HBURST 的值
8:4	RO	5'h0	AHB 出错的 burst 的预计节拍数
3:0	RO	4'h0	在当前 burst 下，AHB 出错前完成的节拍数

### INSNREG07 寄存器

位域	访问	复位值	说明
31:0	RO	32'h0	AHB 出错时控制段的地址

### INSNREG08 寄存器

位域	访问	复位值	说明
31:0	RO	1'b0	HSIC 使能

## 12.3 OHCI 相关寄存器

OHCI 的相关寄存器包括 Operational 寄存器和 OHCI 实现相关寄存器。1C 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。

OHCI 寄存器的基址为 0xbf080000，以下寄存器的地址为基址加偏移。

### 12.3.1 Operational 寄存器

名称	地址	宽度	访问	说明
HcRevision	0x00	32	-	控制和状态
HcControl	0x04	32	-	

HcCommonStatus	0x08	32	-	内存指针
HcInterruptStatus	0x0C	32	-	
HcInterruptEnable	0x10	32	-	
HcInterruptDisable	0x14	32	--	
HcHCCA	0x18	32	-	
HcPeriodCuttentED	0x1C	32	-	
HcControlHeadED	0x20	32	-	
HcControlCurrentED	0x24	32	-	
HcBulkHeadED	0x28	32	-	
HcBulkCurrentED	0x2C	32	-	
HcDoneHead	0x30	32	-	
HcRmInterval	0x34	32	-	
HcFmRemaining	0x38	32	-	
HcFmNumber	0x3C	32	-	
HcPeriodicStart	0x40	32	-	
HcLSThreshold	0x44	32	-	
HcRhDescriptorA	0x48	32	-	根集线器
HcRhDescriptorB	0x04C	32	-	
HcRhStatus	0x50	32	-	
HcRhPortStatus1	0x54	32	-	
HcRhPortStatus2	0x58	32	-	

### 12.3.2 OHCI 实现相关寄存器

除了标准的 OHCI 操作寄存器，还实现了两个额外寄存器（寄存器偏移 0x98 和 0x9C）用来报告 AHB 的错误状态。

名称	地址	宽度	访问	说明
INSNREG06	0x98	32	RO	AHB 错误状态寄存器
INSNREG07	0x9c	32	RO	AHB Master 错误地址寄存器

### 12.3.3 INSNREG06 寄存器

位域	访问	复位值	说明
31	R/W	1'h0	一旦 AHB 出错即被捕获并置 1，写 0 清除该字段
30:12	RO	19'h0	保留
11:9	RO	3'h0	AHB 出错时控制段 HBURST 的值

8:4	RO	5'h0	AHB 出错的 burst 的预计节拍数
3:0	RO	4'h0	在当前 burst 下，AHB 出错前完成的节拍数

#### 12.3.4 INSNREG07 寄存器

位域	访问	复位值	说明
31:0	RO	32'h0	AHB 出错时控制段的地址

# 13 OTG 控制器

## 13.1 概述

1C 的 OTG 支持特性如下：

- 支持 HNP 与 SRP 协议；
- 内嵌 DMA，无需占用处理器带宽即可在 OTG 与外部存储之间移动数据
- 在 device 模式下，为高速设备（480Mbps）；
- 在 host 模式下，仅能支持高速设备（480Mbps）；
- 在 device 模式下，支持 2 个双向的 endpoint，其中仅有默认的 endpoint0 支持控制传输；
- 在 device 模式下，最多同时支持 2 个 IN 方向的传输；
- 在 host 模式下，支持 6 个 channel，且软件可配置每个 channel 的方向；
- 在 host 模式下，支持 periodic OUT 传输；

## 13.2 寄存器描述

应用程序通过 AHB slave 接口来读写 OTG 控制器里的控制与状态寄存器（CSRs），这些寄存器都是 32 位宽，寄存器地址为 32 位对齐。

这里需要注意的是，在 host 模式与 device 模式下都能访问的寄存器组仅包括全局寄存器组（Core Global），功耗与门控时钟（Power and Clock Gating）寄存器组，数据 FIFO 访问（Data FIFO Access）寄存器组，端口（Host Port）寄存器组。当 OTG 控制器处于 host 或 device 模式下，不能访问另一个模式下的寄存器。如果发生非法的寄存器读写，将发生模式不匹配（Mode Mismatch）中断，这个中断将反映在中断寄存器中（Core Interrupt Register）。

当 OTG 从一个模式转换到另一个模式时，必须重新配置这个模式下的寄存器，因为这些寄存器在转换后的状态与上电重启时是一样的。

CSR 的地址映射是固定的，其中 Host 模式下的寄存器与 Device 模式下的寄存器处于不同的地址空间，所有的寄存器都工作在 AHB 时钟域里。下图显示了 CSR 的地址映射关系：

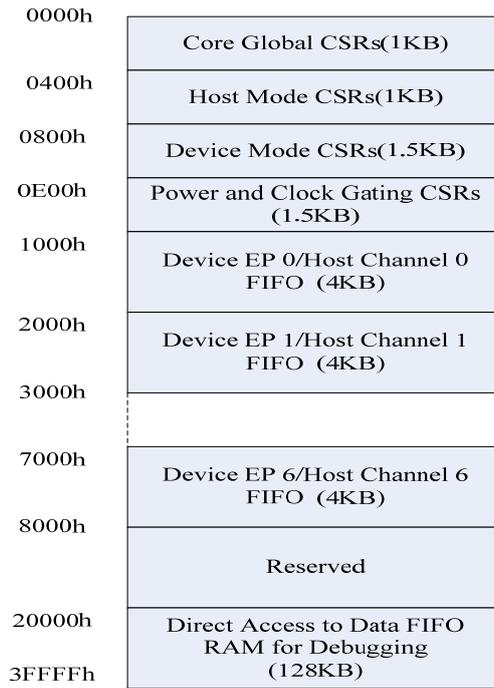


图 13-1 OTG CSRs 地址映射

以下章节分组列出了 OTG 所有的寄存器及其偏移地址，其基地址为 0x1fe0\_0000。

### 13.2.1 全局控制与状态寄存器（Global CSR Map）

不管 OTG 工作在 host 模式下还是 Device 模式下都可访问这组寄存器，其对应的偏移地址如下表所示：

寄存器缩略名	偏移地址	寄存器名
GOTGCTL	000h	Control and Status Register
GOTGINT	004h	Interrupt Register
GAHBCFG	008h	AHB Configuration Register
GUSBCFG	00Ch	USB Configuration Register
GRSTCTL	010h	Reset Register
GINTSTS	014h	Interrupt Status Register
GINTMSK	018h	Interrupt Mask Register
GRXSTSR	01Ch	Receive Status Debug Read/Status Read and Pop Registers
GRXSTSP	020h	
GRXFSIZ	024h	Receive FIFO Size Register
GNPTXFSIZ	028h	Non-Periodic Transmit FIFO Size Register
GNPTXSTS	02Ch	Non-Periodic Transmit FIFO/Queue Status Register
GSNPSID	040h	Synopys ID Register
GHWCFG1	044h	User HW Config1 Register
GHWCFG2	048h	User HW Config2 Register
GHWCFG3	04Ch	User HW Config3 Register
GHWCFG4	050h	User HW Config4 Register

GDFIFOCFG	05Ch	DFIFO Software Config Register
HPTXFSIZ	100h	Host Periodic Transmit FIFO Size Register
DIEPTXFn	104h-3FFh	Device IN Endpoint Transmit FIFO Size Register

### 13.2.2 Host 模式控制与状态寄存器 (Host Mode CSR)

一旦 OTG 工作在 Host 模式下，这组寄存器必须被重新配置。

寄存器缩略名	偏移地址	寄存器名
HCFG	400h	Host Configuration
HFIR	404h	Host Frame Interval Register
HFNUM	408h	Host Frame Number/Frame Time Remaining Register
	40ch	Reserved
HPTXSTS	410h	Host Periodic Transmit FIFO/Queue Status Register
HAINT	414h	Host All Channels Interrupt Register
HAINTMSK	418h	Host All Channels Interrupt Mask Register
HPRT	440h	Host Port Control and Status Register
	444h-4FCh	Reserved
HCCHARn	500h	Moving the Host Core to Test Mode
HCSPLTn	504h	Host Channel-n Split Control Register (HCSPLTn)
HCINTn	508h	Host Channel-n Interrupt Register (HCINTn)
HCINTMSKn	50Ch	Host Channel-n Interrupt Mask Register (HCINTMSKn)
HCTSIZn	510h	Host Channel-n Transfer Size Register (HCTSIZn)
HCDMAAn	514h	Host Channel-n DMA Address Register
	518h	Reserved
HCDMABn	51Ch	Host Channel-n DMA Buffer Address Register
HCCHARn	520h-53Ch 540h-55h ... 6C0h-6DCh 6E0h-6FCh 6FDh-7FFh	Host Channel-n Characteristics Register

### 13.2.3 Device 模式控制与状态寄存器 (Device Mode CSR)

一旦 OTG 工作在 Device 模式下，这组寄存器必须被重新配置。

寄存器缩略名	偏移地址	寄存器名
DCFG	800h	Device Configuration Register
DCTL	804h	Device Control Register
DSTS	808h	Device Status Register
	80Ch	Reserved
DIEPMSK	810h	Device IN Endpoint Common Interrupt Mask Register
DOEPMSK	814h	Device OUT Endpoint Common Interrupt Mask Register
DAINT	818h	Device All Endpoints Interrupt Register
DAINTMSK	81Ch	Device All Endpoints Interrupt Mask Register

DVBUSDIS	828h	Device VBUS Discharge Time Register
DVBUSPULSE	82Ch	Device VBUS Pulsing Time Register
DTHRCTL	830h	Device Threshold Control Register
DIEPEMSK	834h	Device IN Endpoint FIFO Empty Interrupt Mask Register
DIEPCTL0	900h	Device Control IN Endpint0 Control Register
	904h	Reserved
DIEPCTLn	920-AE0h	Device Endpoint-n Control Register
DIEPINTn	908h	Device Endoint-n Interrupt Register
	90Ch	Reserved
DIEPTSIZ0/DOEPTSIZ0	910h	Device Endpoint0 Transfer Size Register
DIEPTSIZn/DOEPTSIZn	910h	Device Endpoinn Transfer Size Register
DIEPDMA <sub>n</sub>	914h	Device Endpoint-n DMA Address Register
DTXFST <sub>n</sub>	918h	Device IN Endpoint Transmit FIFO Status Register
DIEPDMAB0	91Ch	Device Endpoint-n DMA Buffer Address Register
DOEPCTL0	B00h	Device Control OUT Endpint0 Control Register
	B04h	Reserved
DOEPCTLn	B20-BE0h	Device Endpoint-n Control Register
DOEPINTn	B08h	Device Endoint-n Interrupt Register
	B0Ch	Reserved
DOEPTSIZ0	B10h	Device Endpoint0 Transfer Size Register
DOEPTSIZn	B10h	Device Endpoinn Transfer Size Register
DOEPDMA <sub>n</sub>	B14h-CF4h	Device Endpoint-n DMA Address Register
DOEPDMAB0	B1Ch-CFCh	Device Endpoint-n DMA Buffer Address Register

### 13.2.4 数据 FIFO 访问寄存器组 (DFIFO Access Register MAP)

不管 OTG 工作在 host 模式下还是 Device 模式下，都可访问这组寄存器。这组寄存器用作读写某个给定方向的 endpoint 或 channel 的 FIFO。如果 host 模式下 channel 的方向为 IN，那么只能读这个 channel 上的 FIFO；相类似地，若 host 模式下 channel 的方向为 OUT，那么只能写这个 channel 上的 FIFO。

寄存器名	偏移地址	读写特性
Device IN Endpoint 0/Host OUT Channel 0: DFIFO Write Access Device OUT Enpoint 0/Host IN Channel 0 : DFIFO Read Access	1000h-1FFCh	WO/RO
Device IN Endpoint 0/Host OUT Channel 0: DFIFO Write Access Device OUT Enpoint 0/Host IN Channel 0 : DFIFO Read Access	2000h-2FFCh	WO/RO
Device IN Endpoint 14/Host OUT Channel 14: DFIFO Write Access Device OUT Enpoint 14/Host IN Channel 14 : DFIFO Read	F000h-FFFCh	WO/RO

Access		
Device IN Endpoint 15/Host OUT Channel 15: DFIFO Write Access	10000h-10FFCh	WO/RO
Device OUT Endpoint 15/Host IN Channel 15: DFIFO Read Access		

### 13.2.5 功耗控制与门控时钟寄存器组 (Power and Clock Gating CSR Map)

不管 OTG 工作在 host 模式下还是 Device 模式下，都可访问这组寄存器。这组寄存器用作功耗控制与门控时钟。

寄存器缩略名	偏移地址	描述
PCGCCTL	E00h	Power and Clock Gating Control Register

## 13.3 寄存器描述

### 13.3.1 寄存器访问特性

在以下章节描述寄存器时，会有一栏访问特性，用来标示这些寄存器的读写特性。下面的内容将列出这些访问特性的具体含义。

<b>Read Only (RO)</b>	只读
Write Only(WO)	只写
Read and Write(R_W)	可读可写
Read, Write, and Self Clear(R_W_SC)	可读可写；且 OTG 控制器自身可清 0，具体清 0 的条件在以下章节的各个域中有详细解释
Read, Write, Self Set and Self Clear (R_W_SS_SC)	可读可写；且 OTG 控制器自身可置位，可清 0，具体置位与清 0 的条件在以下章节的各个域中有详细解释
Read, Self Set, and Write Clear (R_SS_WC)	可读；且 OTG 控制器自身可置位；且当软件往这位写 1 完成清 0 效果，往这位写 0 不会产生效果。OTG 置位的条件在以下章节的各个域中有详细解释
Read, Write Set, and Self Clear (R_WS_SC)	可读；且当软件往这位写 1 完成置位效果；且 OTG 控制器自身可清 0，软件不可清 0，即软件往这位写 1 不会产生效果。OTG 清 0 的条件在以下章节的各个域中有详细解释
Read, Self Set, and Self Clear or Write Clear (R_SS_SC_WC)	可读；且 OTG 控制器自身可置位，可清 0；当软件往这位写 1 完成清 0 效果，往这位写 0 不会产生效果。OTG 具体置位与清 0 的条件在以下章节的各个域中有详细解释

### 13.3.2 全局寄存器

不管 OTG 工作在 host 模式下还是 Device 模式下都可访问这组寄存器，当 OTG 从一种工作模式转换到另一种工作模式时，无需重新例化这些寄存器。

#### 控制与状态寄存器 (GOTGCTL)

偏移地址：000h

OTG 控制与状态寄存器 (GOTGCTL) 控制 OTG 的功能以及反映其状态。

域	说明	工作模式	复位值	访问特性
31:21	保留域	Host; Device		RO
20	OTG 版本	Host;	1'b0	RW

	<ul style="list-style-type: none"> <li>● 1'b0: 版本 1.3; 此版本 OTG 支持 Data line pulsing 与 VBus pulsing 的 SRP</li> <li>● 1'b1: 版本 2.0; 此版本的 OTG 仅支持 Data line pulsing 的 SRP</li> </ul>	Device		
19	<p>B-session valid(BSesVld) 指定 device 模式下 transceiver 的状态。</p> <ul style="list-style-type: none"> <li>● 1'b0: B-session 非法</li> <li>● 1'b1: B-session 合法</li> </ul> <p>在 OTG 模式下, 用这位来决定设备是否连接上</p>	Device only	1'b1	RO
18	<p>session valid(ASesVld) 指定 host 模式下 transceiver 的状态。</p> <ul style="list-style-type: none"> <li>● 1'b0: A-session 非法</li> <li>● 1'b1: A-session 合法</li> </ul> <p>在 device 模式下, 此位为保留位</p>	Host only	1'b0	RO
17	<p>Long/Short Debounce Time (DbncTime) 指定一个发现连接的 debounce 时间</p> <ul style="list-style-type: none"> <li>● 1'b0: long debounce time, 用作物理连接 (100ms + 2.5us)</li> <li>● 1'b1: short debounce time, 用作软连接 (2.5us)</li> </ul>	Host only	1'b0	RO
16	<p>连接器 ID 的状态 在一次连接事件中, 指示 ID 的状态</p> <ul style="list-style-type: none"> <li>● 1'b0: OTG 工作在 A 设备模式下</li> <li>● 1'b1: OTG 工作在 B 设备模式下</li> </ul>	Host ; Device	1'b1	RO
15:12	保留域	Host ; Device		RO
11	<p>设备 HNP 使能位 (DevHNPEn)</p> <ul style="list-style-type: none"> <li>● 1'b0: HNP 未被使能</li> <li>● 1'b1: HNP 使能</li> </ul>	Device Only	1'b0	R_W
10	<p>Host Set HNP 使能位</p> <ul style="list-style-type: none"> <li>● 1'b0: Host Set HNP 未被使能</li> <li>● 1'b1: Host Set HNP 使能</li> </ul>	Host Only	1'b0	R_W
9	<p>HNP 请求 软件将这位置位从而发起一个 HNP 请求给 host。当 GOTGINT.HstNegSucStsChng 位置</p>	Device Only	1'b0	R_W

	<p>位时，软件通过往这位写 0 来清 0。当 HstNegSucStsChng 位为 0，OTG 控制器将这位清 0。</p> <ul style="list-style-type: none"> <li>● 1'b0: 无 HNP 请求</li> <li>● 1'b1: HNP 请求</li> </ul>			
8	<p>Host Negotiation Success (HstNegScs)</p> <p>当 host 互换 (Host Negotiation) 成功后,OTG 控制器将这位置起。</p> <ul style="list-style-type: none"> <li>● 1'b0: host 互换失败</li> <li>● 1'b1: host 互换成功</li> </ul>	Device only	1'b0	RO
7	<p>B-Peripheral Session Valid 覆盖值 (BvalidOvVal)</p> <p>当 GOTGCTL.BvalidOvEn 置位时，BValidOvVal 位用来重新设置 Bvalid 信号..</p> <ul style="list-style-type: none"> <li>● 1'b0: 当 GOTGCTL.BvalidOvEn = 1 时，Bvalid 信号的值为 0;</li> <li>● 1'b1: 当 GOTGCTL.BvalidOvEn = 1 时，Bvalid 信号的值为 1;</li> </ul>	Device only	1'b0	R_W
6	<p>B-Peripheral Session Valid 重置使能 (BvalidOvEn)</p> <p>这位用来使能软件通过 GOTGCTL.BValidOvVal 位来覆盖 Bvalid 信号。</p> <ul style="list-style-type: none"> <li>● 1'b1:从 PHY 接收到的 Bvalid 信号值被 BvalidOvVal 重新设置。</li> <li>● 1'b0:Bvalid 信号重置功能关闭，OTG 控制器内部使用的 Bvalid 信号的值即位从 PHY 接收到的值</li> </ul>	Device only	1'b0	R_W
5	<p>A-Peripheral Session Valid 重置值 (AvalidOvVal)</p> <p>当 GOTGCTL.AvalidOvEn 置位时，AValidOvVal 位用来重新设置 Avalid 信号</p> <ul style="list-style-type: none"> <li>● 1'b0: 当 GOTGCTL.AvalidOvEn = 1 时，Avalid 信号的值为 0;</li> <li>● 1'b1: 当 GOTGCTL.AvalidOvEn</li> </ul>	Host only	1'b0	R_W

	= 1 时, Avalid 信号的值为 1;			
4	<p>A-Peripheral Session Valid 重置使能 (AvalidOvEn)</p> <p>这位用来使能软件通过 GOTGCTL.AValidOvVal 位来 Override Avalid 信号。</p> <ul style="list-style-type: none"> <li>● 1'b1:从 PHY 接收到的 Avalid 信号值被 AvalidOvVal 重新设置。</li> <li>● 1'b0:Avalid 信号重置功能关闭, OTG 控制器内部使用的 Avalid 信号的值即位从 PHY 接收到的值</li> </ul>	Host only	1'b0	R_W
3	<p>VBUS Valid 重置值(VbvalidOvVal)</p> <p>当 GOTGCTL.VbvalidOvEn 置位时, VbvalidOvVal 位用来重新设置 vbus valid 信号</p> <ul style="list-style-type: none"> <li>● 1'b0: 当 GOTGCTL.VbvalidOvEn = 1 时, Vbus valid 信号的值为 0;</li> <li>● 1'b1: 当 GOTGCTL.VbvalidOvEn = 1 时, vbus valid 信号的值为 1;</li> </ul>	Host only	1'b0	R_W
2	<p>VBUS Valid 重置使能(AvalidOvEn)</p> <p>这位用来使能软件通过 GOTGCTL.VbValidOvVal 位来 Override vbus valid 信号。</p> <ul style="list-style-type: none"> <li>● 1'b1: 从 PHY 接收到的 vbus valid 信号值被 VbvalidOvVal 重新设置。</li> <li>● 1'b0: vbus valid 信号重置功能关闭, OTG 控制器内部使用的 vbus valid 信号的值即位从 PHY 接收到的值</li> </ul>	Host only	1'b0	R_W
1	<p>会话请求 (Session Request, SesReq)</p> <p>软件通过置位 SesReq 位在 USB 总线上发起一次会话请求。当 OTG 中断寄存器的 Host Negotiation Success Status Change 位置位时, 软件通过往这位写 0 来清 0。当 HstNegSucStsChng 位为 0, OTG 控</p>	Device only	1'b0	R_W

	制器将这位清 0. ● 1'b0: 无会话请求 ● 1'b1: 请求会话			
0	会话请求成功 (Session Request Success, SesReqScs) ● 1'b0: 会话请求失败 ● 1'b1: 会话请求成功	Device only	1'b0	R_W

### 中断寄存器 (GOTGINT)

偏移地址: 004h

域	说明	工作模式	复位值	访问特性
31:20	保留域	Host; Device		RO
19	Debounce Done (DbnceDone) 当设备连接上的 debounce 完成后, OTG 控制器置起这位。当应用看到这个中断后, 可以开始 USB 复位操作。	Host only	1'b0	R_SS_W C
18	A 设备超时改变 (ADevTOUTChg) 当 A 设备等待 B 设备连接超时, OTG 控制器置起这位	Host ; Device	1'b0	R_SS_W C
17	检测到角色互换 (HstNegDet): 当在 USB 总线上发现角色互换请求 (Host Negotiation Req) 时, OTG 控制器置起这位。	Host ; Device	1'b0	R_SS_W C
16:10	保留域	Host and Device		RO
9	角色互换成功状态改变位 (HstNegSucStsChng): Host Negotiation Success Status Change 当 USB 总线上的角色请求成功或失败时, OTG 控制器将这位置起。软件可通过查询 GOTGCTL.HstNegSus 位来确定角色互换请求是否成功	Host and Device	1'b0	R_SS_W C
8	会话请求成功状态改变位 (SesReqSusStsChng): Session Request Success Status Change 当 USB 总线上的会话请求成功或失败时, OTG 控制器将这位置起。软件可通过查询 GOTGCTL.SesReqScs 位来确定会话请求是否成功	Host and Device	1'b0	R_SS_W C

7:3	保留域	Host and Device	1'b0	RO
2	会话结束检测位 (SesEndDet): Session End Detected 当 utmisrp_bvalid 信号拉低时, OTG 控制器置起这位	Device	1'b0	R_SS_WC
1:0	保留域	Host and Device		RO

### AHB 配置寄存器 (GAHBCFG)

偏移地址: 008h

这个寄存器用来在上电或工作模式改变时配置 OTG 控制器, 配置的内容主要包括 AHB 相关的系统参数。在初始配置完成后, 不能再修改这个寄存器的值。不管是在 AHB 总线还是 USB 总线上开始一次事务 (transaction) 时, 都需要重新配置这个寄存器。

域	说明	工作模式	复位值	访问特性
31:23	保留域	Host; Device		RO
22	<p>通知所有的 DMA 写事务 (NotiAllDmaWrit): Notify All Dma Write Transactions</p> <p>用来使能对所有的 DMA 写事务 (to endpoint/Channel) 的 DMA Done 功能。只有当 GAHBCFG.RemMemSupp 位为 1'b1 时, 这位才意义。</p> <ul style="list-style-type: none"> <li>● 1'b1: 对于所有的 DMA 写事务, OTG 控制器都会在 AHB 总线上置起 (assert) dma_req, int_dma_done, chep_last_transact 以及 chep_number 信号。为了完成对某个 channel/endpoint 的传输, OTG 控制器需要等待所有写事务的 sys_dma_done 信号。</li> <li>● 1'b0: 仅对写传输 (transfer) 的最后一个事务 (transaction), OTG 控制器才会置起 int_dma_done 信号。同样, 为了完成对某个 channel/endpoint 的传输, OTG 控制器仅需要等待那个 DMA 写事务的 sys_dma_done 信号。</li> </ul>	Host; Device	1'b0	R_W

21	<p>Remote Memory Support (RemMemSupp)</p> <p>使能为 DMA 写传输等待系统 DMA DONE 信号功能</p> <ul style="list-style-type: none"> <li>● 1'b1: 当 OTG DMA 开始对外部存储空间的传输时, 置起输出信号 int_dma_req。当完成这个传输, OTG 控制器置起 int_dma_done 信号用来指示从 OTG 控制器来的 DMA 写完成。然后 OTG 控制器等待从系统来 sys_dma_done 信号, 以便完成对某个 channel/endpoint 的数据传输。</li> <li>● 无需置起 int_dma_req 与 int_dma_done 信号; 一旦 DMA 写传输完成, XferComp 中断之气, OTG 控制器继续工作。且无需等待 sys_dma_done 信号以完成数据传输。</li> </ul>	Host Device and	1'b0	R_W
8	<p>周期性 Tx FIFO Empty Level (PTxFIFOEmpLvl): Periodic Tx FIFO Empty Level 指示周期性 Tx FIFO 空中断位 (GINTSTS.PTxFEMp) 何时置起。这位仅用在 slave 模式下。</p> <ul style="list-style-type: none"> <li>● 1'b0: GINTSTS.PTxFEMp 中断位指示当周期性 Tx FIFO 半空</li> <li>● 1'b1: GINTSTS.PTxFEMp 中断位指示周期性 Tx FIFO 全空</li> </ul>	Host only	1'b0	R_W
7	<p>非周期性 Tx FIFO Empty Level (NPTxFIFOEmpLvl): Non-Periodic Tx FIFO Empty Level 这位仅用在 slave 模式下。</p> <p>指示 IN 端点的 Tx FIFO 空中断位 (DIEPINTn.TxFEMp) 何时置起。</p> <p>Host 模式:</p> <ul style="list-style-type: none"> <li>● 1'b0: GINTSTS.PTxFEMp 中断位指示非周期性 Tx FIFO 半空;</li> <li>● 1'b1: GINTSTS.PTxFEMp 中断位指示非周期性 Tx FIFO 全空;</li> </ul> <p>device 模式:</p>	Host device and	1'b0	R_W

	<ul style="list-style-type: none"> <li>● 1'b0: DIEPINTn.TxFEmp 中断位指示 IN 端点的 Tx FIFO 半空;</li> <li>● 1'b1: DIEPINTn.TxFEmp 中断位指示 IN 端点的 Tx FIFO 全空;</li> </ul>			
6	保留域	Host; Device		RO
5	DMA 使能 (DMAEn) <ul style="list-style-type: none"> <li>● 1'b0: OTG 工作在 Slave 模式</li> <li>● 1'b1: OTG 工作在 DMA 模式</li> </ul>	Host; Device	1'b0	R_W
4:1	Burst 长度(HBstLen) AHB Master 的 burst 类型 <ul style="list-style-type: none"> <li>● 4'b0000: Single</li> <li>● 4'b0001: INCR</li> <li>● 4'b0000: INCR4</li> <li>● 4'b0000: INCR8</li> <li>● 4'b0000: INCR16</li> <li>● 其他: 保留</li> </ul>	Host; Device	4'b0	R_W
0	全局中断屏蔽位(GlblIntrMsk) 用来屏蔽对软件的中断。不管是否置起这位, OTG 控制器总是更新中断状态寄存器。 <ul style="list-style-type: none"> <li>● 1'b0: 中断对软件不可见</li> <li>● 1'b1: 中断对软件可见</li> </ul>	Host; Device	1'b0	R_W

### USB 配置寄存器 (GUSBCFG)

偏移地址: 00Ch

这个寄存器用来在上电或工作模式改变时配置 OTG 控制器, 配置的内容主要包括 USB 与 USB-PHY 相关的参数。不管是在 AHB 总线还是 USB 总线上开始一次事务 (transaction) 时, 都需要配置这个寄存器。在初始配置完成后, 不能再修改这个寄存器的值。

域	说明	工作模式	复位值	访问特性
31	Corrupt Tx package () 仅用来做调试使用, 不可写 1.	Host; Device	1'b0	WO
30	强制 device 模式(ForceDevMod) 写 1 意味着 OTG 控制器将工作在 device 模式下, 而不管 utmiotg_iddig 输入管脚的值。 <ul style="list-style-type: none"> <li>● 1'b0: 正常工作模式</li> <li>● 1'b1: 强制 device 模式</li> </ul> 将这位写 1 后, 软件必须等待至少 25ms 的生效时间。当在 scale down 模式下的仿真环境下, 500us 已经足够。	Host; Device	1'b0	R_W
29	强制 Host 模式(ForceHostMod)	Host;	1'b0	R_W

	<p>写 1 意味着 OTG 控制器将工作在 Host 模式下，而不管 utmiotg_iddig 输入管脚的值。</p> <ul style="list-style-type: none"> <li>● 1'b0: 正常工作模式</li> <li>● 1'b1: 强制 Host 模式</li> </ul> <p>将这位写 1 后，软件必须等待至少 25ms 的生效时间。当在 scale down 模式下的仿真环境下，500us 已经足够。</p>	Device		
28	<p>Tx End 延时 (TxEndDelay)</p> <p>写 1 使能 TxEndDelay 计时器。</p> <ul style="list-style-type: none"> <li>● 1'b0: 正常模式</li> <li>● 1'b1: 引入 Tx end 延时计时器</li> </ul>	Device only	1'b0	R_W
27: 23	保留	Host; Device		RO
22	<p>TermSel DLine Pulsing 选择 (TermSelDLPulse)</p> <p>用来在 SRP 过程中，选择 utmi_termselect 驱动数据脉冲 (data line pulse)。</p> <ul style="list-style-type: none"> <li>● 1'b0: 用 utmi_txvalid 驱动数据脉冲 (默认)</li> <li>● 1'b1: 用 utmi_termselect 驱动数据脉冲</li> </ul>	Device	1'b0	R_W
21: 16	保留	Host; Device		RO
15	<p>PHY 的低功耗时钟选择 (PhyLpwClkSel)</p> <p>选择使用 480MHz 或 48Mhz(低功耗)PHY 模式。在 FS 与 LS 模式下，PHY 通常工作在 48MHz 下以节省功耗。</p> <ul style="list-style-type: none"> <li>● 1'b0: 480MHz 内部 PLL 时钟</li> <li>● 1'b1: 48MHz 外部时钟</li> </ul>	Host; Device	1'b0	R_W
14	保留位	Host; Device		RO
13:10	<p>USB 周期时间 (USBTrdTime): USB Turnaround Time</p> <p>设置 PHY 时钟位单位的周期时间。即指定 MAC 请求 PFC(数据 FIFO 控制器)去数据 FIFO 中获取数据的响应时间。其值必须如下设置:</p> <ul style="list-style-type: none"> <li>● 4'h5: 当 MAC 接口位 16 位的 UTMI+</li> <li>● 4'h6: 当 MAC 接口位 8 位的 UTMI+</li> </ul>	Device	4'h5	R_W
9	<p>HNP-Capable(HNPCap)</p> <p>软件用此控制 OTG 控制器的 HNP 功能。</p>	Host, Device	1'b0	R_W

	<ul style="list-style-type: none"> <li>● 1'b0: HNP 功能不可用</li> <li>● 1'b0: HNP 功能可用</li> </ul> <p>注意，如果 HNP 功能被软件禁掉，那么 PHY 域相应的 OTG 信号必须固定为确定的值 (?)。</p>			
8	<p>SRP-Capable(SRPCap)</p> <p>软件用此控制 OTG 控制器的 SRP 功能。</p> <ul style="list-style-type: none"> <li>● 1'b0: SRP 功能不可用</li> <li>● 1'b0: SRP 功能可用</li> </ul> <p>注意，如果 SRP 功能被软件禁掉，那么 PHY 域相应的 OTG 信号必须固定为确定的值</p>	Host; Device	1'b0	R_W
7: 4	保留位	Host; Device		RO
3	<p>PHY 接口 (PHYIf)</p> <p>用来配置 UTMI+PHY 的接口。</p> <ul style="list-style-type: none"> <li>● 1'b0: 8bits</li> <li>● 1'b1: 16bits</li> </ul> <p>若选择 ULPI 接口，则仅能配置位 8bits。</p>	Host; Device	可配	RO
2:0	<p>HS 超时校准 (TOutCal): HS Timeout Calibration 单位为一个 PHY 时钟。用来修正因为 PHY 引起的包之间超时时间。不同的 PHY 引入的延时不同。</p> <p>对于 HS 而言，USB 标准的超时值为 736 – 816 bit times; 软件必须基于枚举的速度来配置这个域。每个 PHY 时钟的 bits 时间如下解释:</p> <p>HS:</p> <ul style="list-style-type: none"> <li>● 一个 30MHz 的 PHY 时钟 = 16bit times</li> <li>● 一个 60MHz 的 PHY 时钟 = 8bit times</li> </ul>	Host; Device	3'h0	R_W

### 复位寄存器 (GRSTCTL)

◆ 偏移地址: 010h

域	说明	工作模式	复位值	访问特性
31	.AHB Master Idle(AHBIde) 指示 AHB Master 状态机处于 IDLE 状态	Host; Device	1'b1	RO
30	DMA 请求信号 (DMAReq) 指示现在总线上有个 DMA 请求; 调试使用	Host; Device	1'b0	RO
29:11	保留域	Host; Device		RO
10:6	TxFIFO 号 (TxFNum) 指示必须用 TxFIFO Flush 位清空的 FIFO 号。	Host; Device	5'h0	R_W

	<p>直到控制器清空了 TxFIFO 位，TxFNum 才可修改。</p> <ul style="list-style-type: none"> <li>● 5'h0: <ul style="list-style-type: none"> <li>- host 模式下清空非周期性 TxFIFO</li> <li>- device 模式下清空 TxFIFO 0</li> </ul> </li> <li>● 5'h1: <ul style="list-style-type: none"> <li>- host 模式下清空周期性 TxFIFO</li> <li>- device 模式下清空 TxFIFO 1</li> </ul> </li> <li>● 5'h2: <ul style="list-style-type: none"> <li>- device 模式下清空 TxFIFO 2</li> </ul> </li> <li>● 5'h3: <ul style="list-style-type: none"> <li>- device 模式下清空 TxFIFO 3</li> </ul> </li> <li>...</li> <li>● 5'H10: <ul style="list-style-type: none"> <li>- 清空所有的 TxFIFO (不管是 host 模式下还是 device 模式下)</li> </ul> </li> </ul>			
5	<p><b>TxFIFO 清空 (TxFFIsh)</b></p> <p>用来选择性地清空一个或所有的 TxFIFO。但是有事务正在进行时不可进行此操作。</p> <p>软件必须在确认 OTG 控制器没有读或写 TxFIFO 时才可写此位。确认方式如下：</p> <p>读 – NAK 有效中断能确保 OTG 控制器没有读 FIFO；</p> <p>写 – GRSTCTL.AHBIdle 能确保 OTG 控制器没有写 FIFO</p> <p>清空 FIFO 的操作可以在以下情形下使用，比如 FIFO 需要重新配置或在共享 FIFO 模式与专用 FIFO 模式下转换以及禁用某个 device endpoint。</p> <p>软件必须等待 OTG 控制器将此位清 0 后才能继续其他操作。这大概需要 8 个时钟，且是使用 phy_clk 与 hclk 中比较慢的那个时钟。</p>	Host; Device	1'b0	R_WS_SC
4	<p><b>RxFIFO 清空 (RxFFIsh)</b></p> <p>用来清空整个 RxFIFO，同样有事务正在进行时不可进行此操作。</p> <p>软件必须在确认 OTG 控制器没有读或写 RxFIFO 时才可写此位。</p> <p>软件必须等待 OTG 控制器将此位清 0 后才能继续其他操作。这大概需要 8 个时钟，且是使用 phy_clk 与 hclk 中比较慢的那个时钟。</p>	Host; Device	1'b0	R_WS_SC

3	保留位	Host; Device		RO
2	Host Frame Counter Reset (FrmCntrRst) 软件通过写此位来重置 OTG 控制器内的帧数。 一旦帧数被重置，下一个 OTG 控制器发出的 SOF 的帧号为 0.	Host	1'b0	R_WS_S C
1	保留域	Host; Device		RO
0	OTG 控制器软重启 (CSftRst) 复位 hclk 与 phy_clock 时钟域的内容如下： <ul style="list-style-type: none"> <li>● 清空除以下列以外的所有中断以及 CSR 寄存器 <ul style="list-style-type: none"> <li>- PCGCCTL.RstPdown</li> <li>- PCGCCTL.GateHclk</li> <li>- PCGCCTL.PwrClmp</li> <li>- GUSBCFG.DDRSel</li> <li>- GUSBCFG.PHYSel</li> <li>- GUSBCFG.FSIntf</li> <li>- GUSBCFG.ULPI_UTMI_Sel</li> <li>- GUSBCFG.PHYIf</li> <li>- HCFG&gt;FSLSPclkSel</li> <li>- DCFG.DevSpd</li> <li>- GGPIO</li> <li>- GPWRDN</li> <li>- GADPCTL</li> </ul> </li> <li>● 所有模块的状态机（除 AHB Slave 单元）都被复位至 IDLE 状态；清空所有的 TxFIFO 与 Rx FIFO</li> <li>● 在平滑地完整最后一个 AHB 传输的数据相后，立即结束所有的 AHB Master 事务。立即结束所有的 USB 事务</li> <li>● PMU 模块（功耗管理单元）不会被复位</li> </ul> 软件在任意时刻都可软复位 OTG 控制器。这位是自清 0 的，即在所有必要的逻辑复位后控制器会清 0 此位。根据控制器当时的状态不同，这将花费数个时钟。一旦此位被清 0，软件必须等待 3 个 PHY 时钟后才能操作 PHY 域（同步延时）。软件在开始任何动作之前，必须确保 AHBIdle 位为 1。	Host; Device	1'b0	R_WS_S C

### 中断寄存器 (GINTSTS)

偏移地址：014h

域	说明	工作模式	复位值	访问特性
31	系统恢复 / 远程唤醒中断 (WkUpInt):Resume/Remote Wakeup Detected Interrupt 处于挂起(L2)状态时的唤醒中断。 <ul style="list-style-type: none"> <li>● Device 模式：仅当在 USB 总线上发现 Host 发起的唤醒操作时，发生此中断</li> <li>● Host 模式：仅当在 USB 总线上发现 device 发起的远程唤醒操作时，发生此中断</li> </ul>	Host; Device	1'b0	R_SS_WC
30	会话请求 / 发现新会话中断 ( SessReqInt ) :Session Request/New Session Detected Interrupt. <ul style="list-style-type: none"> <li>● 在 Host 模式下，当发现从 device 来的会话请求时，置 1.</li> <li>● 在 device 模式下，当 utmisrp_bvalid 信号拉高时，置 1.</li> </ul>	Host; Device	1'b0	R_SS_WC
29	断开中断 (DisconnInt) 当 device 断开时，置 1.	Host	1'b0	R_SS_WC
28	连接器 ID 状态变化 (ConIDStsChng) 当连接器 ID 状态有变化时，置 1.	Host; Device	1'b1	R_SS_WC
27	保留位	Host; Device		RO
26	周期性 Tx FIFO 空 (PTxFEmp) 当周期性 Tx FIFO 是半空或全空，且在周期性请求队列中至少有一项请求时，此位置 1.是半空还是全空时置 1 由 GAHBCFG.PTxFEmpLvl 来决定。	Host	1'b1	RO
25	Host Channels Interrupt (HChInt) 在 host 模式下，当 OTG 控制器里某一个通道上有一个等待处理的中断时，置 1.软件必须读 HAINTE 寄存器来确定有未处理中断的 channel 号，然后读取对应的 HCINTn 寄存器来确定发生中断的原因。为了将这位清 0，软件必须先将 HCINTn 寄存器中对应状态位清 0.	Host	1'b0	RO
24	Host 端口中断 (PrtInt) 在 Host 模式下，若 OTG 控制器的某个端口状态发生变化时，置 1.软件必须读 HPRT 寄存器来确定引起这个中断的真正原因。为了将这位	Host	1'b0	RO

	清 0, 软件必须先将 HPRT 寄存器中对应状态位清 0.			
23	<p>复位中断 (ResetDet): Reset Detected Interrupt</p> <p>在 device 模式下, 处于挂起状态下的 OTG 控制器发下一复位操作时, 置 1.</p> <p>仅当 OTG 控制器处于 device 模式, 且工作在 partial Power-down, 或是门控挂起状态下才有效。</p> <p>当 OTG 控制器工作在 Hibernation 挂起状态时, 这位无效。</p>	Device	1'b0	R_SS_WC
22	<p>数据获取挂起 (FetSusp): Data Fetch Suspended</p> <p>仅在 DMA 模式下有效。当 TxFIFO 或请求队列没有空间时 OTG 会停止为 IN endpoint 取数据, 这时置 1. 软件在 endpoint 不匹配算法中使用此中断。</p>	Device	1'b0	R_SS_WC
21	<p>未完成周期性传输(incomplIP)</p> <p>Host 模式下, 当有一未完成的周期性事务仍等待在当前帧时间内处理时, 置 1.</p> <p>未完成的实时 OUT 传输 (incomplSOOUT)</p> <p>device 模式下, 若在当前帧时间内, 至少有一个实时 OUT endpoint 上有一个未完成的传输时, 置 1.同时置 EOPF 位</p>	Host  Device	1'b0	R_SS_WC
20	<p>未完成的实时 IN 传输(incomplSOIN)</p> <p>若在当前帧时间内, 至少有一个实时 IN endpoint 上有一个未完成的传输时, 置 1.同时置 EOPF 位</p>	Device	1'b0	R_SS_WC
19	<p>OUT endpoint 中断(OEPInt)</p> <p>当在 OUT endpoint 上有一等待处理的中断时, OTG 控制器置 1.软件必须读 DAIN T 寄存器从而确定具体是哪个 endpoint 上有等待处理的中断; 然后读 DOEPINTn 寄存器从而确定引起这个中断的原因。为了将这位清 0, 软件必须先将 DOEPINTn 寄存器中对应状态位清 0.</p>	Device	1'b0	RO
18	<p>IN endpoint 中断(IEPInt)</p> <p>当在 IN endpoint 上有一等待处理的中断时, OTG 控制器置 1.软件必须读 DAIN T 寄存器从而确定具体是哪个 endpoint 上有等待处理的中断; 然后读 DOEPINTn 寄存器从而确定引起</p>	Device	1'b0	RO

	这个中断的原因。为了将这位清 0，软件必须先将 DOEPINTn 寄存器中对应状态位清 0。			
17	保留位	Host; Device		RO
16	保留位	Host; Device		RO
15	周期性帧结束(EOPF) 当在当前帧时间内，DCFG.PerFrInt 指示的阈值达到时，置 1。	Device	1'b0	R_SS_W C
14	实时的 OUT 包结束 (ISOOutDrop) 若 RxFIFO 没有足够的空间来为实时 OUT endpoint 存放一 maximum package size packet，则 OTG 控制器将无法将一个实时 OUT 包写入到 RxFIFO。当发生这种情况时，置此位为 1。	Device	1'b0	R_SS_W C
13	枚举结束 (EnumDone): Enumeration 当速度枚举结束是，置 1.软件读取 DSTS 寄存器获取枚举的确切速度	Device	1'b0	R_SS_W C
12	USB 复位 (USBRst) 当 USB 总线上有一复位操作时，置 1	Device	1'b0	R_SS_W C
11	USB 挂起 (USBSusp) 当 OTG 控制器检测到 USB 总线上有一挂起操作时，置 1.若在一段较长的时间内 utmi_linestate 信号线上没有活动，那么 OTG 控制器将进去挂起状态	Device	1'b0	R_SS_W C
10	Early Suspend(ErlySusp) 当检测到 USB 已进入空闲状态超过 3ms 时，OTG 控制器置此位为 1。	Device	1'b0	R_SS_W C
9:8	保留位	Host; Device		RO
7	全局 OUT NAK 有效 (GOUTNakEff) 软件置 DCTL.SGOUTNak 位后其开始有效时，置 1.通过清 0 DCTL.SGOUTNak 位达到清 0 此位的效果	Device	1'b0	RO
6	全局 IN 非周期性 NAK 有效(GINNakEff) 软件置 DCTL.SGNPInNak 位后其开始有效时，置 1. 通过清 0 DCTL. SGNPInNak 位达到清 0 此位的效果	Device	1'b0	RO

5	保留位	Host; Device		RO
4	RxFIFO 非空 (RxFLvl) 当 RxFIFO 中至少有一个包时, 置 1.	Host; Device1	1'b0	RO
3	帧开始(Sof) Host 模式下, 当 SOF 在 USB 总线上传输时, OTG 控制器将此位置 1.软件写 1 将此位清 0. Device 模式下, 当 OTG 控制器收到 SOF 标志 (token) 时, 置 1.软件可通过读 Device status 寄存器来获取当前帧的帧号。	Host; Device	1'b0	R_SS_W C
2	OTG 中断(OTGInt) 置 1 指示发生了 OTG 协议事件。软件可读 GOTGINT 寄存器来确定发生中断的原因。为了将这位清 0, 软件必须清空 GOTGINT 寄存器相应的状态位。	Host; Device	1'b0	RO
1	模式不匹配中断(ModeMis) 当软件试图进行一下操作时, 置 1. ● 当 OTG 工作在 device 模式下时, 试图访问 host 模式下的寄存器。 ● 当 OTG 工作在 Host 模式下时, 试图访问 device 模式下的寄存器。	Host; Device	1'b0	R_SS_W C
0	当前工作模式(CurMod) 指示当前 OTG 控制器的工作模式 ● 1'b0: device 模式 ● 1'b1: host 模式	Host; Device	1'b0	RO

### 中断屏蔽寄存器 (GINTMSK)

偏移地址: 018h

这个寄存器与 GINTSTS 寄存器共同使用。当某个中断被屏蔽, 那么这个中断不会被软件看到。当然, GINTSTS 对应的位仍旧置位。

域	说明	工作模式	复位值	访问特性
31	系统恢复 / 远程唤醒中断屏蔽位 (WkUpIntMsk):Resume/Remote Wakeup Detected Interrupt Mask	Host; Device	1'b0	R_SS_WC
30	会话请求 / 发现新会话 中断屏蔽位 ( SessReqIntMsk ) :Session Request/New Session Detected Interrupt Mask.	Host; Device	1'b0	R_SS_WC
29	断开中断屏蔽位 (DisconnIntMsk)	Host	1'b0	R_SS_WC

28	连接器 ID 状态变化屏蔽位 (ConIDStsChngMsk)	Host; Device	1'b1	R_SS_WC
27	保留位	Host; Device		RO
26	周期性 Tx FIFO 空屏蔽位 (PTxFEmpMsk)	Host	1'b1	RO
25	Host Channels Interrupt 屏蔽位(HChIntMsk)	Host	1'b0	RO
24	Host 端口中断屏蔽位 (PrtIntMsk)	Host	1'b0	RO
23	复位中断屏蔽位 (ResetDetMsk): Reset Detected Interrupt Mask	Device	1'b0	R_SS_WC
22	数据获取挂起屏蔽位 (FetSuspMsk): Data Fetch Suspended Mask	Device	1'b0	R_SS_WC
21	未完成周期性传输屏蔽位(incomplIPMsk) 未完成的实时 OUT 传输屏蔽位 (incomplISOOUTMsk)	Host Device	1'b0	R_SS_WC
20	未完成的实时 IN 传输屏蔽位 (incomplISOINMsk)	Device	1'b0	R_SS_WC
19	OUT endpoint 中断屏蔽位(OEPIntMsk)	Device	1'b0	RO
18	IN endpoint 中断屏蔽位(IEPIntMsk)	Device	1'b0	RO
17:16	保留位	Host; Device		RO
15	周期性帧结束屏蔽位(EOPFMsk)	Device	1'b0	R_SS_WC
14	实时的 OUT 包结束屏蔽位(ISOOutDropMsk)	Device	1'b0	R_SS_WC
13	枚举结束屏蔽位 (EnumDoneMsk): Enumeration Mask	Device	1'b0	R_SS_WC
12	USB 复位屏蔽位 (USBRstMsk)	Device	1'b0	R_SS_WC
11	USB 挂起屏蔽位 (USBSuspMsk)	Device	1'b0	R_SS_WC
10	Early Suspend 屏蔽位(ErlySuspMsk)	Device	1'b0	R_SS_WC
9:8	保留位	Host; Device		RO
7	全局 OUT NAK 有效屏蔽位 (GOUTNakEffMsk)	Device	1'b0	RO
6	全局 IN 非周期性 NAK 有效屏蔽位 (GINNakEffMsk)	Device	1'b0	RO
5	保留位	Host; Device		RO
4	RxFIFO 非空屏蔽位 (RxFLvIMsk)	Host;	1'b0	RO

		Device1		
3	帧开始屏蔽位(SofMsk)	Host; Device	1'b0	R_SS_WC
2	OTG 中断屏蔽位(OTGIntMsk)	Host; Device	1'b0	RO
1	模式不匹配中断屏蔽位(ModeMisMsk)	Host; Device	1'b0	R_SS_WC
0	当前工作模式(CurModMsk)	Host; Device	1'b0	RO

### 接收状态调试寄存器/状态 read 以及 pop 寄存器 (GRXSTSR/GRXSTSP)

读偏移地址: 01Ch

Pop 偏移地址: 020h

对接收状态调试寄存器的读操作返回的是 RxFIFO 的第一项内容。对状态 read 以及 pop 寄存器的读操作额外还会将 RxFIFO 的第一项内容 pop 出来。

接收状态寄存器的内容在 host 模式与 device 模式下有着不同的解释。若 RxFIFO 空, 那么 OTG 控制器将忽略对这个寄存器的读以及 pop 操作, 且返回 32'h0000\_0000。当 GINTSTS.RxFLvl 位为 1 时, 软件仅能 pop the receive Status FIFO。

NOTE: OTG 在不同的工作模式下, 此寄存器的各个域有着不同的解释

Host 模式

域	说明	复位值	访问特性
31:21	保留域	1	RO
20:17	包状态 (PktSts) 指示接收包的状态: <ul style="list-style-type: none"> <li>● 4'b0010: 接收到 IN 数据包</li> <li>● 4'b0011: IN 传输结束 (触发中断)</li> <li>● 4'b0101: 数据 toggle 错误(触发中断)</li> <li>● 4'b0111: 通道 halt(触发中断)</li> <li>● 其他: 保留</li> </ul>	4'b0	RO
16:15	数据 PID (DPID) 指示接收包的数据 PID <ul style="list-style-type: none"> <li>● 2'b00:DATA0</li> <li>● 2'b10:DATA1</li> <li>● 2'b01:DATA2</li> <li>● 2'b11:MDATA</li> </ul>	2'b0	RO
14:4	字节数 (BCnt) 指示接收到的 IN 数据包的字节数	11'h0	RO
3:0	通道号 (ChNum) 指示当前接收到的数据包所属的通道号	4'h0	RO

### Device 模式

域	说明	复位值	访问特性
---	----	-----	------

31:25	保留域		RO
24:21	帧号 (FN) 接收包所在帧的帧号最低四位。 仅当支持实时 OUT 端点时, 此域有效	4'h0	RO
20:17	包状态 (PktSts) 指示接收包的状态: <ul style="list-style-type: none"> <li>● 4'b0001: 全局 OUT NAK (触发中断)</li> <li>● 4'b0010: 接收到 OUT 数据包</li> <li>● 4'b0011: OUT 传输结束(触发中断)</li> <li>● 4'b0100: SETUP 事务结束(触发中断)</li> <li>● 4'b0110: 接收到 SETUP 数据包</li> <li>● 其他: 保留</li> </ul>	4'b0	RO
16:15	数据 PID (DPID) 指示接收到的 OUT 包的数据 PID <ul style="list-style-type: none"> <li>● 2'b00:DATA0</li> <li>● 2'b10:DATA1</li> <li>● 2'b01:DATA2</li> <li>● 2'b11:MDATA</li> </ul>	2'b0	RO
14:4	字节数 (BCnt) 指示接收到的 IN 数据包的字节数	11'h0	RO
3:0	端点号 (EPNum) 指示当前接收到的数据包所属的端点号	4'h0	RO

### RxFIFO 容量寄存器 (GRXFSIZ)

读偏移地址: 024h

软件用此寄存器分配分配给 RxFIFO 的 RAM 大小。

域	说明	复位值	访问特性
31:16	保留域		RO
15:0	RxFIFO 深度(RxFDep) 以 32-bits 的字为单位。 <ul style="list-style-type: none"> <li>● 最小值为 16</li> <li>● 最大值为 32768</li> </ul>	533	R_W

### 非周期性 Tx FIFO 容量寄存器 (GNPTXFSIZ)

读偏移地址: 028h

软件用此寄存器分配分配给非周期性 Tx FIFO 的 RAM 大小以及开始地址。

Note: OTG 在 host 工作模式下, 还是在 device 工作模式下, 此寄存器有不同的解释。

Host 模式下

域	说明	复位值	访问特性
31:16	非周期性 Tx FIFO 深度 (NPTxDep)	256	R_W

	以 32-bits 的字为单位。 <ul style="list-style-type: none"> <li>● 最小值为 16</li> <li>● 最大值为 32768</li> </ul>		
15:0	非周期性 TxRAM 开始地址 (NPTxStAddr)	533	R_W

Device 模式下

域	说明	复位值	访问特性
31:16	IN 端点 TxFIFO 0 的深度。仅当 Device 模式下，且 OTG_EN_DED_TX_FIFO = 1 时有效。 <ul style="list-style-type: none"> <li>● 最小值为 16</li> <li>● 最大值为 32768</li> </ul>	32	R_W
15:0	IN 端点 FIFO0 TxRAM 开始地址	533	R_W

**Synopsys ID 寄存器 (GSNPSID)**

读偏移地址: 040h

包含 OTG 控制器的版本号

域	说明	复位值	访问特性
31: 0	Synopsys ID (SynopsysID) OTG 控制器的版本号	32'h4F54	RO

**User HW Config1 寄存器**

读偏移地址: 044h

用户硬件配置寄存器 2, 指定 endpoint 的方向。

域	说明	复位值	访问特性
31: 14	保留域	18'b0	RO
13: 0	Bits[13:12]: endpoint 6 方向 Bits[11:10]: endpoint 5 方向 Bits[9:8]: endpoint 4 方向 Bits[7:6]: endpoint 3 方向 Bits[5:4]: endpoint 2 方向 Bits[3:2]: endpoint 1 方向 Bits[1:0]: endpoint 0 方向 <ul style="list-style-type: none"> <li>● 2'b00: 双向 endpoint</li> <li>● 2'b01: IN endpoint</li> <li>● 2'b10: OUT endpoint</li> <li>● 2'b11: 保留</li> </ul>	14'b0	RO

**User HW Config2 寄存器**

读偏移地址: 048h

用户硬件配置寄存器 2

域	说明	复位值	访问特性
31: 26	保留域	6'b0	RO

25: 24	Host 模式下, 周期性请求队列的深度 <ul style="list-style-type: none"> <li>● 2'b00: 2</li> <li>● 2'b01: 4</li> <li>● 2'b10: 8</li> <li>● 2'b11: 16</li> </ul>	2'b10	RO
23: 22	非周期性请求队列的深度 <ul style="list-style-type: none"> <li>● 2'b00: 2</li> <li>● 2'b01: 4</li> <li>● 2'b10: 8</li> <li>● 2'b11: 16</li> </ul>	2'b10	RO
21	保留域		RO
20	使能多个处理器中断 <ul style="list-style-type: none"> <li>● 1'b0: 否</li> <li>● 1'b1: 是</li> </ul>	1'b0	RO
19	使能动态 FIFO 大小 <ul style="list-style-type: none"> <li>● 1'b0: 否</li> <li>● 1'b1: 是</li> </ul>	1'b1	RO
18	使能周期性 OUT channel <ul style="list-style-type: none"> <li>● 1'b0: 否</li> <li>1'b1: 是</li> </ul>	1'b1	RO
17:14	Host 模式下的 Channel 数	4'd12	RO
13:10	Device 模式下的端点数, 除 endpoint 0 外	4'd6	RO
9: 8	全速 PHY 接口类型 <ul style="list-style-type: none"> <li>● 2'b00: 无</li> <li>● 2'b01: FS 有专用 pin</li> <li>● 2'b10: 与 UTMI+共用</li> <li>● 2'b11: 与 ULPI 共用</li> </ul>	2'b00	RO
7:6	HS PHY 接口类型 <ul style="list-style-type: none"> <li>● 2'b00: 无</li> <li>● 2'b01: UTMI+</li> <li>● 2'b10: ULPI</li> <li>● 2'b11: UTMI+与 ULPI</li> </ul>	2'b01	RO
5	点对点 <ul style="list-style-type: none"> <li>● 1'b0: 否</li> <li>● 1'b1: 是</li> </ul>	1'b1	RO
4: 3	结构 <ul style="list-style-type: none"> <li>● 2'b00: 仅能做从设备</li> <li>● 2'b01: 外部 DMA</li> <li>● 2'b10: 内嵌 DMA</li> <li>● 2'b11: 保留</li> </ul>	2'b10	RO

2:0	<p>操作模式</p> <ul style="list-style-type: none"> <li>● 3'b000: 支持 HNP 与 SRP 的 OTG (host 与 device)</li> <li>● 3'b001: 仅支持 SRP 的 OTG (host 与 device)</li> <li>● 3'b010: 不支持 HNP 与 SRP 的 OTG (host 与 device)</li> <li>● 3'b011: 支持 SRP 的 device</li> <li>● 3'b100: 非 OTG 的 device</li> <li>● 3'b101: 支持 SRP 的 device</li> <li>● 3'b110: 非 OTG 的 host</li> <li>● 其他: 保留</li> </ul>	3'b000	RO
-----	--	--------	----

### User HW Config3 寄存器

读偏移地址: 04Ch

用户硬件配置寄存器 3

域	说明	复位值	访问特性
31: 16	<p>DFIFO 的深度</p> <ul style="list-style-type: none"> <li>● 最小值: 32</li> <li>● 最大值: 32768</li> </ul>	16'd3072	RO
15	<p>OTG_ENABLE_LPM 指定 OTG 是否支持 LPM 模式</p> <ul style="list-style-type: none"> <li>● 1'b0: 否</li> <li>● 1'b1: 是</li> </ul>	1'b0	RO
14	<p>OTG_BC_SUPPOT 指定是否支持 USB 充电功能</p> <ul style="list-style-type: none"> <li>● 1'b0: 否</li> <li>● 1'b1: 是</li> </ul>	1'b0	RO
13	<p>OTG_ENABLE_HSIC 指定是否支持 HSIC</p> <ul style="list-style-type: none"> <li>● 1'b0: 否</li> <li>● 1'b1: 是</li> </ul>	1'b0	RO
12	<p>OTG_ADP_SUPPORT 指定是否支持 ADP</p> <ul style="list-style-type: none"> <li>● 1'b0: 否</li> <li>● 1'b1: 是</li> </ul>	1'b0	RO
11	<p>OTG_SYNC_RESET_TYPE 指定是否支持同步复位</p> <ul style="list-style-type: none"> <li>● 1'b0: 否</li> <li>● 1'b1: 是</li> </ul>	1'b0	RO
10	去掉一些非必须的特性(OptFeature)	1'b1	RO

	<ul style="list-style-type: none"> <li>● 1'b0: 否</li> <li>● 1'b1: 是</li> </ul>		
9	是否支持 vendor control 接口 (VndctlSupt) 此接口用于访问 PHY 的内部寄存器。 <ul style="list-style-type: none"> <li>● 1'b0: 否</li> <li>● 1'b1: 是</li> </ul>	1'b0	RO
8	是否支持 I2C 接口 (I2CIntSel) <ul style="list-style-type: none"> <li>● 1'b0: 否</li> <li>● 1'b1: 是</li> </ul>	1'b0	RO
7	OTG 功能使能 (OtgEn) 指定是否使能 OTG 功能 <ul style="list-style-type: none"> <li>● 1'b0: 否</li> <li>● 1'b1: 是</li> </ul>	1'b1	RO
6: 4	Packet Counters 的宽度 (PktSizeWidth) 表示在一次传输中 USB 控制器 send/receive 的 Packet 的最大个数。 <ul style="list-style-type: none"> <li>● 3'b000: 4bits</li> <li>● 3'b001: 5bits</li> <li>● 3'b010: 6bits</li> <li>● 3'b011: 7bits</li> <li>● 3'b100: 8bits</li> <li>● 3'b101: 9bits</li> <li>● 3'b110: 10bits</li> <li>● 其他: 保留</li> </ul>	3'b110	RO
3: 0	Transfer Size Counters 的宽度 (XferSizeWidth), 表示最大传输数据大小 <ul style="list-style-type: none"> <li>● 4'b0000: 11bits</li> <li>● 4'b0001: 12bits</li> <li>...</li> <li>● 4'b1000: 19bits</li> <li>● 其他: 保留</li> </ul>	4'b1000	RO

### User HW Config4 寄存器

读偏移地址: 050h

用户硬件配置寄存器 4

域	说明	复位值	访问特性
31	保留		RO
30	是否支持 Scatter/Gather DMA <ul style="list-style-type: none"> <li>● 1'b0: 否</li> <li>● 1'b1: 是</li> </ul>	1'b1	RO
29:26	Device 模式下, IN endpoint 的个数, 包括	4	RO

	<p>endpoint0 (INEps)</p> <ul style="list-style-type: none"> <li>● 0: 1 个 IN endpoint</li> <li>● 1: 2 个 IN endpoint</li> <li>● ...</li> <li>● 16: 16 个 IN endpoint</li> </ul>		
25	<p>对 Device In endpoint 使能专用的 Tx FIFO</p> <ul style="list-style-type: none"> <li>● 1'b0: 否</li> <li>● 1'b1: 是</li> </ul>	1'b1	RO
24	<p>是否加一个对 session_end 信号的去抖逻辑</p> <ul style="list-style-type: none"> <li>● 1'b0: 否</li> <li>● 1'b1: 是</li> </ul>	1'b1	RO
23	<p>是否加一个对 b_valid 信号的去抖逻辑</p> <ul style="list-style-type: none"> <li>● 1'b0: 否</li> <li>● 1'b1: 是</li> </ul>	1'b1	RO
22	<p>是否加一个对 a_valid 信号的去抖逻辑</p> <ul style="list-style-type: none"> <li>● 1'b0: 否</li> <li>● 1'b1: 是</li> </ul>	1'b1	RO
21	<p>是否加一个对 vbus_valid 信号的去抖逻辑</p> <ul style="list-style-type: none"> <li>● 1'b0: 否</li> <li>● 1'b1: 是</li> </ul>	1'b1	RO
20	<p>是否加一个对 iddig_valid 信号的去抖逻辑</p> <ul style="list-style-type: none"> <li>● 1'b0: 否</li> <li>● 1'b1: 是</li> </ul>	1'b1	RO
19:16	<p>除去 endpoint0, device 模式下控制 endpoint 的个数</p>	4'b0	RO
15:14	<p>UTMI+ PHY 数据宽度</p> <ul style="list-style-type: none"> <li>● 2'b00: 8bits</li> <li>● 2'b01: 16bits</li> <li>● 2'b10: 8/16 bits 软件可选</li> <li>● 2'b11: 保留</li> </ul>	2'b10	RO
13:7	<p>保留域</p>		RO
6	<p>使能 Hibernation</p> <ul style="list-style-type: none"> <li>● 1'b0: 否</li> <li>● 1'b1: 是</li> </ul>	1'b0	RO
5	<p>AHB 频率是否最小不能低于 60MHA(AhbFreq)</p> <ul style="list-style-type: none"> <li>● 1'b0: 否</li> <li>● 1'b1: 是</li> </ul>	1'b0	RO
4	<p>是否使能降低部分功耗</p>	1'b1	RO

	<ul style="list-style-type: none"> <li>● 1'b0 :否</li> <li>● 1'b1: 是</li> </ul>		
3:0	保留域		RO

### DFIFO 软件配置寄存器 (GDFIFOCFG)

读偏移地址: 05Ch

域	说明	工作模式	复位值	访问特性
31:16	EPInfoBaseAddr Endpoint 控制器的起始地址	Host; Device	EPINFO_BASEADDR	R_W
15:0	GDFIFOCfg 动态配置 DFIFO 的大小	Host; Device	3072	R_W

### Host 周期性 Tx FIFO 容量寄存器 (HPTXFSIZ)

读偏移地址: 0100h

域	说明	复位值	访问特性
31:16	PTxFSize 周期性 Tx FIFO 深度, 以 32-bits 的字位单位。 <ul style="list-style-type: none"> <li>● 最小值: 16</li> <li>● 最大值: 32768</li> </ul>	512	R_W
15:0	PTxFStAddr Host 模式下周期性 Tx FIFO 的起始地址	789	R_W

### Device 模式下 IN 端点 Tx FIFO1 容量寄存器 (DIEPTXF1)

读偏移地址: 0104h

域	说明	复位值	访问特性
31:16	INEPnTxFDep IN 端点 Tx FIFO1 深度, 以 32-bits 的字位单位。 <ul style="list-style-type: none"> <li>● 最小值: 16</li> <li>● 最大值: 32768</li> </ul>	256	R_W
15:0	INEPnTxFStAddr IN 端点 FIFO 1 开始地址	565	R_W

### Device 模式下 IN 端点 Tx FIFO2 容量寄存器 (DIEPTXF2)

读偏移地址: 0108h

域	说明	复位值	访问特性
31:16	INEPnTxFDep IN 端点 TxFIFO2 深度，以 32-bits 的字位单位。 ● 最小值：16 ● 最大值：32768	256	R_W
15:0	INEPnTxFStAddr IN 端点 FIFO 2 开始地址	821	R_W

### Device 模式下 IN 端点 TxFIFO3 容量寄存器 (DIEPTXF3)

读偏移地址：010Ch

域	说明	复位值	访问特性
31:16	INEPnTxFDep IN 端点 TxFIFO3 深度，以 32-bits 的字位单位。 ● 最小值：16 ● 最大值：32768	512	R_W
15:0	INEPnTxFStAddr IN 端点 FIFO 3 开始地址	1077	R_W

### 13.3.3 Host 模式寄存器

这组寄存器仅在 Host 模式下有效；在 Device 模式下不可访问这组寄存器。

### Host 模式配置寄存器(HCFG)

偏移地址：400h

域	说明	复位值	访问特性
31:28	保留域		RO
27	ModeChTimEn 是否使能模式转换准备计时器。 用来决定是否使能 host 控制器在 Resume 状态结束时，等待 200PHY 时钟以改变对 PHY 的 opmode 信号从 2'b10 到 00 ● 1'b0: 否 ● 1'b1: 是	1'b0	R_W
26	PerSchedEna 是否使能周期调度。 ● 1'b0: 否 ● 1'b1: 是	1'b0	R_W

25:24	FrListEn Frame list 项数 <ul style="list-style-type: none"> <li>● 2'b00: 8</li> <li>● 2'b01: 16</li> <li>● 2'b00: 32</li> <li>● 2'b00: 64</li> </ul>	2'b00	R/W
23	DescDMA Host 模式下，使能 Scatter/gather DMA 注意，在复位之后，这位仅能背修改一次。以下配置是可用的： <ul style="list-style-type: none"> <li>● GAHBCFG.DMAEn = 0, HCFG.DescDMA = 0 =&gt; slave 模式；</li> <li>● GAHBCFG.DMAEn = 0, HCFG.DescDMA = 1 =&gt; 非法配置；</li> <li>● GAHBCFG.DMAEn = 1, HCFG.DescDMA = 0 =&gt; Buffered DMA 模式；</li> <li>● GAHBCFG.DMAEn = 0, HCFG.DescDMA = 1 =&gt; Scatter/Gather 模式；</li> </ul>	1'b0	RO
15:0	保留域		RO

### Host 帧间隔寄存器(HFIR)

偏移地址：404h

域	说明	复位值	访问特性
31:17	保留域		RO
16	Reload 控制 是否允许动态重新载入 HFIR 寄存器。 <ul style="list-style-type: none"> <li>● 1'b0: 否</li> <li>● 1'b1: 是</li> </ul>	1'b0	R_W
15:0	帧间隔 (FrInt) 软件用这个值来制定两个连续 micro-SOFs 之间的间隔。以 PHY 时钟位单位。仅当 HPRT.PrtEnaPort 置 1 后才能更改此位。在最初配置后，不能再更改。 125us * (PHY 时钟频率)	16'd60000	R_W

### Host 帧号/剩余帧时间寄存器 (HFNUM)

读偏移地址：408h

域	说明	复位值	访问特性
31:16	剩余帧时间 指示当前 micorFrame 的剩余时间，以 PHY 时钟为单位。当为 0 时，重新载入 HFIR.FrInt 的值，并在	16'h0	RO

	USB 总线上发送一个新的 SOF		
15:0	帧号 (FrNum) 当一个新的 SOF 传输时, 此域加 1, 当达到 16'h3FFF 时复位为 0.	16'h3FFF	RO

### Host 周期性 Tx FIFO/Queue 状态寄存器 (HPTXSTS)

偏移地址: 410h

域	说明	复位值	访问特性
31:24	PTxQTop 周期性 Tx FIFO 请求队列的第一项 (TOP) 用作调试。 <ul style="list-style-type: none"> <li>● Bits[31]: 奇偶帧 <ul style="list-style-type: none"> <li>- 1'b0: 在偶帧发送</li> <li>- 1'b1: 在奇帧发送</li> </ul> </li> <li>● Bits[30:27]: channel 号</li> <li>● Bits[26:25]: 类型 <ul style="list-style-type: none"> <li>- 2'b00: IN/OUT</li> <li>- 2'b01: 0 长度包</li> <li>- 2'b10: CSPLIT</li> <li>- 2'b11: 无效 channel 命令</li> </ul> </li> <li>● Bits[24]: 终止 (对选定的 channel 来讲为最后一项)</li> </ul>	8'h0	RO
23:16	PTxQSpAvail 周期性 Tx 请求队列可用项数。 指示周期性请求队列中的空项数, 请求队列中包含 IN 与 OUT <ul style="list-style-type: none"> <li>● 8'h0: 队列满</li> <li>● 8'h1: 1 个空项</li> <li>● 8'h2: 2 个空项</li> <li>○ ○ ○ ○</li> <li>● 8'h8: 8 个空项</li> <li>● 其他: 保留</li> </ul>	8	RO
15:0	PTxFSpcAvail 周期性 Tx FIFO 可用项数。 以 32-bits 的字为单位。 <ul style="list-style-type: none"> <li>● 16'h0: FIFO 满</li> <li>● 16'h1: 1 个空项</li> <li>● ...</li> <li>● 16'h200: 512 个空项</li> </ul>	512	RO

### Host 所有 channel 中断寄存器 (HAINT)

读偏移地址：414h

域	说明	复位值	访问特性
31:16	保留域		RO
15:0	Channel 中断(HAINT) 一个 bit 对应一个 channel: Bits 11: channel 11 ... Bits0 : channel 0	16'h0	RO

### Host 所有 channel 中断屏蔽寄存器 (HAINTMSK)

读偏移地址：414h

域	说明	复位值	访问特性
31:16	保留域		RO
15:0	Channel 中断屏蔽位(HAINTMSK) 一个 bit 对应一个 channel: Bits 11: channel 11 ... Bits 0: channel 0	16'h0	R_W

### Host 端口控制与状态寄存器 (HPRT)

读偏移地址：440h

域	说明	复位值	访问特性
31:19	保留域		RO
18:17	PrtSpd 端口速度 ● 2'b00: 高速 ● 2'b01: 全速 ● 2'b10: 低速 ● 2'b11: 保留	2'b0	RO
16:13	PrtTstCtl 端口测试控制 当软件往此域写非 0 值时，使此端口进入测试模式。 相应的 Pattern 出现端口上。 ● 4'b0000: 非测试模式 ● 4'b0001: Test_J 模式 ● 4'b0010: Test_K 模式 ● 4'b0011: Test_SE0_NAK 模式 ● 4'b0100: Test_Packet 模式 ● 4'b0101: Test_Force 模式	4'h0	R_W

	● 其他： 保留		
12	PrtPwr 端口供电。 ● 1'b0: power off ● 1'b1: power on	1'b0	R_W_SC
11:10	Port Line Status(PrtLnSts) ● Bits[10]: D+ ● Bits[11]: D-	1'b0	RO
9	保留		RO
8	端口复位 (PrtRst)	1'b0	R_W
7	端口挂起 (PrtSup)	1'b0	R_WS_SC
6	端口恢复 (PrtRes)	1'b0	R_W_SS_SC
5	端口过载改变(PrtOvrCurrChng) 当 PrtOvrCurrAct 位改变时, OTG 控制器将这位置 1	1'b0	R_SS_WC
4	端口过载活动 (PrtOvrCurrAct) ● 无 ● 有	1'b0	RO
3	端口使能状态改变(PrtEnChng) 当 PrtEna 位改变时, OTG 控制器将这位置 1	1'b0	R_SS_WC
2	端口使能位 (PrtEna) ● 1'b0: 否 ● 1'b1: 是	1'b0	R_SS_SC_WC
1	发现端口连接 (PrtConnDet) 当发现设备连接时, 将通过 GINTSTS.PrtInt 触发中断。软件写 1 将此位清 0.	1'b0	R_SS_WC
0	端口连接状态 (PrtConnSts) ● 无设备连接到此端口 ● 有设备连接到此端口	1'b0	RO

### Host Channel-n 特性寄存器(HCCHARn)

Channel\_num:  $0 \leq n \leq 11$

读偏移地址:  $500h + (\text{Channel\_num} * 20h)$

域	说明	复位值	访问特性
31	Channel 使能 (ChEna) 当 Scatter/Gather 模式使能时 ● 1'b0: 指示描述符结构还未准备好 ● 1'b1: 指示描述符结构以及数据缓冲已经准备好, 这个通道可以访问描述符 当 Scatter/Gather 模式未使能时: ● 1'b0: 通道失效 ● 1'b1: 通道使能	1'b0	R_WS_SC

30	通道失效 (ChDis) 即使此通道上的传输还未完成, 软件也可通过将这位置 1 来停止在此通道上的数据发送与接收。软件必须等待通道失效中断发生后, 才能认为此通道已失效。	1'b0	R_WS_SC_SS
29	奇帧(OddFrm) 软件通过将此位置 1 来告知 OTG host 必须在奇数 micro Frame 内完成一个传输。此域仅对周期性事务 (实时与中断) 有效 ● 1'b0: 偶帧 ● 1'b1: 奇帧	1'b0	R_W
28:22	设备地址 (DevAddr)	7'h0	R_W
21:20	Multi Count (MC) /Error Count(EC) 当 HCSPLTn.SplitEna 为 1'b0 时, 此域指示对于每个周期性的 endpoint, 在每个 microFrame 时间内完成的事务数。对于非周期性传输, 此域尽在 DMA 模式下使用, 用来指定在 DMA 改变仲裁之前, host 必须为此通道获取的包数。 ● 2'b00: 保留 ● 2'b01: 1 个事务 ● 2'b10: 2 个事务 ● 2'b11: 3 个事务 当 CSPLTn.SplitEna 为 1'b1 时, 此域指定对于一个周期性 split 事务出错时, 必须重传的次数。至少设置为 2'b01	2'b0	R_W
19:18	端点类型 (EPTyPe) ● 2'b00: 控制 ● 2'b01: 实时 ● 2'b10: 批量 ● 2'b11: 中断	2'b0	R_W
17:16	保留		RO
15	端点方向(EPDir) ● 1'b0: OUT ● 1'b1: IN	1'b0	R_W
14:11	端点号	4'b0	R_W
10:0	最大的包容量 (MPS)	11'h0	R_W

### Host Channel-n Split 控制寄存器(HCSPLTn)

Channel\_num:  $0 \leq n \leq 11$

读偏移地址:  $504h + (\text{Channel\_num} * 20h)$

域	说明	复位值	访问特性
31	Split 使能	1'b0	R_W
30:17	保留		RO
16	Do Complete Split 指示 OTG host 必须完成一个完整的 split 传输	1'b0	R_W
15:14	事务位置(XactPos) <ul style="list-style-type: none"> <li>● 2'b00; all, 此事务的整个数据 (少于或等于 188B)</li> <li>● 2'b01: begin, 此事务的开始数据 (少于或等于 188B)</li> <li>● 2'b10: Mid, 此事务的中间数据 (少于或等于 188B)</li> <li>● 2'b01: End, 此事务的最后数据 (少于或等于 188B)</li> </ul>	2'h0	R_W
13:7	Hub 地址 (HubAddr)	7'b0	R_W
6: 0	端口地址(PrtAddr)	7'h0	R_W

### Host Channel-n 中断寄存器(HCINTn)

Channel\_num:  $0 \leq n \leq 11$

读偏移地址:  $508h + (\text{Channel\_num} * 20h)$

域	说明	复位值	访问特性
31:14	保留		RO
13	描述符 rollover 中断(DESC_LST_ROLLIntr) 仅当 Scatter/Gather DMA 模式使能时, 此域有效。当对应通道的描述符列表 roll over 时置 1。 对于非 Scatter/Gather DMA 模式, 此域位保留域	1'b0	R_SS_WC
12	过多事务错 (XCS_XACT_ERR) 当 Scatter/Gather DMA 模式使能时, 此域有效。当 3 个连续的事务出错时时置 1。 对于非 Scatter/Gather DMA 模式, 此域位保留域	1'b0	R_SS_WC
11	BNA(buffer 不可用)中断(BNAIntr) 当 Scatter/Gather DMA 模式使能时, 此域有效。当描述符访问还未准备好时置 1。 对于非 Scatter/Gather DMA 模式, 此域位保留域	1'b0	R_SS_WC
10	数据反转位错(DataTglErr)	1'b0	R_SS_WC
9	帧 overrun (FrmOvrn)	1'b0	R_SS_WC
8	空泡错(BblErr)	1'b0	R_SS_WC

7	事务错 (XactErr) 当 USB 总线上发生了以下错误时置 1: <ul style="list-style-type: none"> <li>● CRC 验证失败</li> <li>● 超时</li> <li>● 位填充错</li> <li>● 错误的 EOP</li> </ul>	1'b0	R_SS_WC
6	收到 NYET 握手 (NYET)	1'b0	R_SS_WC
5	收到 ACK 握手 (ACK)	1'b0	R_SS_WC
4	收到 NAK 握手 (NAK)	1'b0	R_SS_WC
3	收到 STALL 握手 (STALL)	1'b0	R_SS_WC
2	AHB 错 (AHBErr)	1'b0	R_SS_WC
1	通道停止 (ChHltd) 对于非 Scatter/Gather DMA 模式, 由于 USB 传输错或软件置无效请求而引起传输的非正常结束, 或正常结束时置 1。 对于 Scatter/Gather DMA 模式, 若因为以下原因导致传输结束, 则置 1: <ul style="list-style-type: none"> <li>● 描述符中 EOL 置位</li> <li>● AHB 错</li> <li>● 过多的事务错</li> <li>● 软件置无效请求</li> <li>● 空泡</li> <li>● Stall</li> </ul>	1'b0	R_SS_WC
0	传输结束 (XferCompl) 对于 Scatter/Gather DMA 模式, 指示完成对当前描述符的处理, 其描述符的 IOC 位置 1。 对于非 Scatter/Gather DMA 模式, 表示没有任何错误的传输完成。	1'b0	R_SS_WC

### Host Channel-n 中断屏蔽寄存器 (HCINTMSKn)

Channel\_num:  $0 \leq n \leq 11$

读偏移地址:  $50Ch + (\text{Channel\_num} * 20h)$

域	说明	复位值	访问特性
31:14	保留		RO
13	描述符 rollover 中断屏蔽位 (DESC_LST_ROLLIntrMsk) 仅当 Scatter/Gather DMA 模式使能时, 此域有效。 对于非 Scatter/Gather DMA 模式, 此域位保留域	1'b0	R_W

12	保留		RO
11	保留		RO
10	数据反转位错中断屏蔽位(DataTglErrMsk)	1'b0	R_W
9	帧 overrun 中断屏蔽位 (FrmOvrnMsk)	1'b0	R_W
8	空泡错中断屏蔽位(BblErrMsk)	1'b0	R_W
7	事务错中断屏蔽位 (XactErrMsk)	1'b0	R_W
6	收到 NYET 握手中断屏蔽位 (NYETMsk)	1'b0	R_W
5	收到 ACK 握手中断屏蔽位 (ACKMsk)	1'b0	R_W
4	收到 NAK 握手中断屏蔽位(NAKMsk)	1'b0	R_W
3	收到 STALL 握手中断屏蔽位 (STALLMsk)	1'b0	R_W
2	AHB 错中断屏蔽位 (AHBErrMsk)	1'b0	R_W
1	通道停止中断屏蔽位(ChHltdMsk)	1'b0	R_W
0	传输结束中断屏蔽位 (XferComplMsk)	1'b0	R_W

### Host Channel-n 传输大小寄存器(HCTSIZn)

Channel\_num: 0 <= n <= 11

读偏移地址: 510h + (Channel\_num \* 20h)

在 Scatter/Gather DMA 模式下, 此寄存器各个域定义如下

域	说明	复位值	访问特性
31	Do ping(DoPng) 仅对 OUT 传输有效, 置 1 时, 指示 Host 做 PING 协议。 若对 IN 传输将此位置 1, 将使此通道无效	1'b0	R_W
30:29	PID(Pid) ● 2'b00:DATA0 ● 2'b01:DATA2 ● 2'b10:DATA1 ● 2'b11:MDATA (非控制)	2'b00	R_W
28:16	保留域	9'b0	RO
15:8	NTD(传输描述符的数目) (非实时传输) ● 0 – 1 个描述符 ● 63 – 64 个描述符 (实时传输) ● 7 – 8 个描述符 ● 15 – 16 个描述符 ● 31 – 32 个描述符 ● 63 – 64 个描述符	8'h0	R_W

	<ul style="list-style-type: none"> <li>● 127 – 128 个描述符</li> <li>● 255 – 256 个描述符</li> </ul>		
7:0	<p>SCHED_INFO (调度信息)</p> <p>对应的每一位代表着在那个 microFrame 中角度。Bit0 代表在 1st microFrame 中调度。Bits7 代表在 8th 中调度。8'b11111111 代表着对应的中断通道被调度，在对应帧时间内每个 microFrame 都发送一个标记。8'b10101010 代表对应的中断通道被调度，在对应帧时间内每隔一个 microFrame 发送一个标记。</p> <p>注意，此域仅对周期性传输有效。</p>	8'h0	R_W

在非 Scatter/Gather DMA 模式下，此寄存器各个域定义如下

域	说明	复位值	访问特性
31	<p>Do ping(DoPng)</p> <p>仅对 OUT 传输有效，置 1 时，指示 Host 做 PING 协议。若对 IN 传输将此位置 1，将使此通道无效</p>	1'b0	R_W
30:29	<p>PID(Pid)</p> <ul style="list-style-type: none"> <li>● 2'b00:DATA0</li> <li>● 2'b01:DATA2</li> <li>● 2'b10:DATA1</li> <li>● 2'b11:MDATA (非控制)</li> </ul>	2'b00	R_W
28:19	<p>包数目(PktCnt).</p> <p>软件写入传输的包数据。每当一个传输完成，host 将此域减 1。一旦为 0，软件将接收到中断表示正常结束。</p>	10'h0	R_W
18:0	<p>传输大小 (XferSize)</p> <p>对于 OUT 传输，指定 host 将发送的数据字节数；对于 IN 传输，指定 host 将接收到的数据字节数。</p>	19'h0	R_W

### Host Channel-n DMA 地址寄存器(HCDMA<sub>n</sub>)

Channel\_num: 0 <= n <= 11

读偏移地址: 514h + (Channel\_num \* 20h)

在 IN/OUT 传输中，此寄存器用来维持当前 buffer 的地址。DMA 传输的起始地址必须双字对齐。

域	说明	复位值	访问特性
<b>Buffer DMA Mode</b>			
31:0	<p>DMA 地址 (DMAAddr)</p> <p>每个 AHB 事务后累加。</p>	X	R_W
<b>描述符 DMA 模式</b>			

<p>31: N (实时) 31:9 (非实时)</p>	<p>DMA 地址 (DMAAddr) 非实时传输: 保存 512 字节页的起始地址。描述符列表的第一个描述符即在此地址上。OTG 控制器从 CTD 开始处理描述符列表 实时传输: 保存 2*(nTD+1)个字节的地址, 此地址为实时描述符所在位置。其中 N 的值基于 nTD 而来, 具体如下表:</p> <table border="1"> <tr> <td>31: N</td> <td>N-1: 3</td> <td>2:0</td> </tr> <tr> <td>基址</td> <td>偏移</td> <td>000</td> </tr> </table> <table border="1"> <tr> <td>nTD</td> <td>N</td> </tr> <tr> <td>7</td> <td>6</td> </tr> <tr> <td>15</td> <td>7</td> </tr> <tr> <td>31</td> <td>8</td> </tr> <tr> <td>63</td> <td>9</td> </tr> <tr> <td>127</td> <td>10</td> </tr> <tr> <td>255</td> <td>11</td> </tr> </table>	31: N	N-1: 3	2:0	基址	偏移	000	nTD	N	7	6	15	7	31	8	63	9	127	10	255	11	23'h0	R/W
31: N	N-1: 3	2:0																					
基址	偏移	000																					
nTD	N																						
7	6																						
15	7																						
31	8																						
63	9																						
127	10																						
255	11																						
<p>N-1: 3 (实时) 8: (非实时)</p>	<p>当前传输描述符 (CTD) 非实时传输: 以描述符为单位, 范围 0 – 63.指定当前正在处理的描述符。 实时传输: 基于当前 micorFrame 的值, 需被软件清 0</p>	6'h0	R/W																				
2:0	保留	3'h0	RO																				

### Host Channel-n DMA Buffer 地址寄存器(HCDMBAn)

Channel\_num: 0 <= n <= 11

读偏移地址: 518h + (Channel\_num \* 20h)

此仅在 Scatter/Gather DMA 模式下有效

域	说明	复位值	访问特性
31:0	当前 buffer 的地址。	X	RO

### Host Frame List 基地址寄存器(HFLBAddr)

Offset: 41Ch

此仅在 Scatter/Gather DMA 模式下有效

域	说明	复位值	访问特性
31:0	Frame list 的起始地址	32'h0	RO

### 13.3.4 Device 模式寄存器

这组寄存器仅在 device 模式下有效, host 模式下不可访问。

### Device 配置寄存器(DCFG)

Offset: 800h

在初始配置过后，不能修改此寄存器的值

域	说明	复位值	访问特性
31:26	Resume 有效期(ResValid) 此域控制 OTG 控制器从挂起状态到 resume 的时间。以时钟周期为单位。仅当 DCFG.Ena32KHzSusp 置 1 时，此位有效	6'd2	R_W
25:24	周期性调度间隔 (PerSchIntvl) 仅在 Scatter/Gather DMA 模式下有效。 用来指定 DMA 引擎给周期性 IN 端点获取数据分配的时间。基于周期性 IN 端点的数目，其值必须设定为 75% microFrame。 <ul style="list-style-type: none"> <li>● 当任意周期性端节点活动时，DMA 引擎给周期性 IN 端点获取数据分配一定的时间</li> <li>● 当无周期性端节点活动时，忽略此域</li> <li>● 在指定时间后，DMA 引擎开始给非周期性端节点获取数据。</li> <li>● 2'b00: 25% microFrame</li> <li>● 2'b01: 50% microFrame</li> <li>● 2'b10: 75% microFrame</li> <li>● 2'b11: 保留</li> </ul>	2'b00	R_W
23	在 device 模式下使能 Scatter/Gather DMA 模式 (DescDMA) <ul style="list-style-type: none"> <li>● GAHBCFG.DMAEn=0, DCFG.DescDMA=0 =&gt; 从模式</li> <li>● GAHBCFG.DMAEn=0, DCFG.DescDMA=1 =&gt; 非法</li> <li>● GAHBCFG.DMAEn=1, DCFG.DescDMA=0 =&gt; buffer DMA 模式</li> <li>● GAHBCFG.DMAEn=0, DCFG.DescDMA=0 =&gt; Scatter/Gather DMA 模式</li> </ul>	1'b0	R_W
22:14	保留		RO
13	是否使能 device OUT NAK。 仅对描述符 DMA 模式有效。 <ul style="list-style-type: none"> <li>● 1'b0: 在批量 OUT 传输结束时, OTG 控制器不送出 NAK 握手信号</li> <li>● 1'b1: 在批量 OUT 传输结束时, OTG 控制器送出 NAK 握手信号</li> </ul>	1'b0	R_W
12:11	周期帧的间隔 (PerFrInt) 指示在一个 microFrame 内何时用 EOP 中断来通知软件。此域可用来决定当前 microFrame 内是否所有的实时传输已经完成。 <ul style="list-style-type: none"> <li>● 2'b00: 80%</li> <li>● 2'b01: 85%</li> <li>● 2'b10: 90%</li> <li>● 2'b11: 95%</li> </ul>	2'h0	R_W

10:4	Device 地址(DevAddr)	7'h0	R_W
3	保留	1'b0	RO
2	非 0 长度的 Status OUT 握手(NZStsOUTHShk) 当控制传输的 Status 阶段的 OUT 事务中，OTG 接收到一个非 0 的数据包，此域用来选着 OTG 控制送出的不同的握手信号。 <ul style="list-style-type: none"> <li>● 1'b1: STALL</li> <li>● 1'b0: 送出接收到的 OUT 数据包，并基于 device endpoint 控制寄存器中 NAK 与 STALL 位，送出握手信号</li> </ul>	1'b0	R_W
1:0	设备速度 (DevSpd) <ul style="list-style-type: none"> <li>● 2'b00: HS(USB2.0 PHY)</li> <li>● 2'b01: FS(USB2.0 PHY)</li> <li>● 2'b10: LS(USB1.1 PHY)</li> <li>● 2'b11: FS(USB1.1 PHY)</li> </ul>	2'b0	R_W

### Device 控制寄存器(DCTL)

Offset: 804h

域	说明	复位值	访问特性
31:17	保留		RO
16	遇到 babble 发送 NAK 握手。	1'b0	R_W
15	对实时端节点忽略帧号 (IgnrFrmNum) <ul style="list-style-type: none"> <li>● GAHBCFG.DMAEn=0,此域无效</li> <li>● GAHBCFG.DMAEn=1, DCFG.DescDMA = 0: 当 threshold 模式有效时，此位无效。否则，软件用此位来使能周期性传输中断。 <ul style="list-style-type: none"> <li>- 1'b0 周期性传输中断无效。</li> <li>- 1'b1 当接收到 ISOC IN 标记时，无需清空包。OTG 控制器忽略帧号，一旦包准备好且接收到对应的标记，就将这个包发送出去。此域也用来使能周期性传输中断</li> </ul> </li> <li>● GAHBCFG.DMAEn=1, DCFG.DescDMA = 1: 对高速的传输无效</li> </ul>	1'b0	R_W
14:13	全局 Multi Count(GMC) 仅在 Scatter/Gater DMA 模式下有效。用来指示在转移到下一个结束点之前，接收或发送的包个数。仅对非周期性端节点有效。 <ul style="list-style-type: none"> <li>● 2'b00: 非法;</li> <li>● 2'b01: 1 个包;</li> <li>● 2'b00: 2 个包;</li> <li>● 2'b00: 3 个包;</li> </ul>	2'h0	R_W

	当 DCFG.DescDMA =1 时，此域自动改变成 2'h1		
12	保留域		RO
11	Powr-on 编程完成(PWROnPrgDone) 从 power_down 模式恢复后，寄存器编程完成时，此位置 1.	1'b0	R_W
10	清空全局 OUT NAK(CGOUTNak) 对此位的写，将清空全局 OUT NAK	1'b0	WO
9	设置全局 OUT NAK(SGOUTNak) 对此位的写，将设置全局 OUT NAK	1'b0	WO
8	清空全局的非周期性 IN NAK (CGNPIInNak) 此位的写，将清空全局非周期性 IN NAK	1'b0	WO
7	设置全局的非周期性 IN NAK (SGNPIInNak) 此位的写，将设置全局非周期性 IN NAK	1'b0	WO
6: 4	测试控制(TstCtl) <ul style="list-style-type: none"> <li>● 3'b000: 无效 Test 模式</li> <li>● 3'b001: Test_J 模式</li> <li>● 3'b010: Test_K 模式</li> <li>● 3'b011: Test_SE0_NAK 模式</li> <li>● 3'b100: Test_Packet 模式</li> <li>● 3'b101: Test_Force_Enable 模式</li> <li>● 其他: 保留</li> </ul>	3'b0	R_W
3	全局 OUT NAK 状态(GOUTNakSts) <ul style="list-style-type: none"> <li>● 1'b0: 基于 FIFO 的状态以及 NAK, STALL 位的设置情况来发送相应的握手</li> <li>● 1'b1: 不管 RxFIFO 是否有空项，都无数据写入 RxFIFO. 除 SETUP 事务以外，对所有的包都发送 NAK 握手。终止所有的实时 OUT 包。</li> </ul>	1'b0	RO
2	全局非周期性 IN NAK 状态位 1'b0: 基于 TxFIFO 的有效数据发送相应的握手; 1'b1: 对于所有的非周期性 IN 端节点，发送 NAK 握手信号，不管 TxFIFO 是否有有效数据	1'b0	RO
1	软断开 (SftDisCon) 一旦此位置 1，host 将看不到 device 的连接，device 也讲接收不到 USB 上的信号。 <ul style="list-style-type: none"> <li>● 1'b0: 正常操作</li> <li>● 1'b1: OTG 控制器将 utmi+接口上的 phy_opmode_o 信号置为 2'b01,这将给 USB host 产生一个断开事件</li> </ul>	1'b0	R_W

0	远程唤醒信号(RmtWkUpSig) 当次位置 1, OTG 控制器将发送一个远程唤醒信号去唤醒 USB host.	1'b0	R_W
---	---	------	-----

### Device 状态寄存器(DSTS)

Offset: 808h

域	说明	复位值	访问特性
31: 22	保留		RO
21:8	接收到的 SOF 的帧号(SOFN)	14'h0	RO
7: 4	保留		RO
3	严重错 (ErrticErr) OTG 控制器通过此域来报告发生了严重错误(由于 PHY 出错, phy_rxvalid_i/phy_rxvldh_i 或 phy_rxactive_i 置位超过 2ms), 一旦发生严重错误, OTG 控制器进入挂起模式, GINTSTS.ErlySusp 位置 1.且软件仅能通过软断开来恢复		RO
2:1	枚举速度(EnumSpd) <ul style="list-style-type: none"> <li>● 2'b00: HS(USB2.0 PHY)</li> <li>● 2'b01: FS(USB2.0 PHY)</li> <li>● 2'b10: LS(USB1.1 PHY)</li> <li>● 2'b11: FS(USB1.1 PHY)</li> </ul>		RO
0	挂起状态(SuspSts) Device 模式下, 只要在 USB 总线上发现挂起状态, 则置 1. 当 utmi_linestate 信号在一定事件内无活动, 则 OTG 控制器进入挂起状态。一旦以下条件发生, 则 OTG 控制器退出挂起状态: <ul style="list-style-type: none"> <li>● 当 utmi_linestate 信号有活动</li> <li>● 当软件往 DCTL.RmtWkUpSig 位写 1 时。</li> </ul>	1'b0	RO

### Device IN 端点共用的中断屏蔽寄存器(DIEPMSK)

◆ Offset: 810h

域	说明	复位值	访问特性
31: 14	保留		RO
13	NAK 中断屏蔽位 (NAKMsk)	1'h0	R_W
12:10	保留		RO
9	BNA 中断品屏蔽位 (BNAIntrMsk) 仅当描述符 DMA 模式时有效	1'h0	R_W
8	FIFO 下溢中断屏蔽位屏蔽 (TxFifoUndrnMSK)	1'b0	R_W
7	保留	1'b0	RO

6	IN 端节点 NAK 有效中断屏蔽位 (INEPNakEffMsk)	1'b0	R_W
5	接收到 EP 不匹配的 IN 标记中断屏蔽位 (INTknEPMisMsk)	1'b0	R_W
4	当 TxFIFO 空时接收到 IN 标记中断屏蔽位 (INTknTXFEmpMsk)	1'b0	R_W
3	超时中断屏蔽位 (TimeOUTMsk) 仅对非实时端节点有效	1'b0	R_W
2	AHB 错误中断屏蔽位 (AHBErrMsk)	1'b0	R_W
1	端节点失效中断屏蔽位 (EPDisbldMsk)	1'b0	R_W
0	传输完成中断屏蔽位 (XferCompIMsk)	1'b0	R_W

**device OUT 端点共用的中断屏蔽寄存器(DOEPMSK)**

◆ Offset: 814h

域	说明	复位值	访问特性
31: 15	保留		RO
14	NYET 中断屏蔽位 (NYETMsk)	1'b0	R_W
12	空泡中断屏蔽位 (BbleErrMsk)	1'b0	R_W
11:10	保留		RO
9	BNA 中断屏蔽位 (BnaOutIntrMsk)	1'b0	R_W
8	OUT 包错误中断屏蔽位 (OutPktErrMsk)	1'b0	R_W
7	保留		RO
6	接收到背对背的 SETUP 包中断屏蔽位 (Back2BackSETup)	1'b0	R_W
5	保留		RO
4	当端节点失效时接收到 OUT 标记中断屏蔽位 (OUTTknEPdisMsk)	1'b0	R_W
3	SETUP 阶段完成中断屏蔽位 (SetUPMsk)	1'b0	R_W

2	AHB 错误中断屏蔽位 (AHBErrMsk)	1'b0	R_W
1	端节点失效中断屏蔽位 (EPDisbldMsk)	1'b0	R_W
0	传输完成中断屏蔽位 (XferCompIMsk)	1'b0	R_W

**Device 所有端点共用的中断寄存器(DAINT)**

◆ Offset: 818h

域	说明	复位值	访问特性
31: 16	OUT 端节点中断位 (OutEPInt) 一个 OUT 端节点对应一位。 位 16 对应端节点 0	16'h0	RO
15:0	IN 端节点中断位 (InEPInt) 一个 IN 端节点对应一位。 位 0 对应端节点 0	16'h0	RO

**Device 所有端点共用的中断屏蔽寄存器(DAINTMSK)**

◆ Offset: 81Ch

域	说明	复位值	访问特性
31: 16	OUT 端节点中断屏蔽位 (OutEPIntMsk) 一个 OUT 端节点对应一位。 位 16 对应端节点 0	16'h0	R_W
15:0	IN 端节点中断屏蔽位 (InEPIntMsk) 一个 IN 端节点对应一位。 位 0 对应端节点 0	16'h0	R_W

**Device VBUS 放电寄存器(DVBUSDIS)**

◆ Offset: 828h

域	说明	复位值	访问特性
31:16	保留		RO
15:0	Device Vbus 放电时间(DVBUSDis) SRP 期间, 在 Vbus 脉冲后 Vbus 放电的时间。以 PHY 时钟为单位。	30MHZ: 16'h0B8F  60MHZ: 16'h17D7	R_W

**Device VBUS 脉冲寄存器(DVBUSPULSE)**

◆ Offset: 82Ch

域	说明	复位值	访问特性
31:12	保留		RO
11:0	Device Vbus 脉冲时间(DVBUSPulse) SRP 期间, 在 Vbus 脉冲的时间。以 PHY 时钟为单 位。	30MHZ: 12'h2c6  60MHZ: 12'h5b8	R_W

**Device 阈值控制寄存器(DTHRCTL)**

◆ Offset: 830h

域	说明	复位值	访问特性
31:28	保留		RO
27	仲裁器 parking 使能	1'b1	R_W
26	保留		RO
25:17	接收阈值长度 (RxThrLen)	9'h8	R_W
16	接收阈值使能 (RxThrEn)	1'b0	R_W
15:13	保留		RO
12:11	AHB 阈值率 (AHBThrRatio)	2'b0	R_W
10:2	发送阈值长度 (TxThrLen)	9'h8	R_W
1	实时 IN 端节点阈值使能 (ISOThrEn)	1'b0	R_W
0	非实时 IN 端节点阈值使能 (NonISOThrEn)	1'b0	R_W

**Device IN 端节 FIFO 空中断屏蔽寄存器(DIEPEMPSK)**

◆ Offset: 834h

域	说明	复位值	访问特性
31:16	保留		RO
15:0	IN 端节点 Tx FIFO 空中断屏蔽位 (InEpTxfEmpMsk) 作为 DIEPINTn 的中断屏蔽位, 一位对应一个端节点。 Bit0 对应端节点 0, 以此类推	16'b0	R_W

**Device 控制 IN 端节 0 控制寄存器(DIEPCTL0)**

◆ Offset: 900h

域	说明	复位值	访问特性
31	端节点使能 (EPEna) ● 对于 Scatter/Gather DMA 模式, 对于 IN 端节点 来讲, 此域置 1 意味着描述符以及数据缓冲都已	1'b0	R_WS_SC

	<p>经准备好。</p> <ul style="list-style-type: none"> <li>● 对于非 Scatter/Gather DMA 模式，此域置 1 意味着数据缓冲已经准备好。</li> </ul> <p>当 OTG 设置以下中断时，将此位清 0:</p> <ul style="list-style-type: none"> <li>● 端节点失效</li> <li>● 传输完成</li> </ul>		
30	<p>端节点失效 (EPDis)</p> <p>软件将此位置 1，即使此端节点上的传输未完成，也将停止在此端节点上发送数据。软件必须等待端节点失效中断发生后，才能认为此端节点已经失效。当 OTG 控制器设置端节点失效中断时，将此位清 0。软件仅当在 EPEna 位为 1 时才能置此位为 1。</p> <p>此域仅在 DMA 模式下有效。</p>	1'b0	R_WS_SC
29:28	保留		RO
27	<p>设置 NAK(SNAK)</p> <p>置 1,设置此端节点对应的 NAK 位为 1.</p>	1'b0	WO
26	<p>清空 NAK (CNAK)</p> <p>置 1,设置此端节点对应的 NAK 位为 0.</p>	1'b0	WO
25:22	<p>TxFIFO 号 (TxFNum)</p> <p>设置为 IN 端点 0 对应的 FIFO 号</p>	4'h0	R_W
21	<p>STALL 握手 (Stall)</p> <p>当收到 SETUP 包时，软件置 1，OTG 控制器清 0。</p> <p>如果同时 NAK 位，全局非周期性 IN NAK，全局 OUT NAK 位都置 1，那么 STALL 位优先级最高。</p>	1'b0	R_WS_SC
20	保留		RO
19:18	<p>端节点类型 (EPType)</p> <p>对于控制节点为 2'b00</p>	2'h0	RO
17	<p>NAK 状态 (NAKSts)</p> <ul style="list-style-type: none"> <li>● 1'b0: 基于 FIFO 的状态，OTG 控制器发送非 Non-NAK 握手</li> <li>● 1'b1: OTG 控制器发送 NAK 握手</li> </ul>	1'b0	RO
16	保留		RO
15	<p>USB 活动端节点(USBActEP)</p> <p>一直为 1，表示控制端节点 0 总是可用</p>	1'b1	RO
14:2	保留		RO

1:0	最大的包大小 (MPS) <ul style="list-style-type: none"> <li>● 2'b00: 64B</li> <li>● 2'b01: 32B</li> <li>● 2'b10: 16B</li> <li>● 2'b11: 8B</li> </ul>	2'h0	R_W
-----	---	------	-----

**Device 控制 OUT 端节 0 控制寄存器(DOEPCTL0)**

◆ Offset: B00h

域	说明	复位值	访问特性
31	端节点使能 (EPEna) <ul style="list-style-type: none"> <li>● 对于 Scatter/Gather DMA 模式, 对于 OUT 端节点来讲, 此域置 1 意味着描述符以及数据缓冲都已经准备好接收数据。</li> <li>● 对于非 Scatter/Gather DMA 模式, 此域置 1 意味着数据缓冲已经准备好。</li> </ul> 当 OTG 设置以下中断时, 将此位清 0: <ul style="list-style-type: none"> <li>● SETUP 时相完成</li> <li>● 端节点失效</li> <li>● 传输完成</li> </ul> 在 DMA 模式下, 若 OTG 控制器传输 SETUP 数据包时, 此位必须置 1.	1'b0	R_WS_SC
30	端节点失效 (EPDis) 软件不能使 OUT 控制节点 0 失效	1'b0	RO
29:28	保留		RO
27	设置 NAK(SNAK) 置 1,设置此端节点对应的 NAK 位为 1.	1'b0	WO
26	清空 NAK (CNAK) 置 1,清空此端节点对应的 NAK 位为 1.	1'b0	WO
25:22	保留		RO
21	STALL 握手 当收到 SETUP 包时, 软件置 1, OTG 控制器清 0. 如果同时 NAK 位, 全局非周期性 NAK, 全局 OUT NAK 位都置 1, 那么 STALL 位优先级最高。但是不管此位是否置 1, OTG 控制器对 SETUP 数据包的握手信号总是 ACK	1'b0	R_WS_SC
20	Snoop 模式(Snp) 在 snoop 模式下, 在传输数据给内存时, OTG 控制器不检查 OUT 数据包的正确性	1'b0	R_W
19:18	端节点类型 (EPType) 对于控制节点为 2'b00	2'h0	RO

17	NAK 状态 (NAKSts) <ul style="list-style-type: none"> <li>● 1'b0: 基于 FIFO 的状态, OTG 控制器发送非 Non-NAK 握手</li> <li>● 1'b1: OTG 控制器发送 NAK 握手</li> </ul>	1'b0	RO
16	保留		RO
15	USB 活动端节点(USBActEP) 一直为 1, 表示控制端节点 0 总是可用	1'b1	RO
14:2	保留		RO
1:0	最大的包大小 (MPS) <ul style="list-style-type: none"> <li>● 2'b00: 64B</li> <li>● 2'b01: 32B</li> <li>● 2'b10: 16B</li> <li>● 2'b11: 8B</li> </ul>	2'h0	R_W

### Device 端节-n 控制寄存器(DIEPCTLn/DOEPCTLn)

- ◆ Endpoint\_num: 1 <= n <= 6
- ◆ Offset for IN: 900h + (Endpoint\_num\*20h)
- ◆ Offset for OUT: B00h + (Endpoint\_num\*20h)

域	说明	复位值	访问特性
31	端节点使能(EPEna) 应用于 IN/OUT 端节点。 <ul style="list-style-type: none"> <li>● 对于 Scatter/Gather DMA 模式, 对于 IN 端节点来讲, 此域置 1 意味着描述符以及数据缓冲都已经准备好; 对于 OUT 端节点来讲, 此域置 1 意味着描述符以及数据缓冲都已经准备好接收数据。</li> <li>● 对于非 Scatter/Gather DMA 模式, 对于 IN 端节点来讲, 此域置 1 意味着数据缓冲已经准备好发送数据; 对于 IN 端节点来讲, 此域置 1 意味着数据缓冲已经准备好接收数据;</li> </ul> 当 OTG 设置以下中断时, 将此位清 0: <ul style="list-style-type: none"> <li>● SETUP 时相完成</li> <li>● 端节点失效</li> <li>● 传输完成</li> </ul>	1'b0	R_WS_SC
30	端节点失效 (EPDis) 应用于 IN/OUT 端节点 软件将此位置 1, 即使此端节点上的传输未完成, 也将停止在此端节点上发送数据。软件必须等待端节点失效中断发生后, 才能认为此端节点已经失效。当 OTG 控制器设置端节点失效中断时, 将此位清 0。软件仅当在 EPEna 位为 1 时才	1'b0	R_WS_SC

	能置此位为 1.		
29	<p>设置 DATA1 PID (SetD1PID)</p> <p>仅对中断/批量 IN 以及 OUT 端节点有效。</p> <p>对此位写 1, 则置 DPID 位 DATA1。</p> <p>对描述符 DMA 与非描述符 DMA 模式都有效</p>	1'b0	WO
28	<p>设置 DATA0 PID (SetD0PID)</p> <p>仅对中断/批量 IN 以及 OUT 端节点有效。</p> <p>对此位写 1, 则置 DPID 位 DATA0。</p> <p>对描述符 DMA 与非描述符 DMA 模式都有效。</p>	1'b0	WO
27	<p>设置 NAK(SNAK)</p> <p>应用于 IN/OUT 端节点</p> <p>置 1,设置此端节点对应的 NAK 位为 1.</p>	1'b0	WO
26	<p>清空 NAK (CNAK)</p> <p>应用于 IN/OUT 端节点</p> <p>置 1,清空此端节点对应的 NAK 位为 1.</p>	1'b0	WO
25:22	<p>TxFIFO 号 (TxFNum)</p> <p>仅对 IN 端节点有效;</p> <p>设置为 IN 端对应的 FIFO 号</p>	4'h0	R_W
21	<p>STALL 握手</p> <p>仅对非控制, 非实时的 IN/OUT 端节点有效。</p> <p>通过设置此位, 来 stall 所有从 host 来的标记。如果同时 NAK 位, 全局非周期性 NAK, 全局 OUT NAK 位都置 1, 那么 STALL 位优先级最高。但是不管此位是否置 1, OTG 控制器对 SETUP 数据包的握手信号总是 ACK</p> <p>仅对控制端节点有效。</p> <p>当收到 SETUP 包时, 软件置 1, OTG 控制器清 0.</p> <p>如果同时 NAK 位, 全局非周期性 NAK, 全局 OUT NAK 位都置 1, 那么 STALL 位优先级最高。但是不管此位是否置 1, OTG 控制器对 SETUP 数据包的握手信号总是 ACK</p>	1'b0	R_W
20	<p>Snoop 模式(Snp)</p> <p>仅对 OUT 端节点有效</p> <p>在 snoop 模式下, 在传输数据给内存时, OTG 控制器不检查 OUT 数据包的正确性</p>	1'b0	R_W
19:18	<p>端节点类型 (EPTyep)</p> <ul style="list-style-type: none"> <li>● 2'b00: 控制</li> <li>● 2'b01: 实时</li> <li>● 2'b10: 批量</li> <li>● 2'b11: 中断</li> </ul>	2'h0	R_W
17	NAK 状态(NAKSts)	1'b0	RO

	<p>应用于所有的 IN/OUT 端节点</p> <ul style="list-style-type: none"> <li>● 1'b0: 基于 FIFO 的状态, OTG 控制器发送非 Non-NAK 握手</li> <li>● 1'b1: OTG 控制器发送 NAK 握手</li> </ul>		
16	<p>端节点数据 PID (DPID)</p> <p>应用于中断/批量 IN 以及 OUT 端节点。</p> <p>包括在此端节点接收或发送的数据包的 PID。当此端节点活动后, 软件必须设置此端节点接收或发送的第一个包的 PID。使用此寄存器的 SetD0PID 以及 SetD1PID 域来设置 DATA0 或 DATA1 PID</p> <ul style="list-style-type: none"> <li>● 1'b0: DATA0</li> <li>● 1'b1: DATA1</li> </ul> <p>奇偶帧(EO_FrNum)</p> <p>在非描述符 DMA 模式下, 应用于实时的 IN/OUT 端节点。指定为此端节点发送或接收实时数据所用地帧号。通过此寄存器的 SetEvnFr 与 SetOddFr 域来设置。</p> <ul style="list-style-type: none"> <li>● 1'b0: 偶帧</li> <li>● 1'b1: 奇帧</li> </ul> <p>在描述符 DMA 模式下, 此域位保留域</p>	1'b0	RO
15	<p>USB 活动端节点</p> <p>应用于 IN 或 OUT 端节点。</p> <p>指示在当前配置与接口下, 端节点是否活动。</p>	1'b0	R_W_Sc
14:11	保留		RO
10:0	<p>最大包大小 (MPS)</p> <p>应用于 IN 或 OUT 端节点。设置最大的包大小, 以字节为单位</p>	11'b0	R_W

### Device 端节-n 中断寄存器(DIEPCTLn /DOEPCTLn)

- ◆ Endpoint\_num: 1 ≤ n ≤ 6
- ◆ Offset for IN: 908h + (Endpoint\_num\*20h)
- ◆ Offset for OUT: B08h + (Endpoint\_num\*20h)

域	说明	复位值	访问特性
31:15	保留		RO
14	NYET 中断 (NYETIntrpt)	1'b0	R_SS_WC
13	NAK 中断 (NAKIntrpt)	1'b0	R_SS_WC
12	空泡错中断 (BbleErrIntrpt)	1'b0	R_SS_WC
11	包 drop 状态 (PktDrpSts)	1'b0	R_SS_WC
10	保留		RO
9	BNA (缓冲不可用) 中断(BNAIntr)	1'b0	R_SS_WC

8	FIFO 下溢 (TxFifoUndrn)	1'b0	R_SS_WC
7	TxFIFO 空 (TxFEmp)	1'b0	R_SS_WC
6	IN 端节点 NAK 有效 (INEPNakEff)	1'b0	R_SS_WC
5	接收到 EP 不匹配的 IN 标记中断 (INTknEPMis)	1'b0	R_SS_WC
4	当 TxFIFO 空时接收到 IN 标记中断 (INTknTXFEmp)	1'b0	R_SS_WC
3	超时中断 (TImeOUT) 仅对非实时端节点有效	1'b0	R_SS_WC
2	AHB 错误中断 (AHBErr)	1'b0	R_SS_WC
1	端节点失效中断 (EPDisbld)	1'b0	R_SS_WC
0	传输完成中断 (XferCompl)	1'b0	R_SS_WC

### Device 端节 0 传输大小寄存器(DIEPTSIZ0/DOEPTSIZ0)

◆ Offset for IN: 910h

◆ Offset for OUT: B10h

对于 IN 端节点:

域	说明	复位值	访问特性
31:15	保留		RO
20:19	包数量(PktCnt) 指示在此传输大小数量下的包数目。 每当从 TxFIFO 中读取一个包, 此域减 1。	2'b0	R_W
18:7	保留域	7'h0	R_W
6:0	传输大小 指示端节点 0 的传输大小 (以字节为单位), 当超出这个值, 则将引发中断。可被设置为此端节点的 MPS。每次一个包写入 TxFIFO, 此域减 1	7'h0	R_W

对于 OUT 端节点:

域	说明	复位值	访问特性
31	保留		RO

30:29	SETUP 包数量(SUPPktCnt) 指示此端节点能接受到的背对背 SETUP 包的数量 <ul style="list-style-type: none"> <li>● 2'b01: 1 个包</li> <li>● 2'b10: 2 个包</li> <li>● 2'b11: 3 个包</li> </ul>	2'b0	R_W
28:20	保留域	7'h0	R_W
19	包数量(PktCnt) 每当一个包写入 RxFIFO, 此域减到 0.	1'b0	R_W
18:7	保留		RO
6:0	传输大小 (XferSize) 指示端节点 0 的传输大小 (以字节为单位), 当超出这个值, 则将引发中断。可被设置为此端节点的 MPS。每次从 RxFIFO 中读取一个包, 此域减 1	7'h0	R_W

**Device 端节-n 传输大小寄存器 (DIEPSIZn /DOEPSIZn)**

- ◆ Endpoint\_num: 1 <= n <= 6
- ◆ Offset for IN: 910h + (Endpoint\_num\*20h)
- ◆ Offset for OUT: B10h + (Endpoint\_num\*20h)

域	说明	复位值	访问特性
31:29	保留		RO
30:29	MC (MultiCount) 仅对 IN 端节点有效。 对周期性 IN 端节点, 此域指定在一个 microFrame 时间内传输的包个数。OTG 控制器使用此域为实时 IN 端节点计算数据 PID。 <ul style="list-style-type: none"> <li>● 2'b01: 1 个包;</li> <li>● 2'b10: 2 个包;</li> <li>● 2'b11: 3 个包</li> </ul> 对于非周期性 IN 端节点, 指定 OTG 控制器必须为此端节点获取的包数, 在转换到下由 DIEPCTLn.Nextep 指定的下一个端节点时。  RxPID (接收到的数据 PID) 仅对实时 OUT 端节点有效 此端节点接收到的最后一个包的数据 PID <ul style="list-style-type: none"> <li>● 2'b00: DATA0</li> <li>● 2'b01: DATA2</li> <li>● 2'b10: DATA1;</li> <li>● 2'b11: MDATA;</li> </ul>	2'b0	R_W

	<p>SETUP 包数(SUPCnt)</p> <p>仅对控制 OUT 端节点有效</p> <p>指示此端节点能接受到的背对背 SETUP 包的数量</p> <ul style="list-style-type: none"> <li>● 2'b01: 1 个包</li> <li>● 2'b10: 2 个包</li> <li>● 2'b11: 3 个包</li> </ul>		
28:19	<p>包数量(PktCnt)</p> <p>指示在此传输大小数量下的包数目 (PktCnt = XferSize/MPS)</p> <ul style="list-style-type: none"> <li>● IN 端节点: 每当从 TxFIFO 中读取一个包, 此域减 1。</li> <li>● OUT 端节点: 每当一个包写入 RxFIFO, 此域减到 1。</li> </ul>	10'h0	R_W
18:0	<p>传输大小 (XferSize)</p> <p>指示端节点的传输大小 (以字节为单位)。</p> <ul style="list-style-type: none"> <li>● 每次从 RxFIFO 中读取一个包, 此域减 1</li> <li>● 每次一个包写入 TxFIFO, 此域减 1</li> </ul>	19'b0	R_W

#### Device 端节-n DMA 地址寄存器 (DIEPDMA<sub>n</sub>/DOEPDMA<sub>n</sub>)

- ◆ Endpoint\_num: 1 ≤ n ≤ 6
- ◆ Offset for IN: 914h + (Endpoint\_num\*20h)
- ◆ Offset for OUT: B14h + (Endpoint\_num\*20h)

域	说明	复位值	访问特性
31:0	<p>DMA 地址(DMAAddr)</p> <p>保存分配给此 endpoint 的外部存储的起始地址</p> <ul style="list-style-type: none"> <li>● 非 Scatter/Gather DMA 模式下, 为起始地址</li> <li>● Scatter/Gather DMA 模式下, 为描述符表的基准地址</li> </ul>	X	R_W

#### Device 端节-n DMA Buffer 地址寄存器 (DIEPDMAB<sub>n</sub>/DOEPDMAB<sub>n</sub>)

- ◆ Endpoint\_num: 1 ≤ n ≤ 6
- ◆ Offset for IN: 91Ch + (Endpoint\_num\*20h)
- ◆ Offset for OUT: B1Ch + (Endpoint\_num\*20h)

域	说明	复位值	访问特性
31:0	<p>DMA buffer 地址(DMABufferAddr)</p> <p>保存当前 buffer 的地址。当对应此端节点的数据传输在进行时, 此地址将实时更新。</p> <p>仅在 Scatter/Gather DMA 模式下有效。</p>	X	RO

#### Device IN 端节点-nTxFIFO 状态寄存器 (DTXSTSn)

- ◆ Endpoint\_num: 1 ≤ n ≤ 6
- ◆ Offset 918h + (Endpoint\_num\*20h)

域	说明	复位值	访问特性
---	----	-----	------

31:16	保留		RO
15: 0	IN 端节点 Tx FIFO 可用空间(INEPTxFSpCavail) 以 32 字节为单位。 <ul style="list-style-type: none"> <li>● 16'h0: 端节点 Tx FIFO 满</li> <li>● 16'h1: 一个字可用</li> <li>...</li> <li>● 16'hn: n 个字可用</li> <li>● 其他: 保留</li> </ul>		RO

### 13.3.5 功耗与门控时钟寄存器

#### 时钟控制寄存器功耗与门控 (PCGCCTL)

◆ Offset E00h

域	说明	复位值	访问特性
31:9	保留		RO
8	挂起后复位 (ResetAfterSusp) 如果 host 需要在挂起后发出复位, 那么在 clamp 移除之前, 软件需置 1 此位。如果此位未置 1, 那么 host 在挂起之后发出复位信号	1'b1	R_W
7	深度睡眠 (L1 Suspended) 当处于 L1 状态时, PHY 进入深度睡眠。	1'b0	RO
6	PHY 睡眠 (PhySleep) PHY 进入睡眠状态。	1'b0	RO
5	使能睡眠门控 (Enbl_L1Gating) 如果 OTG 控制器不能置位 utmi_l1_suspend_n 时, 设置此位后, 在睡眠状态下能对内部时钟做门控	1'b0	R_W
4	保留		RO
3	复位低功耗模块 (RstPdownModule) 当断电时, 软件置 1。在上电以及 PHY 时钟起来后, 软件清 0	1'b0	R_W
2	供电隔离 (PwrClmp) 在关电之前, 软件将此位置 1 用来隔离 power-on 与 power-off 模块。在上电之前, 软件清 0	1'b0	R_W
1	门控 Hclk(GateHclk) 置 1, 则当 USB 处于挂起状态时, 对 Hclk 使用门控时钟。当 USB 恢复或新的对话开始时, 软件清 0	1'b0	R_W
0	停止 Pclk(StopPclk) 置 1, 则当 USB 处于挂起状态时, 停止 PHY 时钟。当 USB 恢复或新的对话开始时, 软件清 0	1'b0	R_W



# 14 通用 DMA

龙芯 1C 处理器包含一个 DMA 模块，共有三路 DMA，用于块数据或者流数据的操作，以提高处理器工作效率。1C 芯片一共有五个模块(SDIO、ADC、AC97、I2S、NAND)会用到通用 DMA，通过软件配置共用三路 DMA。

## 14.1 DMA 控制器结构描述

DMA 传送数据的过程由三个阶段组成：

1. 传送前的预处理：由 CPU 完成以下步骤：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 控制器限定为以字为单位的数据搬运。根据 DMA 的定义，设计了下个描述符地址寄存器、源地址寄存器、目的地址寄存器、传送字数计数器、传送步长间隔、传送循环次数、DMA 控制逻辑等必备寄存器。DMA 的缓存大小为 128Byte (32x4Byte)，以字为单位读写。

CPU 通过配置 DMA 寄存器，将来自于 SDRAM 或设备的数据保存在缓存中，将缓存中的数据写入要对应的内存或设备中去，最后发送 DMA 传输结束信号。在 DMA 传输过程中，CPU 可以随时监听 DMA 的工作状态。

## 14.2 DMA 寄存器

### 14.2.1 ORDER\_ADDR\_IN

中文名： 该寄存器广播到三路 DMA，被选中的 DMA 根据寄存器的配置开始工作

寄存器位宽： [31: 0]

地址： 0x1fd0\_1160

复位值： 0x00000000

位域	位域名称	位宽	访问	描述
31:6	Ask_addr	26	R/W	被选中 DMA 第一个描述符地址的高 26 位，低 6 位为 0；相当于 26 为的 Ask_addr 左移 6 位。
5	保留	1		
4	dma_stop	1	R/W	用户请求停止 DMA 操作；完成当前数据读写操作后，停止操作
3	dma_start	1	R/W	可以开始读描述符链的第一个 DMA 描述符；当第一个描述符相关的寄存器读回后，该位清零
2	Ask_valid	1	R/W	用户请求将当前 DMA 操作的相关信息写回到指定的内存地址；当用户写回 DMA 操作相关信息后，该位清零。
1:0	Dev_num	2	R	2'b00 nand flash

				2'b01 AC97 read device 2'b10 AC97 write device
--	--	--	--	---

说明:

第一个描述符的地址在 ORDER\_ADDR\_IN 寄存器中, 该寄存器由 CPU 来配置, 也就是 Ask\_addr 左移 6 位后组成了所有描述符寄存器的基地址。

每次 DMA 操作, DMA\_ORDER\_ADDR 寄存器存放的下个描述符的地址和有效位。

如果 ask\_valid=1, 表示 CPU 要侦听 DMA 操作, 此时要将 DMA 控制器寄存器的值写回到 ask\_addr 指向的内存中。

如果 dma\_start=1, 表示开始 DMA 操作, DMA 先从 ask\_addr 指向的内存地址读描述符, 然后根据描述符的信息开始执行 DMA 操作。

### 14.2.2 DMA\_ORDER\_ADDR

中文名: 下一个描述符地址寄存器

寄存器位宽: [31: 0]

基地址: Ask\_addr<<6

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	存储器内部下一个描述符地址寄存器
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。

### 14.2.3 DMA\_SADDR

中文名: 内存地址寄存器

寄存器位宽: [31: 0]

基地址: Ask\_addr<<6

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 操作的内存地址

说明: DMA 操作分为: 从内存中读数据, 保存在 DMA 控制器的缓存中, 由 APB 发请求来访问 DMA 缓存中的数据, 该寄存器指定了读 ddr2 的地址; 从 APB 设备读数据保存在 DMA 缓存中, 当 DMA 缓存中的字超过一定数目, 就往内存中写, 该寄存器指定了写内存的地址。

### 14.2.4 DMA\_DADDR

中文名: 设备地址寄存器

寄存器位宽: [31: 0]

基地址: Ask\_addr<<6

偏移地址: 0x8

复位值: 0x00000000

位域	位域名称	位宽	访问	描述
31		1	R/W	AC97 写使能, “1” 表示是写操作
30		1	R/W	0:mono 1: 2 stero
29:28		2	R/W	AC97 写模式, 0: 1byte, 1: 2byte, 2: 4byte
27:0	dma_daddr	32	R/W	DMA 操作的 APB 设备地址

说明：从内存中读数据，保存在 DMA 控制器的缓存中，由 APB 发请求来访问 DMA 缓存中的数据，该寄存器指定了写 APB 设备的地址；从 APB 设备读数据保存在 DMA 缓存中，当 DMA 缓存中的字超过一定数目，就往内存中写，该寄存器指定了读 APB 设备的地址。

### 14.2.5 DMA\_LENGTH

中文名： 长度寄存器  
寄存器位宽： [31: 0]  
基地址： Ask\_addr<<6  
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。

### 14.2.6 DMA\_STEP\_LENGTH

中文名： 间隔长度寄存器  
寄存器位宽： [31: 0]  
基地址： Ask\_addr<<6  
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。

### 14.2.7 DMA\_STEP\_TIMES

中文名： 循环次数寄存器  
寄存器位宽： [31: 0]  
基地址： Ask\_addr<<6  
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。

### 14.2.8 DMA\_CMD

中文名： 控制寄存器  
寄存器位宽： [31: 0]  
基地址： Ask\_addr<<6  
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
位域	位域名称	位宽	访问	描述

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

### 14.3 多个模块使用 DMA 配置说明

如果某个模块需要使用 DMA 操作，则需要发送请求到 DMA 的 req 信号。通过配置 misc\_ctrl 寄存器，将该模块的 req 信号连接到 DMA 的 req[2:0]信号上之后，才能正常使用 DMA。

其中 SDIO 的优先级最高，可以配置为 3 通道中的任意一路；当 SDIO 不使

用第 0 通道时，NAND 使用第 0 通道；当 SDIO 不使用第 1 通道时，I2S 和 AC97 复用第 1 通道；当 SDIO 不使用第 2 通道时，ADC、I2S、AC97 复用第 2 通道。ADC 优先级比 I2S 和 AC97 要高。

DMA 复用配置对应 misc\_ctrl[25:22]，其中 misc\_ctrl[22]为 adc\_dma\_en，misc\_ctrl[24:23]为 sdio\_dma\_en，misc\_ctrl[25]为 ac97\_en。

比如 SDIO 使用第 1 通道，ADC 使用第 2 通道，NAND 使用第 0 通道，则应该配置如下：

sdio\_dma\_en 为 2，adc\_dma\_en 为 1，ac97 和 I2S 都不能使用 DMA。

如果 SDIO 使用第 0 通道，ADC 使用第 2 通道，I2S 使用第 1 通道，则应该配置如下：

sdio\_dma\_en 为 1，adc\_dma\_en 为 1，ac97\_en 为 0。

# 15 SPI0 控制器

串行外围设备接口 SPI 总线技术是 Motorola 公司推出的多种微处理器、微控制器以及外围设备之间的一种全双工、同步、串行数据接口标准。

## 15.1 SPI 控制器结构

本系统集成的 SPI 控制器仅可作为主控端，所连接的是从设备。对于软件而言，SPI 控制器除了有若干 IO 寄存器外还有一段映射到 SPI Flash 的只读 memory 空间。如果将这段 memory 空间分配在 0x1fc00000，复位后不需要软件干预就可以直接访问，从而支持处理器从 SPI Flash 启动。SPI 的 IO 寄存器的基地址 0x1fe8\_0000，外部存储地址空间是 0x1f00\_0000 - 0x1f7f\_ffff 共 8MB。

其结构如图 15-1 所示，由 AXI 内部总线接口、简单的 SPI 主控制器、SPI Flash 读引擎和总线选择模块组成。根据访问的地址和类型，来自内部总线接口上的合法请求转发到 SPI 主控制器或者 SPI Flash 读引擎中(非法请求被丢弃)。

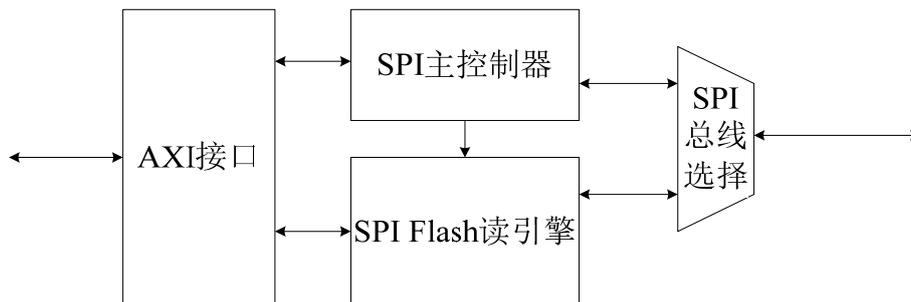


图 15-1 SPI 控制器结构

## 15.2 配置寄存器

表 15-1 SPI 配置寄存器列表

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

### 15.2.1 控制寄存器(SPCR)

表 15-2 SPI 控制寄存器(SPCR)

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

位域	名称	访问	初值	描述
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

### 15.2.2 状态寄存器(SPSR)

表 15-3 SPI 状态寄存器(SPSR)

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	wffull	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

### 15.2.3 数据寄存器(TxFIFO/RxFIFO)

表 15-4 SPI 数据寄存器(TxFIFO/RXFIFO)

位域	名称	访问	初值	描述
7:0	TxFIFO	W	-	数据发送端口
	RxFIFO	R		数据接收端口

### 15.2.4 外部寄存器(SPER)

表 15-5 SPI 外部寄存器(SPER)

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	spi 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 15-6 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

### 15.2.5 参数控制寄存器(SFC\_PARAM)

表 15-7 SPI 参数控制寄存器(SFC\_PARAM)

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。

注：通过 EJTAG 往 SPI flash 烧写时需要将 memory\_en 置为 0。

### 15.2.6 片选控制寄存器(SFC\_SOFTCS)

表 15-8 SPI 片选控制寄存器(SFC\_SOFTCS)

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

### 15.2.7 时序控制寄存器(SFC\_TIMING)

表 15-9 SPI 时序控制寄存器(SFC\_TIMING)

位域	名称	访问	初值	描述
7:3	-	-	-	保留
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

## 15.3 接口时序

### 15.3.1 SPI 主控制器接口时序

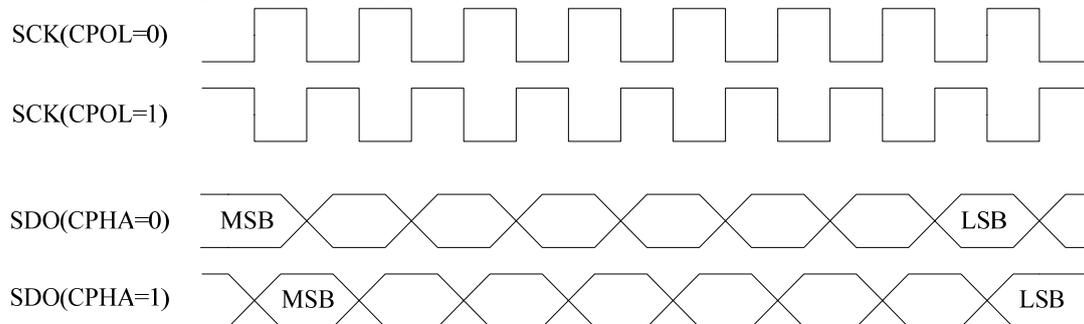


图 15-2 SPI 主控制器接口时序

### 15.3.2 SPI Flash 访问时序

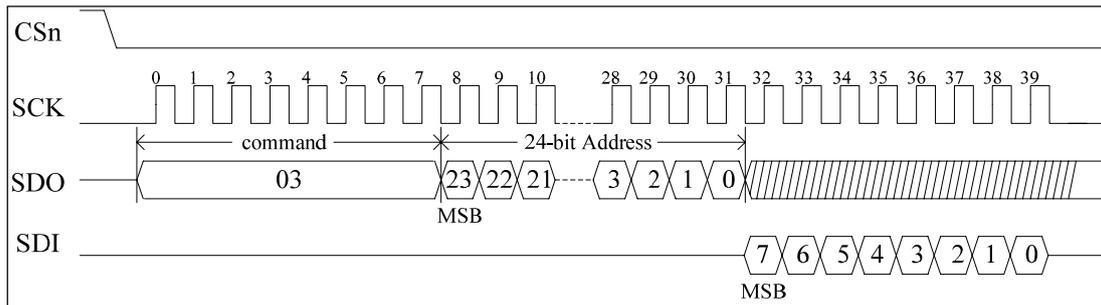


图 15-3 SPI Flash 标准读时序

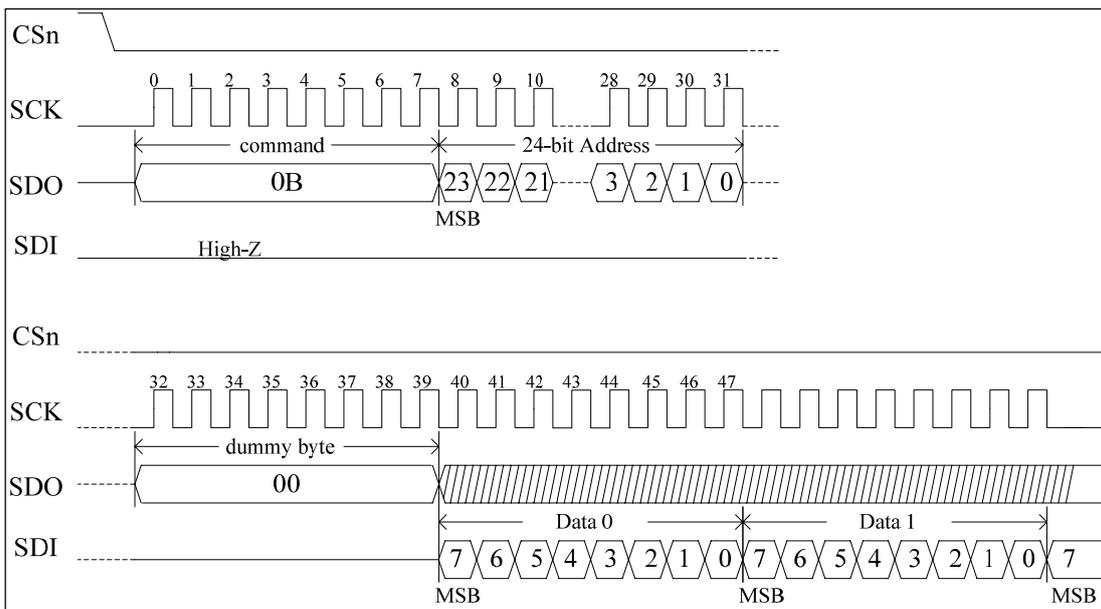


图 15-4 SPI Flash 快速读时序

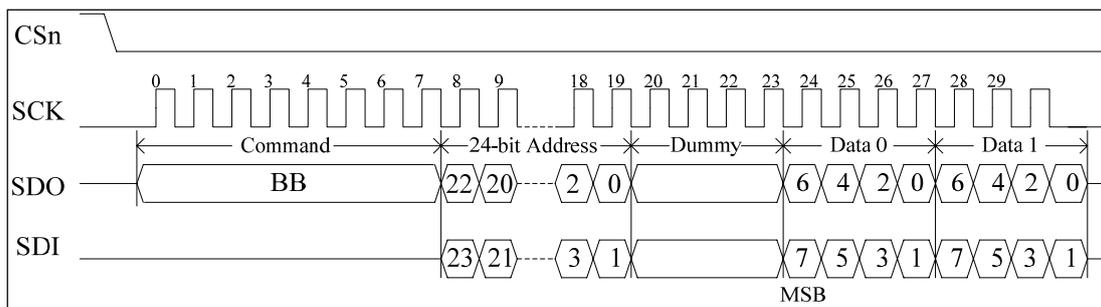


图 15-5 SPI Flash 双向 I/O 读时序

## 15.4 使用指南

### 15.4.1 SPI 主控制器的读写操作

#### 1. 模块初始化

- 停止 SPI 控制器工作，对控制寄存器 `spcr` 的 `spe` 位写 0
- 重置状态寄存器 `spsr`，对寄存器写入 `8'b1100_0000`
- 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

- 配置 SPI 时序，包括 sPCR 的 cpol、cpha 和 sper 的 mode 位。mode 为 1 时是标准 SPI 实现，为 0 时为兼容模式。
- 配置中断使能，sPCR 的 spie 位
- 启动 SPI 控制器，对控制寄存器 sPCR 的 spe 位写 1

## 2. 模块的发送/传输操作

- 往数据传输寄存器写入数据
- 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。

## 3. 中断处理

- 接收到中断申请
- 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
- 读或写数据传输寄存器
- 往状态寄存器 sPSR 的 spif 位写 1，清除控制器的中断申请

### 15.4.2 硬件 SPI Flash 读

#### 1. 初始化

- 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。
- 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。

#### 2. 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。

### 15.4.3 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。

## 16 SPI1

串行外围设备接口 SPI 总线技术是 Motorola 公司推出的多种微处理器、微控制器以及外围设备之间的一种全双工、同步、串行数据接口标准。

### 16.1 SPI 主控制器结构

SPI1 和 SPI0 的实现完全一样，系统启动地址不会映射到 SPI1 控制器，所以 SPI1 不支持系统启动。SPI1 的 IO 寄存器的基地址 0x1fec0000，SPI1 的外部存储地址空间是 0x1f80,0000 - 0x1fbf,ffff 共 4MB。所有结构和配置相关请参考第 15 章信息。

# 17 AC97

## 17.1 概述

在系统里一个 AC97 应用系统如图 19-1 所示。在一个片上系统中,与 AC97 控制器相连的有 3 部分:一是外设总线,接收来自微处理器的控制信息以及配置信息;二是 AC97 Codec,多媒体数字信号编解码器,该解码器对 PCM 信号进行调制,输出人耳接受的模拟声音或者把真实的声音转换为 PCM 信号,转换通过 D/A 转换器实现;三是 DMA 引擎,通过 DMA 的方式写或读 AC97 控制器内部的 FIFO,实现 PCM 音频数据的不间断操作。DMA 是通过微处理器配置的,从处理器设定的内存区域搬运数据给 FIFO 或者把 FIFO 的数据搬运到设定的内存区域。

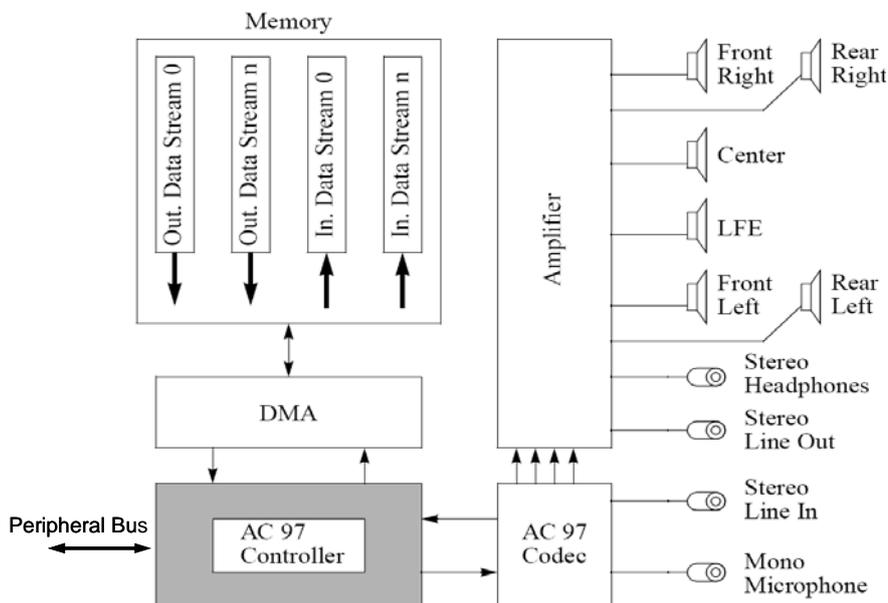


图 17-1 AC97 应用系统

## 17.2 AC97 控制器寄存器

本模块寄存器物理地址基址为 0x1fe74000。

寄存器名	宽度	偏移量	描述
CSR	2	0x00	配置状态寄存器
OCC0	24	0x04	输出通道配置寄存器 0
OCC1	24	0x08	保留
OCC2	24	0x0c	保留
ICC	24	0x10	输入通道配置寄存器
CODEC_ID	32	0x14	Codec ID 寄存器
CRAC	32	0x18	Codec 寄存器访问命令
OC0	20	0x20	输出声道 0

寄存器名	宽度	偏移量	描述
OC1	20	0x24	输出声道 1
OC2	20	0x28	保留
OC3	20	0x2c	保留
OC4	20	0x30	保留
OC5	20	0x34	保留
OC6	20	0x38	保留
OC7	20	0x3c	保留
OC8	20	0x40	保留
IC0	20	0x44	保留
IC1	20	0x48	保留
IC2	20	0x4c	输入声道 2
INTRAW	32	0x54	中断状态寄存器
INTM	32	0x58	中断掩膜
INTS	32	0x5c	保留

### 17.2.1 CSR 寄存器

中文名：配置状态寄存器  
寄存器位宽：[31: 0]  
偏移量：0x00  
复位值：0x00000000

位域	位域名称	位宽	访问	描述
31:2	Reserved	30	RO	保留
1	RESUME	1	R/W	挂起，读此位返回现在 AC97 子系统的状态 1: AC97 子系统挂起 0: 正常工作状态 在挂起状态下，写入 1 到该位，将会开始恢复操作。
0	RST_FORCE	1	W	AC97 冷启动 写入 1 会导致 AC97 Codec 冷启动

### 17.2.2 OCC 寄存器

中文名：输出通道配置寄存器  
寄存器位宽：[31: 0]  
偏移量：0x04  
复位值：0x00004141

位域	位域名称	位宽	访问	描述
31:24	Reserved	8	R/W	保留
23:16	Reserved	8	R/W	保留
15:8	OC1_CFG_R	8	R/W	输出通道 1: 右声道配置。
7:0	OC0_CFG_L	8	R/W	输出通道 0: 左声道配置。

### 17.2.3 ICC 寄存器

中文名： 输入通道配置寄存器  
寄存器位宽： [31: 0]  
偏移量： 0x10  
复位值： 0x00410000

位域	位域名称	位宽	访问	描述
31:24	Reserved	8	R/W	保留
23:16	IC_CFG_MIC	8	R/W	输入通道 2: MIC 声道配置。
15:8	Reserved	8	R/W	保留
7:0	Reserved	8	R/W	保留

声道格式 (IC\_CFG\_MIC) 说明:

位域	位域名称	位宽	访问	描述
7	Reserved	1	R/W	保留
6	DMA_EN	1	R/W	DMA 使能 1: DMA 打开 0: DMA 关闭
5:4	FIFO_THRES	2	R/W	FIFO 门限 5: 4 输出通道 输入通道 00: FIFO 1/4 空 FIFO 1/4 满 01: FIFO 1/2 空 FIFO 1/2 满 10: FIFO 3/4 空 FIFO 3/4 满 11: FIFO 全空 FIFO 全满
3:2	SW	2	R/W	采样位数 00: 8 位 10: 16 位
1	VSR	1	R/W	采样率 1: 采样率可变 0: 采样率固定 (48KHz)
0	CH_EN	1	R/W	通道使能 1: 通道打开 0: 通道关闭 (或者进入节能状态)

### 17.2.4 Codec 寄存器访问命令

中文名： Codec 寄存器访问命令  
寄存器位宽： [31: 0]  
偏移量： 0x18  
复位值： 0x00000000

位域	位域名称	位宽	访问	描述
31	CODEC_WR	1	R/W	读/写选择 1: 读, 读取数据时, 先设置 CODEC_WR 为读方式, 并在 CODEC_ADR 设置欲访问的寄存器地址; 等到返回数据完成中断时再读 CODEC_DAT 寄存器读取值。 0: 写
30:23	Reserved	8	R	保留

22:16	CODEC_ADR	7	R/W	Codec 寄存器地址
15:0	CODEC_DAT	16	R/W	Codec 寄存器数据

### 17.2.5 中断状态寄存器/中断掩膜寄存器

中文名： 中断状态/中断掩膜寄存器  
寄存器位宽： [31: 0]  
偏移量： 0x54/58  
复位值： 0x00000000

位域	位域名称	位宽	访问	描述
31	IC_FULL	1	R/W	输入通道 2: FIFO 满
30	IC_TH_INT	1	R/W	输入通道 2: FIFO 达到门限
29:8	Reserved	22	R/W	保留
7	OC1_FULL	1	R/W	输出通道 1: FIFO 满
6	OC1_EMPTY	1	R/W	输出通道 1: FIFO 空
5	OC1_TH_INT	1	R/W	输出通道 1: FIFO 达到门限
4	OC0_FULL	1	R/W	输出通道 0: FIFO 满
3	OC0_EMPTY	1	R/W	输出通道 0: FIFO 空
2	OC0_TH_INT	1	R/W	输出通道 0: FIFO 达到门限
1	CW_DONE	1	R/W	Codec 寄存器写完成
0	CR_DONE	1	R/W	Codec 寄存器读完成

### 17.2.6 中断状态/清除寄存器

中文名： 中断状态/清除寄存器  
寄存器位宽： [31: 0]  
偏移量： 0x5c  
复位值： 0x00000000

位域	位域名称	位宽	访问	描述
31:0	INT_CLR	32	RO	屏蔽后的中断状态寄存器，对本寄存器的读操作将清除寄存器 0x54 中的所有中断状态

### 17.2.7 OC 中断清除寄存器

中文名： OC 中断清除  
寄存器位宽： [31: 0]  
偏移量： 0x60  
复位值： 0x00000000

位域	位域名称	位宽	访问	描述
31:0	INT_OC_CLR	32	RO	对本寄存器的读操作将清除寄存器 0x54 中的所有 output channel 的中断状态对应的 bit[7:2]

### 17.2.8 IC 中断清除寄存器

中文名： IC 中断清除  
寄存器位宽： [31: 0]

偏移量: 0x64  
复位值: 0x00000000

位域	位域名称	位宽	访问	描述
31:0	INT_IC_CLR	32	RO	对本寄存器的读操作将清除寄存器 0x54 中的所有 input channel 的中断状态对应的 bit[31:30]

#### 17.2.9 CODEC WRITE 中断清除寄存器

中文名: CODEC WRITE 中断清除  
寄存器位宽: [31: 0]  
偏移量: 0x68  
复位值: 0x00000000

位域	位域名称	位宽	访问	描述
31:0	INT_CW_CLR	32	RO	对本寄存器的读操作将清除寄存器 0x54 中的中 bit[1]

#### 17.2.10 CODEC READ 中断清除寄存器

中文名: CODEC READ 中断清除  
寄存器位宽: [31: 0]  
偏移量: 0x6c  
复位值: 0x00000000

位域	位域名称	位宽	访问	描述
31:0	INT_CR_CLR	32	RO	对本寄存器的读操作将清除寄存器 0x54 中的中 bit[0]

# 18 I2C

## 18.1 概述

本章给出 I2C 的详细描述和配置使用。本系统芯片集成了 I2C 接口，主要用于实现两个器件之间数据的交换。I2C 总线是由数据线 SDA 和时钟 SCL 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 400kbps。龙芯 1C 共集成 3 路 I2C 接口。

## 18.2 I2C 控制器结构

I2C 主控制器结构的主要模块包括，时钟发生器（Clock Generator）、字节命令控制器（Byte Command Controller）、位命令控制器（Bit Command controller）、数据移位寄存器（Data Shift Register）。其余为 LPB 总线接口和一些寄存器。

时钟发生器模块：产生分频时钟，同步位命令的工作。

字节命令控制器模块：将一个命令解释为按字节操作的时序，即把字节操作分解为位操作。

位命令控制器模块：进行实际数据的传输，以及位命令信号产生。

数据移位寄存器模块：串行数据移位。

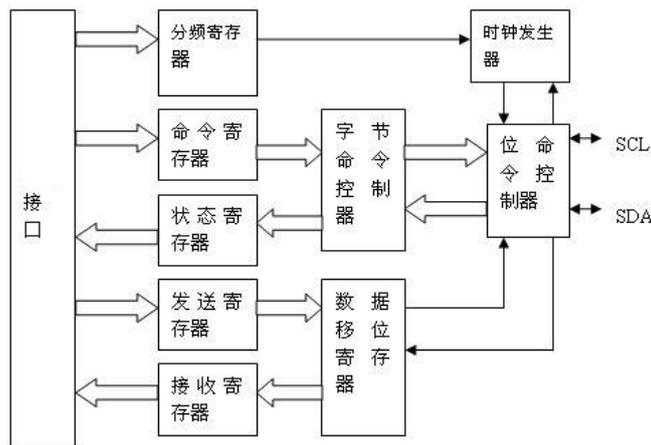


图 21-1 I2C 主控制器结构

## 18.3 I2C 控制器寄存器说明

I2C-0 模块寄存器物理地址基址为：0xbfe5\_8000,地址空间 16KB。

I2C-1 模块寄存器物理地址基址为：0xbfe6\_8000,地址空间 16KB。

I2C-2 模块寄存器物理地址基址为：0xbfe7\_0000,地址空间 16KB。

### 18.3.1 分频锁存器低字节寄存器 (PRERlo)

中文名： 分频锁存器低字节寄存器  
寄存器位宽： [7: 0]  
偏移量： 0x00  
复位值： 0xff

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

### 18.3.2 分频锁存器高字节寄存器 (PRERhi)

中文名： 分频锁存器高字节寄存器  
寄存器位宽： [7: 0]  
偏移量： 0x01  
复位值： 0xff

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

模块中被分频时钟 clock\_a 的频率是 DDR\_clk 频率的的一半 (DDR\_clk 配置见 29 章); 假设分频锁存器的值为 prescale, SCL 总线的输出频率为 clock\_s (该时钟根据用户需要和外部 I2C 设备特性确定), 则应满足如下关系:

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

或者  $\text{Prcescale} = \text{DDR\_clk}/(10*\text{clock\_s})-1$

### 18.3.3 控制寄存器 (CTR)

中文名： 控制寄存器  
寄存器位宽： [7: 0]  
偏移量： 0x02  
复位值： 0x00

位域	位域名称	位宽	访问	描述
7	EN	1	RW	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	RW	中断使能位 为 1 则打开中断
5:0	Reserved	6	RW	保留

### 18.3.4 发送数据寄存器 (TXR)

中文名： 发送寄存器  
寄存器位宽： [7: 0]  
偏移量： 0x03  
复位值： 0x00

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	DRW	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

### 18.3.5 接受数据寄存器 (RXR)

中文名：接收寄存器

寄存器位宽：[7: 0]

偏移量：0x03

复位值：0x00

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

### 18.3.6 命令控制寄存器 (CR)

中文名：命令寄存器

寄存器位宽：[7: 0]

偏移量：0x04

复位值：0x00

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

都是在 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’。

### 18.3.7 状态寄存器 (SR)

中文名：状态寄存器

寄存器位宽：[7: 0]

偏移量：0x04

复位值：0x00

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	I <sup>2</sup> C 总线忙标志位 1 总线在忙 0 总线空闲
5	AL	1	R	当 I <sup>2</sup> C 核失去 I <sup>2</sup> C 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

# 19 UART 控制器

## 19.1 概述

1C 集成了四个 UART 核，通过 APB 总线与总线桥通信。UART 控制器提供与 MODEM 或其他外部设备串行通信的功能，例如与另外一台计算机，以 RS232 为标准使用串行线路进行通信。该控制器在设计上能很好地兼容国际工业标准半导体设备 16550A。

UART 控制器支持通过配置寄存器扩展为 USART 模式。USART 控制器遵循 ISO7816 的 T=0, T=1 两个基本协议，提供了对红外和智能卡接口模式的支持。

## 19.2 控制器结构

UART 控制器有发送和接收模块 (Transmitter and Receiver)、MODEM 模块、中断仲裁模块 (Interrupt Arbitrator)、和访问寄存器模块 (Register Access Control)，这些模块之间的关系见下图所示。主要模块功能及特征描述如下：

- **发送和接收模块：**负责处理数据帧的发送和接收。发送模块是将 FIFO 发送队列中的数据按照设定的格式把并行数据转换为串行数据帧，并通过发送端口送出去。接收模块则监视接收端信号，一旦出现有效开始位，就进行接收，并实现将接收到的异步串行数据帧转换为并行数据，存入 FIFO 接收队列中，同时检查数据帧格式是否有错。UART 的帧结构是通过行控制寄存器 (LCR) 设置的，发送和接收器的状态被保存在行状态寄存器 (LSR) 中
- **MODEM 模块：**MODEM 控制寄存器 (MCR) 控制输出信号 DTR 和 RTS 的状态。MODEM 控制模块监视输入信号 DCD,CTS,DSR 和 RI 的线路状态，并将这些信号的状态记录在 MODEM 状态寄存器 (MSR) 的相对应位中。
- **中断仲裁模块：**当任何一种中断条件被满足，并且在中断使能寄存器 (IER) 中相应位置 1，那么 UART 的中断请求信号 UAT\_INT 被置为有效状态。为了减少和外部软件的交互，UART 把中断分为四个级别，并且在中断标识寄存器 (IIR) 中标识这些中断。四个级别的中断按优先级级别由高

到低的排列顺序为，接收线路状态中断；接收数据准备好中断；传送拥有寄存器为空中断；MODEM 状态中断。

- 访问寄存器模块：当 UART 模块被选中时，CPU 可通过读或写操作访问被地址线选中的寄存器。

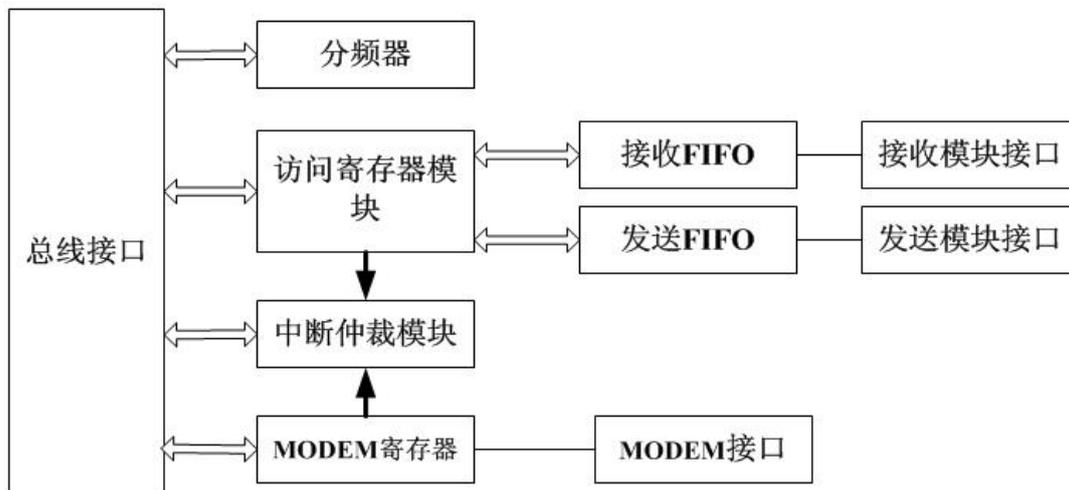


图 17-19-1 UART 控制器结构

USART 扩展部分增加了支持 ISO7816 规范的相关寄存器，并修改了收发模块，增加了收发错误校验和重传控制功能。

### 19.3 寄存器描述

龙芯 1C 中有 4 个 UART/USART(记为 URT)接口，其功能寄存器完全一样，只是访问基址不一样。

- URT0 寄存器物理地址基址为 0x1fe4\_0000。
- URT1 寄存器物理地址基址为 0x1fe4\_4000。
- URT2 寄存器物理地址基址为 0x1fe4\_8000。
- URT3 寄存器物理地址基址为 0x1fe4\_c000。

#### 19.3.1 数据寄存器(DAT)

中文名： 数据传输寄存器

寄存器位宽： [7: 0]

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
-----	---------	---	---	---------

### 19.3.2 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： 0x01

复位值： 0x00

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

### 19.3.3 中断标识寄存器 (IIR)

中文名： 中断源寄存器

寄存器位宽： [7: 0]

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	1 <sup>st</sup>	接收线路状态	奇偶、溢出或帧错误，或打断中断	读 LSR
0	1	0	2 <sup>nd</sup>	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	2 <sup>nd</sup>	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	3 <sup>rd</sup>	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	4 <sup>th</sup>	Modem 状态	CTS, DSR, RI or DCD.	读 MSR

### 19.3.4 FIFO 控制寄存器 (FCR)

中文名: FIFO 控制寄存器

寄存器位宽: [7: 0]

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 '00' – 1 字节 '01' – 4 字节 '10' – 8 字节 '11' – 14 字节
5:3	Reserved	3	W	保留
2	Txset	1	W	'1' 清除发送 FIFO 的内容, 复位其逻辑
1	Rxset	1	W	'1' 清除接收 FIFO 的内容, 复位其逻辑
0	Reserved	1	W	保留

### 19.3.5 线路控制寄存器 (LCR)

中文名: 线路控制寄存器

寄存器位宽: [7: 0]

偏移量: 0x03

复位值: 0x03

位域	位域名称	位宽	访问	描述
7	dlab	1	RW	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	RW	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	RW	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	RW	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	RW	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	RW	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	RW	设定每个字符的位数 '00' – 5 位 '01' – 6 位 '10' – 7 位 '11' – 8 位

### 19.3.6 MODEM 控制寄存器 (MCR)

中文名: Modem 控制寄存器

寄存器位宽: [7: 0]

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

### 19.3.7 线路状态寄存器 (LSR)

中文名: 线路状态寄存器

寄存器位宽: [7: 0]

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

位域	位域名称	位宽	访问	描述
				'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。

### 19.3.8 MODEM 状态寄存器 (MSR)

中文名: Modem 状态寄存器  
寄存器位宽: [7: 0]  
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

### 19.3.9 分频锁存器

中文名: 分频锁存器 1  
寄存器位宽: [7: 0]  
偏移量: 0x00  
复位值: 0x00

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

中文名: 分频锁存器 2  
寄存器位宽: [7: 0]  
偏移量: 0x01  
复位值: 0x00

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

## 19.4 USART 寄存器描述

USART 在 UART 的基础上又增加了支持 ISO7816 规范的寄存器 CR、MR 和 FIDIR。

### 19.4.1 控制寄存器 (CR)

中文名： 控制寄存器  
寄存器位宽： [7: 0]  
偏移量： 0x8  
复位值： 0x00

位域	位域名称	位宽	访问	描述
0	UART_EN	1	RW	串口使能 0: 不使能 1: 使能
1	IRDA_EN	1	RW	红外接口使能 0: 不使能 1: 使能
2	SC_EN	1	RW	智能卡接口使能 0: 不使能 1: 使能
3-7	-	-	-	保留

### 19.4.2 模式寄存器 (MR)

中文名： 模式寄存器  
寄存器位宽： [7: 0]  
偏移量： 0x9  
复位值： 0x00

位域	位域名称	位宽	访问	描述
0-3	USART_MODE	4	RW	USART 的工作模式: 0000: Normal 0001: 串口 0010: 硬件握手 0011: modem 0100: ISO7816 T=0 0110: ISO7816 T=1 1000: IrDA Others: 保留
4-7	-	-	-	保留

### 19.4.3 FI/DI 参数寄存器 (FIDIR)

中文名： 模式寄存器  
寄存器位宽： [31: 0]  
偏移量： 0xA  
复位值： 0x00000000

位域	位域名称	位宽	访问	描述
0-10	FI_DI_RATIO	11	RW	0: ISO7816 模式不产生波特率信号 1~2047: ISO7816 产生波特率信号
11-31	-	-	-	保留

## 20 NAND

### 20.1 NAND 控制器结构描述

NAND FLASH 控制器最多支持 4 个片选和 4 个 RDY 信号，控制器支持 SLC 和 MLC 两种类型 FLASH 的操作，支持多种页大小 FLASH 操作，NAND FLASH 控制器支持系统启动，启动模式包括 ECC 模式启动和普通模式启动。

系统启动模式选择包括两种，如下表所示：

启动模式	配置	说明
ECC 模式启动	NAND_CLE 外部上拉	外部 NAND FLASH 中的第一个 page 的内容必须是原始数据经过 RS(204,188)译码后生成的数据
普通启动	NAND_CLE 外部下拉	外部 NAND FLASH 第一个 page 的数据普通原始数据

### 20.2 NAND 控制器寄存器配置描述

NAND 内部的寄存器的设置如下：

地址	寄存器名称
0x1fe7_8000	NAND_CMD
0x1fe7_8004	ADDR_C
0x1fe7_8008	ADDR_R
0x1fe7_800C	NAND_TIMING
0x1fe7_8010	ID_L
0x1fe7_8014	STATUS & ID_H
0x1fe7_8018	NAND_PARAMETER
0x1fe7_801C	NAND_OP_NUM
0x1fe7_8020	CS_RDY_MAP
0x1fe7_8040	DMA access address

#### 20.2.1 NAND\_CMD (地址：0x1fe7\_8000)

位	位域名	读写	描述
31	DMA_REQ	R/-	非 ECC 模式下 NAND 发出 DMA 请求
30	ECC_DMA_REQ	R/-	ECC 模式下 NAND 发出 DMA 请求
29:24			Reserved
23: 20	NAND_CE	R/-	外部 NAND 芯片片选情况
19: 16	NAND_RDY	R/-	外部 NAND 芯片 RDY 情况
15			Reserved
14	Wait_rs_rd_done	R/W	等待 ECC 读完成 (用于 ECC 读)
13	INT_EN	R/W	NAND 中断使能信号
12	RS_WR	R/W	写操作时候 ECC 功能开启
11	RS_RD	R/W	读操作时候 ECC 功能开启
10	done	R/W	操作完成
9	Spare	R/W	操作发生在 NAND 的 SPARE 区
8	Main	R/W	操作发生在 NAND 的 MAIN 区

7	Read status	R/W	读 NAND 的状态
6	Reset	R/W	Nand 复位操作
5	read id	R/W	读 ID 操作
4	blocks erase	R/W	连续擦除标志, 缺省 0; 1 有效, 连续擦除块的数目由 nand_op_num 决定
3	erase operation	R/W	擦除操作
2	write operation	R/W	写操作
1	read operation	R/W	读操作
0	command valid	R/W	命令有效

### 20.2.2 ADDR\_C (地址: 0x1fe7\_8004)

位	位域名	读写	描述
31:12		R/-	Reserved
11:0	Nand_address[11:0]	R/W	读、写、擦除操作起始地址页内地址

### 20.2.3 ADDR\_R (地址: 0x1fe7\_8008)

位	位域名	读写	描述
31:25		R/-	Reserved
24:0	Nand_address[33:13]	R/W	读、写、擦除操作起始地址页地址 512B 页时, 列地址为[8:0], 行地址为[30:9] 2K 页时, 列地址为[10:0], 行地址为[31:11] 4K 页时, 列地址为[11:0], 行地址为[32:12] 8K 页时, 列地址为[12:0], 行地址为[33:13]

### 20.2.4 NAND\_TIMING (地址: 0x1fe7\_800C)

位	位域名	读写	描述
31:16		R/-	Reserved
15:8	Hold cycle	R/W	NAND 命令有效需等待的周期数, 缺省 4
7:0	Wait cycle	R/W	NAND 一次读写所需总时钟周期数, 缺省 18,ECC 模式下配置为 8'hb

### 20.2.5 ID\_L (地址: 0x1fe7\_8010)

位	位域名	读写	描述
31: 0	ID[31:0]	R/-	ID[31:0]

### 20.2.6 STATUS & ID\_H (地址: 0x1fe7\_8014)

位	位域名	访问	描述
31:16		R/-	Reserved
15:8	STATUS	R/-	NAND 设备当前的读写完成状态
7:0	ID[40:32]	R/-	ID 高 8 位

### 20.2.7 NAND\_PARAMETER (地址: 0x1fe7\_8018)

位	位域名	读写	描述
31:30		R/-	Reserved
29:16	op_scope	R/W	每次操作的范围

15		R/-	Reserved
14:12	ID_number	R/-	ID 号的字节数
11:8	外部颗粒容量大小	R/W	0: 1Gb (2K 页) 1: 2Gb (2K 页) 2: 4Gb (2K 页) 3: 8Gb (2K 页) 4: 16Gb (4K 页) 5: 32Gb (8K 页) 6: 64Gb(8K 页) 7: 128Gb(8K 页) 9: 64Mb(512B 页) a:128Mb(512B 页) b:256Mb(512B 页) c:512Mb(512B 页) d:1Gb(512B 页)
7:0		R/-	Reserved

### 20.2.8 NAND\_OP\_NUM (地址: 0x1fe7\_801C)

位	位域名	读写	描述
31: 0	NAND_OP_NUM	R/W	NAND 读写操作 Byte 数; 擦除为块数

### 20.2.9 CS\_RDY\_MAP (地址: 0x1fe7\_8020)

NAND 的 4 个 CS 由所访问的地址硬件自动生成, CS0/RDY0 对应最低一块空间, CS1/RDY1 对应次低一块空间, 其它以此类推。如果需要调整外部芯片和 NAND 地址的关系, 可通过设置本寄存器, 对 cs\_rdy 1/2/3 进行重新映射。

位	位域名	读写	描述
31:28	rdy3_sel	R/W	rdy3 信号从芯片引脚到 NAND 控制器的映射 4'b0001:NAND_RDY[0] 4'b0010:NAND_RDY[1] 4'b0100:NAND_RDY[2] 4'b1000:NAND_RDY[3]
27:24	cs3_sel	R/W	cs3 信号从 NAND 控制器到芯片引脚的映射 4'b0001:NAND_CS[0] 4'b0010:NAND_CS[1] 4'b0100:NAND_CS[2] 4'b1000:NAND_CS[3]
23:20	rdy2_sel	R/W	rdy2 信号从芯片引脚到 NAND 控制器的映射 4'b0001:NAND_RDY[0] 4'b0010:NAND_RDY[1] 4'b0100:NAND_RDY[2] 4'b1000:NAND_RDY[3]
19:16	cs2_sel	R/W	cs2 信号从 NAND 控制器到芯片引脚的映射 4'b0001:NAND_CS[0] 4'b0010:NAND_CS[1] 4'b0100:NAND_CS[2] 4'b1000:NAND_CS[3]
15:12	rdy1_sel	R/W	rdy1 信号从芯片引脚到 NAND 控制器的映射 4'b0001:NAND_RDY[0] 4'b0010:NAND_RDY[1] 4'b0100:NAND_RDY[2] 4'b1000:NAND_RDY[3]

11:8	cs1_sel	R/W	cs1 信号从 NAND 控制器到芯片引脚的映射 4'b0001:NAND_CS[0] 4'b0010:NAND_CS[1] 4'b0100:NAND_CS[2] 4'b1000:NAND_CS[3]
7:0		R/-	Reserved

### 20.2.10 DMA\_ADDRESS (地址: 0x1fe7\_8040)

位	位域名	读写	描述
31: 0	DMA_ADDRESS	R/W	DMA 读写 NAND flash 数据(ID/STATUS 除外)时候的访问地址, 读/写地址相同, 读写方向通过 DMA 配置实现

## 20.3 NAND ADDR 说明

以 2K 页的 NAND flash 为例, 定义如下:

每一页的 main 区大小为 2KB, spare 区大小为 64B

main\_op = NAND\_CMD[8];

spare\_op = NAND\_CMD[9];

addr\_in\_page = { A11, A10.. A2, A1, A0}=ADDR\_C

page\_number = { ...A30,A29,A28,A27...A13,A12}= ADDR\_R

NAND flash 的 main 区总容量计算公式为

容量=2^(ADDR\_C-1) \* 2^(ADDR\_R)\*8bit = 2K\*2^(ADDR\_R)\*8bit

NAND 地址空间示例

	I/O	0	1	2	3	4	5	6	7
Column1	1st Cycle	A0	A1	A2	A3	A4	A5	A6	A7
Column2	2nd Cycle	A8	A9	A10	A11	*L	*L	*L	*L
Row1	3rd Cycle	A12	A13	A14	A15	A16	A17	A18	A19
Row2	4th Cycle	A20	A21	A22	A23	A24	A25	A26	A27
Row3 (注)	5th Cycle	A28	A29	A30	A31	A32	A33	....	....

(注: 2K 页的 1Gb 容量 NAND flash 对应的 Row 最大值为 A27, 发送地址给 NAND flash 时只用发 Column1~2 和 Row1~2, 不用发 Row3。配置 NAND 参数时, 注意不要配错型号, 否则可能会读不出数据并且控制器会死等)

对系统板上 NAND 颗粒来说, 如果仅仅操作 spare 区, A11=1 是唯一标志。所以软件配置内部寄存器时, 需要配置 A11 和 spare\_op 均为 1(见 Examples5), 错误的示例见 Examples2。

对系统板上 NAND 颗粒来说, 如果仅仅操作 main 区, A11=0 是唯一标志。; 所以软件配置内部寄存器时, 需要配置 A11 和 spare\_op 均为 0(见 Examples1), 错误的示例见 Examples4。

对系统板上 NAND 颗粒来说，如果操作 main+spare 区，A11 可以为 0（见 Examples3）；也可以为 1（见 Examples6）。

Examples1:（非 ECC 模式下。NAND 颗粒中一个 page 的数据只能位于 0x0-0x83f，第一个 op 表示读写开始的数据，接下来的 op 表示随后的读写数据；NO\_op 表示不能被本次 NAND 配置读写的数据）

(spare\_op = 1' b0 & main\_op = 1' b0) equal to (spare\_op = 1' b0 & main\_op = 1' b1); ADDR\_C = 0x30

Data in a page	0	0x30	....	0x7ff	0x800	0x830	0x83f
Page 0	NO_op	op	op	op	NO_op	NO_op	NO_op
Page 1	op	op	op	op	NO_op	NO_op	NO_op
Page 2	op	op	op	op	NO_op	NO_op	NO_op

Examples2:

spare\_op=1'b1 & main\_op=1'b0; ADDR\_C = 0x30 (配置出错！！开始操作不在 spare 区，下图是可能的错误访问顺序)

Data in a page	0	0x30	....	0x7ff	0x800	0x830	0x83f
Page 0	NO_op	op	op	op	op	op	op
Page 1	NO_op	NO_op	NO_op	NO_op	op	op	op
Page 2	NO_op	NO_op	NO_op	NO_op	op	op	op
Page 3	NO_op	NO_op	NO_op	NO_op	op	op	op

Examples3:

spare\_op = 1'b1 & main\_op = 1'b1; ADDR\_C = 0x30

Data in a page	0	0x30	....	0x7ff	0x800	0x830	0x83f
Page 0	NO_op	op	op	op	op	op	op
Page 1	op	op	op	op	op	op	op
Page 2	op	op	op	op	op	op	op

Examples4:

(spare\_op=1'b0 & main\_op=1'b0), (equal to spare\_op=1'b0 & main\_op=1'b1); ADDR\_C = 0x830: (配置出错！！开始操作在 spare 区，下图是可能的错误访问顺序)

Data in a page	0	0x30	....	0x7ff	0x800	0x830	0x83f
Page 0	NO_op						
Page 1	NO_op	op	op	op	op	NO_op	NO_op
Page 2	op	op	op	op	op	NO_op	NO_op
Page 3	op	op	op	op	op	NO_op	NO_op

Examples5:

spare\_op = 1'b1 and main\_op = 1'b0; ADDR\_C = 0x830

Data in a page	0	0x30	....	0x7ff	0x800	0x830	0x83f
Page 0	NO_op	NO_op	NO_op	NO_op	NO_op	op	op
Page 1	NO_op	NO_op	NO_op	NO_op	op	op	op
Page 2	NO_op	NO_op	NO_op	NO_op	op	op	op

Examples6:

spare\_op = 1'b1 & main\_op = 1'b1; ADDR\_C = 0x830

Data in a page	0	0x30	...	0x7ff	0x800	0x830	0x83f
Page 0	NO_op	NO_op	NO_op	NO_op	NO_op	op	op
Page 1	op						
Page 2	op						
Page 3	op						

512B 页大小的 NAND flash 和 2KB 页大小配置类似，但在以下几个地方会有不同，需要注意：

每一页的 main 区大小为 512B，spare 区大小为 16B。其中 main 区分为两个 256B 区，每个 256B 区通过 A0~A7 来寻址。读写操作时，通过发送命令 0x00、0x01 和 0x50 来选择是在哪个 256B 区（软件不必关心，硬件自动选择，如配置 NAND 控制器写 0x100 时，硬件会自动发送到高 256B 区）。

发送地址命令顺序如下：

	I/O	0	1	2	3	4	5	6	7
Column1	1st Cycle	A0	A1	A2	A3	A4	A5	A6	A7
Row1	2nd Cycle	A9	A10	A11	A12	A13	A14	A15	A16
Row2	3rd Cycle	A17	A18	A19	A20	A21	A22	A23	A24
Row3 (注)	4th Cycle	A25	A26	*L	*L	*L	*L	*L	*L

（注：Nand flash 容量为 64Mb、128Mb 和 256Mb 时，对应的最大列地址 ADDR\_R 分别为 A22、A23 和 A24，只用发送三次地址命令 Column1 和 Row1~2，不用发送 Row3；容量为 512Mb 和 1Gb 时，需要发送 Row3）

4K/8K 页大小的配置和 2K 页配置一样，4K 页的 main 区大小为 4KB，spare 区大小为 128B；8K 页的 main 区大小为 8KB，spare 区大小为 640B。都需要发送五次地址命令。

## 20.4 Nand-flash 读写操作举例

寄存器的'command valid'位不能与其他读写使能位同时置位，要先设置好要进行的操作，最后才置'command valid'位。

例如：现在要读 Main 区的数据，那么操作分成以下两步：

- a、先 NAND\_CMD = 0x102
- b、再 NAND\_CMD = 0x103

## 20.5 NAND ECC 说明

硬件集成 ECC 功能，ECC 采用 RS(204,188)方法进行编码和解码，在配置软件过程中需要注意以下几点：

1. 每次读写 NAND 的时候，推荐配置 PAGE 的页内部地址 (ADDR\_C) 为 0;
2. NAND 每个 PAGE 有 2048Bytes，采用 RS(204,188)方式编解码后，只会用到前面的 2040Bytes，会有 8 个 Bytes 不用；采用 ECC 后 NAND 利用率为 188/204;
3. 在配置操作数的时候，如果每次操作一个页面，请配置 NAND 里面的 op\_num 为 204 的倍数(byte 为单位); 在配置 DMA 控制器时候，操作数为 47(188/4)的倍数(word 为单位)。
4. ECC 操作和 OOB 操作可以分开，比如对一个页完成 ECC 读/写后可以对其 OOB 进行操作。

可以在 ECC 操作完成后通过普通方式读回所有内容，包括原始数据和 ECC 校验增加的数据（此时配置操作数 op\_num 和 DMA 相同）。

校验能力说明：最多可以纠错 8 个 Bytes，这些 Bytes 内部出错的位数可以是 1-8 个。

第一行数据共出错 64bit，恰好是 8 个 Bytes，可以纠错；最后一行数据虽然错 9bits，分散在 9Bytes 中，无法纠错。

原始数据 (204Bytes)	ff	...												
数据 1	00	00	00	00	00	00	00	00	ff	ff	ff	ff	...	可纠错
数据 2	1	ff	...	可纠错										
数据 3	fe	ff	...	可纠错										
数据 4	1	2	3	4	5	6	07	08	ff	ff	ff	ff	...	可纠错
数据 5	fe	ff	ff	ff	...	不可纠错								

## 20.6 NAND BOOT 说明

NAND BOOT 支持 512B、2K 页、4K 页和 8K 页 NAND flash 启动。根据 FLASH 的容量大小和页大小配置好 NAND\_D[7:6]的上下拉电阻（需要对应好，否则可能会出现死等的情况）。如下表所示：

NAND_D[7:6]	nand_type	NAND boot 时配置 NAND flash 颗粒容量大小 11: 表示容量大等于 2GB (2K 页/4K 页/8K 页) 10: 表示容量为 1GB (2K 页) 01: 表示容量为 512MB (512 Bytes 页) 00: 表示容量为小等于 256MB (512 Bytes 页)
-------------	-----------	--

NAND 启动时会取出第 0 页开始的 1K 字节,进行芯片的初始化。如果是 512B 页则会读两页,如果是 2K 页或者更大,则会读取 1K 字节。故系统初始化代码需要放在从第 0 页 0 字节开始。

NAND BOOT 可以通过 NAND\_CLE 的上下拉来选择 ECC 和非 ECC 方式启动。两者的区别在于: ECC 方式采用了 RS(204,188)编码, ECC 启动时读取 1020 字节,有效数据为 940 字节,其余为编码数据。(注: ECC 启动时不支持 512B 页。)

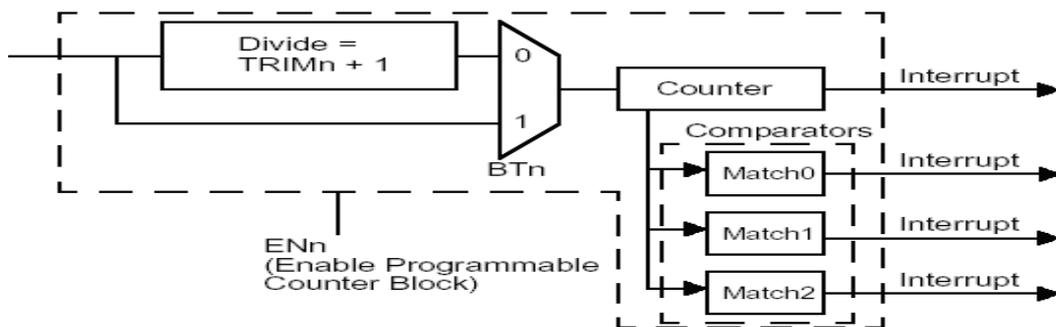
## 21 RTC

### 21.1 概述

龙芯 1C 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, 该单元任然运作, 可以仅靠板上的电池供电就正常运行。RTC 单元运行时功耗仅几个微瓦。

RTC 由外部 32.768KHZ 晶振驱动, 内部经可配置的分频器分频后, 该时钟用来计数, 年月日, 时分秒等信息被更新。同时该时钟也用于产生各种定时和计数中断。

RTC 单元由计数器和定时器组成, 其构架如下图所示:



### 21.2 寄存器描述

RTC 模块寄存器位于 0xbfe6\_4000——0x1fe6\_7fff 的 16KB 地址空间内, 其基地址为 0Xbfe6\_4000, 所有寄存器位宽均为 32 位。

#### 21.2.1 寄存器地址列表

名称	地址	位宽	RW	描述	复位值
sys_toytrim	0xbfe6_4020	32	RW	对 32.768kHz 的分频系数 (计数器时钟)	
sys_toywrite0	0xbfe6_4024	32	W	TOY 低 32 位数值写入	
sys_toywrite1	0xbfe6_4028	32	W	TOY 高 32 位数值写入	
sys_toyread0	0xbfe6_402C	32	R	TOY 低 32 位数值读出	
sys_toyread1	0xbfe6_4030	32	R	TOY 高 32 位数值读出	
sys_toymatch0	0xbfe6_4034	32	RW	TOY 定时中断 0	
sys_toymatch1	0xbfe6_4038	32	RW	TOY 定时中断 1	

sys_toymatch2	0xbf6_403C	32	RW	TOY 定时中断 2	
sys_rtctrl	0xbf6_4040	32	RW	TOY 和 RTC 控制寄存器	
sys_rtctrim	0xbf6_4060	32	RW	对 32.768kHz 的分频系数(定时器时钟)	
sys_rtcwrite0	0xbf6_4064	32	W	RTC 定时计数写入	
sys_rtcread0	0xbf6_4068	32	R	RTC 定时计数读出	
sys_rtcmatch0	0xbf6_406C	32	RW	RTC 时钟定时中断 0	
sys_rtcmatch1	0xbf6_4070	32	RW	RTC 时钟定时中断 1	
sys_rtcmatch2	0xbf6_4074	32	RW	RTC 时钟定时中断 2	

注意：其中 sys\_toytrim 及 sys\_rtctrim 寄存器复位后，其值不确定，如果不需要对外部晶振进行分频，请对这两个寄存器清零，这样 RTC 模块才能正常计时工作。

### 21.2.2 SYS\_TOYWRITE0

中文名： TOY 计数器低 32 位数值

寄存器位宽： [31: 0]

偏移量： 0x20

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

### 21.2.3 SYS\_TOYWRITE1

中文名： TOY 计数器高 32 位数值

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
31:0	TOY_YEAR	W		年，范围 0~16383

### 21.2.4 SYS\_TOYMATCH0/1/2

中文名： TOY 计数器中断寄存器 0/1/2

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	RW		年, 范围 0~16383
25:22	MONTH	RW		月, 范围 1~12
21:17	DAY	RW		日, 范围 1~31
16:12	HOURL	RW		小时, 范围 0~23
11:6	MIN	RW		分, 范围 0~59
5:0	SEC	RW		秒, 范围 0~59

### 21.2.5 SYS\_RTCCTRL

中文名： RTC 定时器中断寄存器 0/1/2

寄存器位宽： [31: 0]

偏移量： 0x40

复位值： 0x00000000

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	RM2	R	0	Sys_rtcmatch2 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	BP	R/W	0	旁路 32.768k 晶振 0: 选择晶振输入; 1: GPIO8 来驱动计数器, 这是测试模式, GPIO8 通过外部时钟或者 GPIO8 控制器。
13	REN	R/W	0	0: RTC 禁止; 1: RTC 使能
12	BRT	R/W	0	旁路 RTC 分频 0: 正常操作; 1: RTC 直接被 32.768k 晶振驱动
11	TEN	R/W	0	0: TOY 禁止; 1: TOY 使能
10	BTT	R/W	0	旁路 TOY 分频

				0: 正常操作; 1: TOY 直接被 32.768k 晶振驱动
9	保留	R	0	保留, 置 0
8	E0	R/W	0	0: 32.768k 晶振禁止; 1: 32.768k 晶振使能
7	ETS	R	0	TOY 使能写状态
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	TTS	R	0	Sys_toytrim 写状态
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

### 21.2.6 SYS\_RTCMATCH0/1/2

中文名: RTC 定时器中断寄存器 0/1/2

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	YEAR	RW		年, 范围 0~16383
25:22	MONTH	RW		月, 范围 1~12
21:17	DAY	RW		日, 范围 1~31
16:12	HOURL	RW		小时, 范围 0~23
11:6	MIN	RW		分, 范围 0~59
5:0	SEC	RW		秒, 范围 0~59

## 22 SDIO

### 22.1 功能概述

龙芯 1C 集成了一个 SDIO 控制器，用于 SD Memory 和 SDIO 卡的读写，支持 SD Memory 卡启动。SDIO 控制器特性如下：

- 兼容 SD 存储卡规格（2.0 版本）
- 兼容 SDIO 卡规格（2.0 版本）
- 8 字（32 字节）数据发送/接收 FIFO
- 扩展的 256 位 SD 卡状态寄存器
- 8 位预分频逻辑（频率=系统时钟/(p+1)）
- DMA 数据传输模式
- 1 位/4 位（宽总线）的 SD 模式

### 22.2 SDIO 协议概述

SDIO 是一个串行通信方式，主设备和从设备通过消息传递来实现数据和状态的传输。如下图是一个写多块数据的示意框图，过程如下：

1. 主设备通过命令线发送写命令消息给从设备
2. 从设备接收完消息之后通过命令线发送应答消息给主设备
3. 主设备接收到正确的应答消息后，通过数据线发送一块数据(512K Byte 或者更多)给从设备，并且检测数据线忙状态
4. 从设备接收到正确的数据后会进入编程状态，此时将数据线置为忙状态，不再响应主设备的数据请求
5. 主设备检测到从设备编程完成，继续发送下一块数据。
6. 主设备发送完最后一块数据时，通过命令线发送停止命令给从设备，收到正确应答之后完成这次多块写操作。

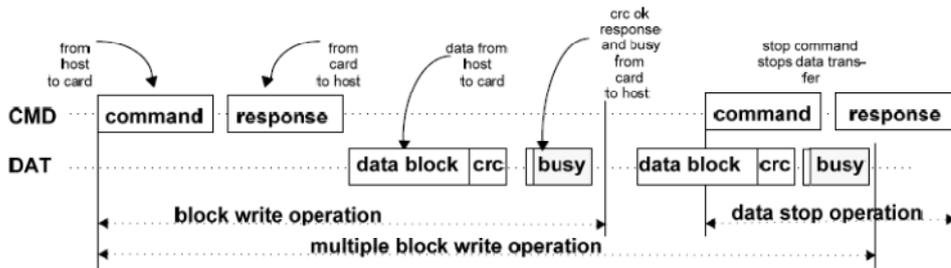


图 22.3.1. SD 卡多块写操作示意图

多块读操作的过程和多块写操作的过程类似（等补充说明）。

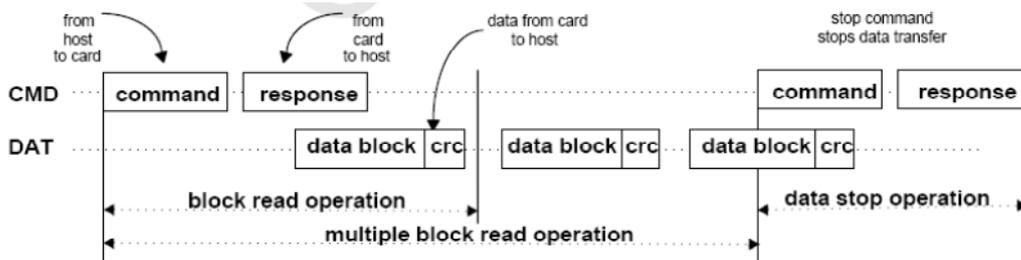


图 22.3.2. SD 卡多块读操作示意图

不同的命令都有统一的格式。一般命令的格式如下表，1bit 起始位，1bit 传输方向位，6bit 命令序号，32bit 命令参数，7bit CRC 检验位再加上 1bit 结束位。

Bit position	47	46	[45:40]	[39:8]	[7:1]	0
Width (bits)	1	1	6	32	7	1
Value	'0'	'1'	x	x	x	'1'
Description	start bit	transmission bit	command index	argument	CRC7	end bit

其中 command index 对应命令的序号，比如命令 0，cmdindex 为 0，命令 55，cmdindex 为 37。不同命令的参数可能不同，具体参考 SD 协议规范。

## 22.3 专用寄存器

SDIO 控制器的寄存器详细说明如下:

寄存器名称	地址	读/写(R/W)	功能描述	复位值
sdi_con	0xbf6c000	r/w	SDIO 控制寄存器	0x0

sdi_con	位	缺省值	描述
reserved	31:9	0x0	
soft_rst	8	0x0	软件复位，整个模块复位。复位完成后硬

			件自动清零
reserved	7:0	0x0	
enclk	0	0x0	SD 时钟输出使能

寄存器名称	地址	读/写(R/W)	功能描述	复位值
sdi_pre	0xbf6c004	r/w	SDIO 预分频寄存器	0x1

sdi_pre	位	缺省值	描述
reserved	31:8	0x0	
sdi_pre	7:0	0x1	SDIO 时钟预分频值, 输出频率=PCLK/(预分频值+1)

寄存器名称	地址	读/写(R/W)	功能描述	复位值
sdi_cmd_arg	0xbf6c008	r/w	SDIO 命令参数寄存器	0x0

sdi_cmd_arg	位	缺省值	描述
sdi_cmd_arg	31:0	0x0	命令参数

寄存器名称	地址	读/写(R/W)	功能描述	复位值
sdi_cmd_con	0xbf6c00c	r/w	SDIO 命令控制寄存器	0x0

sdi_cmd_con	位	缺省值	描述
reserved	31:14	0x0	
check_on	13	0x0	是否检查 CRC, 为 1 是有效
reserved	12:11	0x0	
long_rsp	10	0x0	是否为 136 位长响应, 为 1 是有产
wait_rsp	9	0x0	决定是否主机等待响应

CMST	8	0x0	命令开始，置 1 时开始，硬件自动清零
cmd_index	7:0	0x0	带开始 2 位的命令索引（共 8 位）

寄存器名称	地址	读/写(R/W)	功能描述	复位值
sdi_cmd_sta	0xbf6c010	ro	SDIO 命令状态寄存器	0x0

sdi_cmd_sta	位	缺省值	描述
reserved	31:13	0x0	
rsp_crc	12	0x0	响应 CRC 错误。接收到的响应 CRC 错误。为 1 时表示响应 CRC 错误，为 0 时未发现
cmd_send	11	0x0	命令发送。命令发送完成（不关心响应）。为 1 时表示命令发送完成，为 0 时未完成
cmd_tout	10	0x0	命令超时。命令响应超时(64 个时钟周期)，为 1 时表示响应超时，为 0 时未超时。
rsp_fin	9	0x0	响应结束，接收完成从设备的返回信息。为 1 时表示响应结束，为 0 时未完成。
cmd_on	8	0x0	命令传输标志位。为 1 时表示传输进行中，为 0 表示结束
rsp_index	7:0	0x0	从设备返回的带开始 2 位的响应索引（共 8 位）

寄存器名称	地址	读/写(R/W)	功能描述	复位值
sdi_rsp0	0xbf6c014	ro	SDIO 命令响应寄存器 0	0x0

sdi_cmd_arg	位	缺省值	描述
sdi_rsp0	31:0	0x0	卡状态[31:0]（短），卡状态[127:96]（长）

寄存器名称	地址	读/写(R/W)	功能描述	复位值
-------	----	----------	------	-----

sdi_rsp1	0xbf6c018	ro	SDIO 命令响 应寄存器 1	0x0
----------	-----------	----	--------------------	-----

sdi_cmd_arg	位	缺省值	描述
sdi_rsp1	31:0	0x0	未使用（短），卡状态[95:64]（长）

寄存器名称	地址	读/写(R/W)	功能描述	复位值
sdi_rsp2	0xbf6c01c	ro	SDIO 命令响 应寄存器 2	0x0

sdi_cmd_arg	位	缺省值	描述
sdi_rsp2	31:0	0x0	未使用（短），卡状态[63:32]（长）

寄存器名称	地址	读/写(R/W)	功能描述	复位值
sdi_rsp3	0xbf6c020	ro	SDIO 命令响 应寄存器 3	0x0

sdi_cmd_arg	位	缺省值	描述
sdi_rsp3	31:0	0x0	未使用（短），卡状态[31:0]（长）

寄存器名称	地址	读/写(R/W)	功能描述	复位值
sdi_dtimer	0xbf6c024	r/w	SDIO 命令数 据超时寄存 器	0x0

sdi_dtimer	位	缺省值	描述
reserved	31:23	0x0	
sdi_dtimer	22:0	0x0	数据命令超时

寄存器名称	地址	读/写(R/W)	功能描述	复位值
-------	----	----------	------	-----

sdi_bsize	0xbf6c028	r/w	SDIO 块大小寄存器	0x0
-----------	-----------	-----	-------------	-----

sdi_bsize	位	缺省值	描述
reserved	31:23	0x0	
sdi_bsize	11:0	0x0	块大小值(0~4095)

寄存器名称	地址	读/写(R/W)	功能描述	复位值
sdi_dat_con	0xbf6c02c	r/w	SDIO 数据控制寄存器	0x0

sdi_dat_con	位	缺省值	描述
reserved	31:15	0x0	
DTST	14	0x0	数据传输开始，硬件清零
reserved	13:12	0x0	
blk_num	11:0	0x0	读写操作的块数

寄存器名称	地址	读/写(R/W)	功能描述	复位值
sdi_dat_cnt	0xbf6c030	r/w	SDIO 数据计数器	0x0

sdi_dat_cnt	位	缺省值	描述
reserved	31:15	0x0	
blk_num_cnt	23:12	0x0	传输的块数
blk_cnt	11:0	0x0	传输的块的字节数

寄存器名称	地址	读/写(R/W)	功能描述	复位值
sdi_dat_sta	0xbf6c034	ro	SDIO 数据状态寄存器	0x0

sdi_dat_sta	位	缺省值	描述
reserved	31:11	0x0	
r_wait_req	10	0x0	读等待发生。发送读等待请求信号到 SDIO 卡。
lo_int_err	9	0x0	SDIO 中断标志位。为 1 时表示中断
dat_add_err	8	0x0	附加数据 CRC 错误
crc_sta	7	0x0	数据发送后，从设备返回 CRC 错误
dat_crc	6	0x0	数据接收 CRC 错误
dat_tout	5	0x0	数据传输超时。为 1 时表示数据超时
dat_fin	4	0x0	数据传输结束标志位。为 1 时标志数据传输结束
busy_fin	3	0x0	忙结束标志位（比如编程时）。为 1 时标志忙结束
prog_err	2	0x0	编程错误标志位，为 1 时表示编程错误
tx_dat_on	1	0x0	Tx 数据发送中，为 1 时表示正在发送，为 0 时发送完成
rx_dat_on	0	0x0	rX 数据接收中，为 1 时表示正在接收，为 0 时发送完成

寄存器名称	地址	读/写(R/W)	功能描述	复位值
sdi_fifo_sta	0xbf6c038	ro	SDIO FIFO 状态寄存器	0x0

sdi_fifo_sta	位	缺省值	描述
reserved	31:12	0x0	
tx_full	11	0x0	tx fifo 满标志位
tx_empty	10	0x0	tx fifo 空标志位
reserved	9	0x0	
rx_full	8	0x0	rx fifo 满标志位
rx_empty	7	0x0	rx fifo 空标志位
reserved	6:0	0x0	

寄存器名称	地址	读/写(R/W)	功能描述	复位值
sdi_int_msk	0xbf6c03c	r/c	SDIO 中断寄存器	0x0

sdi_int_msk	位	缺省值	描述
reserved	31:9	0x0	
rsp_crc_int	2	0x0	命令响应 CRC 错误中断, 写 1 清零
cmd_tout_int	1	0x0	命令超时中断, 写 1 清零
cmd_sent_int	0	0x0	数命令发送完成中断, 硬件清零
cmd_dat_int	5	0x0	附加数据 CRC 错误中断, 写 1 清零
prog_err_int	4	0x0	SD 卡编程错误中断, 写 1 清零
crc_sta_int	3	0x0	数据发送后从设备返回 CRC 错误中断, 写 1 清零
dat_crc_int	2	0x0	数据接收 CRC 错误中断, 写 1 清零
dat_tout_int	1	0x0	数据超时中断, 写 1 清零
dat_fin_int	0	0x0	数据完成中断, 硬件清零

寄存器名称	地址	读/写(R/W)	功能描述	复位值
sdi_dat	0xbf6c040	ro	SDIO 命令数据寄存器	0x0

sdi_dat	位	缺省值	描述
sdi_dat	31:0	0x0	SDIO 控制器发送或者接收的数据 (用于 DMA 操作)

寄存器名称	地址	读/写(R/W)	功能描述	复位值
sdi_sta_add0	0xbf6c044	ro	SDIO 附加寄存器 0	0x0

sdi_sta_add0	位	缺省值	描述
sdi_sta_add1	31:0	0x0	SDIO 返回附加状态寄存器[255:224]

寄存器名称	地址	读/写(R/W)	功能描述	复位值
sdi_sta_add1	0xbf6c048	ro	SDIO 附加寄存器 1	0x0

sdi_sta_add0	位	缺省值	描述
sdi_sta_add1	31:0	0x0	SDIO 返回附加状态寄存器[223:192]

寄存器名称	地址	读/写(R/W)	功能描述	复位值
sdi_sta_add2	0xbf6c04c	ro	SDIO 附加寄存器 2	0x0

sdi_sta_add0	位	缺省值	描述
sdi_sta_add3	31:0	0x0	SDIO 返回附加状态寄存器[191:160]

寄存器名称	地址	读/写(R/W)	功能描述	复位值
sdi_sta_add3	0xbf6c050	ro	SDIO 附加寄存器 3	0x0

sdi_sta_add0	位	缺省值	描述
sdi_sta_add3	31:0	0x0	SDIO 返回附加状态寄存器[159:128]

寄存器名称	地址	读/写(R/W)	功能描述	复位值
sdi_sta_add4	0xbf6c054	ro	SDIO 附加寄存器 4	0x0

sdi_sta_add0	位	缺省值	描述
sdi_sta_add4	31:0	0x0	SDIO 返回附加状态寄存器[127:96]

寄存器名称	地址	读/写(R/W)	功能描述	复位值
-------	----	----------	------	-----

sdi_sta_add5	0xbfe6c058	ro	SDIO 附加寄存器 5	0x0
--------------	------------	----	--------------	-----

sdi_sta_add0	位	缺省值	描述
sdi_sta_add5	31:0	0x0	SDIO 返回附加状态寄存器[95:64]

寄存器名称	地址	读/写(R/W)	功能描述	复位值
sdi_sta_add6	0xbfe6c05c	ro	SDIO 附加寄存器 6	0x0

sdi_sta_add0	位	缺省值	描述
sdi_sta_add6	31:0	0x0	SDIO 返回附加状态寄存器[63:32]

寄存器名称	地址	读/写(R/W)	功能描述	复位值
sdi_sta_add7	0xbfe6c060	ro	SDIO 附加寄存器	0x0

sdi_sta_add0	位	缺省值	描述
sdi_sta_add7	31:0	0x0	SDIO 返回附加状态寄存器[31:0]

## 22.4 配置寄存器说明

SDIO 要想正常工作，必须要先初始化。初始化的过程需要发送不同的命令序列来配置从设备。初始化的流程示意图如下：

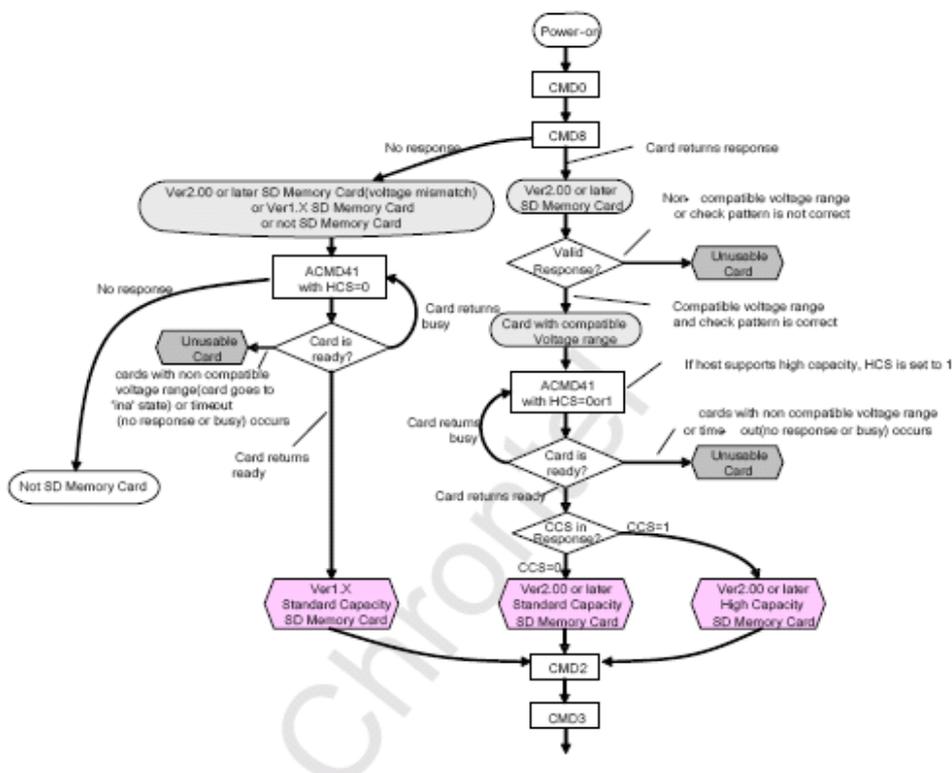


图 22.4.1. SDIO 设备初始化流程示意图

初始化完成之后就可以正常工作了。

配置寄存器的流程如下：

1. 配置 sdi\_con，使能输出时钟
2. 配置 sdi\_pre，设置一个分频系数，如果时序不满足，可以设置输出反向时钟来调整时序。
3. 按照上图初始化流程初始化控制器

发送命令的配置寄存器过程如下：

- 根据发送的命令，配置 cmd\_arg 寄存器
- 配置 cmd\_con 寄存器，发送命令
- 读 sdi\_int\_msk 寄存器，检查是否传输完成，是否有错误
- 如果需要，读 sdi\_rsp 寄存器

初始化的流程如下：

CMD0→CMD8→ACMD41(即 CMD55→CMD41)→CMD2→CMD3  
→CMD7→ACMD6（用于配置是否用 4bit 数据线传输）

4. 进行数据操作之前需要配置 Bsize 寄存器，Dtimer 寄存器
5. 数据的操作必须要配置 DMA，配置 dat\_con 寄存器并配置 DMA（注：读操作时先配置 DMA，写操作时先配置 dat\_con）
6. 读 sdi\_int\_msk 寄存器，检测是否传输完成，是否有错误。
7. 没有错误则完成一次数据传输，不需要软件发送停止命令

## 23 CAN

### 23.1 概述

1C 集成了两路 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。

两个 CAN 总线控制器的中断连接到中断控制的第一组寄存器中，其中 can0 的中断对应 bit6，can1 的中断对应 bit7。参考第 15 章的说明。

CAN0 总线控制器的寄存器基地址为 0xbfe5\_0000 开始的 16KB；

CAN1 总线控制器的寄存器基地址为 0xbfe5\_4000 开始的 16KB。

### 23.2 CAN 控制器结构

下图为 CAN 主控制器的结构，主要模块有 APB 总线接口、位流处理单元、位时序逻辑、错误管理逻辑、接收滤波和数据缓存区。

- APB 总线接口：接收 APB 总线的指令和返回数据。
- 位流处理单元：实现对发送缓存器、接收 FIFO 和 CAN 总线之间数据流的控制，同时还执行错误检测、总线仲裁、数据填充和错误处理等功能。
- 位时序逻辑：监视串口的 CAN 总线和处理与总线有关的位时序。还提供了可编程的时间段来补偿传播延迟时间、相位转换（例如由于振荡漂移）和定义采样点和一位时间内的采样次数。
- 错误管理逻辑：判断传输的 CRC 错误并对错误计数。
- 接收滤波：把接收的识别码的内容相比较以决定是否接收信息
- 数据缓存区：接收缓冲器是验收过滤器和 CPU 之间的接口，用来储存从 CAN 总线上接收和接收的信息。接收缓冲器（13 个字节）作为接收 FIFO（长 64 字节）的一个窗口可被 CPU 访问

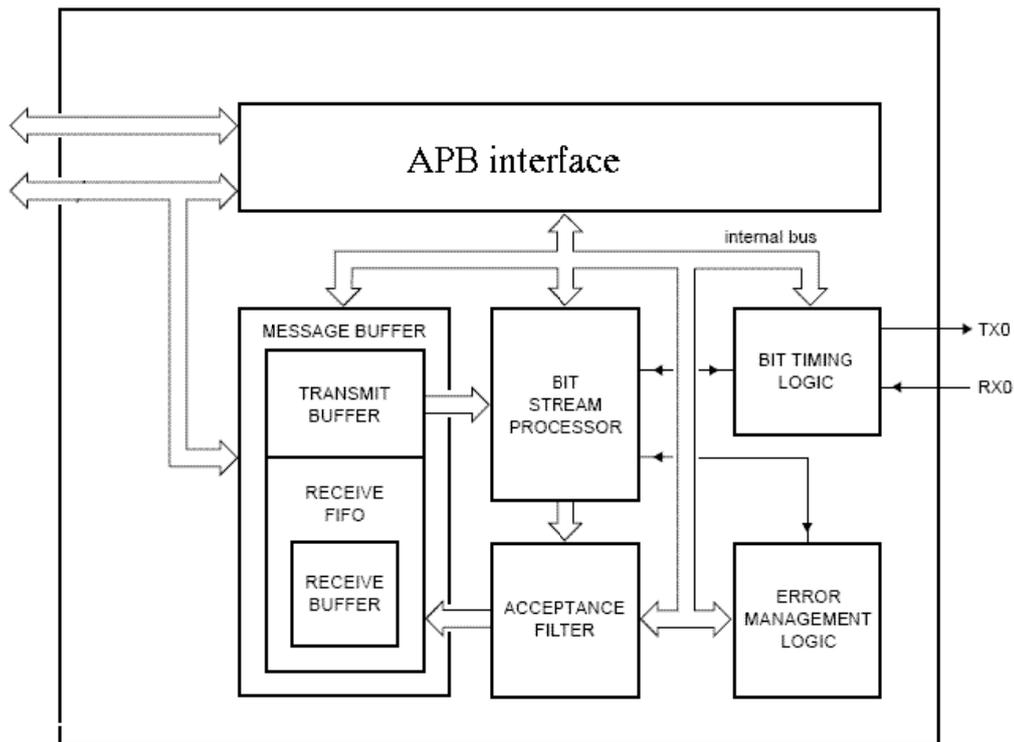


图 23-1 CAN 主控制器结构

CAN 支持两种工作模式，即标准模式和扩展模式。工作模式通过命令寄存器中的 CAN 模式位来选择。复位默认是标准模式。

## 23.3 标准模式

### 23.3.1 标准模式地址表

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。

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	—

### 23.3.2 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 个弱勢位），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。

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

### 23.3.3 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 1111

1111

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

### 23.3.4 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

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

### 23.3.5 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

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

### 23.3.6 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。

位域	位域名称	位宽	访问	描述
7:0	AC	8	RW	ID 验收代码

### 23.3.7 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。

位域	位域名称	位宽	访问	描述
7:0	AM	8	RW	ID 屏蔽位

### 23.3.8 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 FF。

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

### 23.3.9 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。

## 23.4 扩展模式

### 23.4.1 扩展模式地址表

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
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	TX 帧信息	验收代码 0	验收代码 0
17	RX 识别码 1	TX 识别码 1	验收代码 1	验收代码 1
18	RX 识别码 2	TX 识别码 2	验收代码 2	验收代码 2
19	RX 识别码 3	TX 识别码 3	验收代码 3	验收代码 3
20	RX 识别码 4	TX 识别码 4	验收屏蔽 0	验收屏蔽 0
21	RX 数据 1	TX 数据 1	验收屏蔽 1	验收屏蔽 1
22	RX 数据 2	TX 数据 2	验收屏蔽 2	验收屏蔽 2
23	RX 数据 3	TX 数据 3	验收屏蔽 3	验收屏蔽 3
24	RX 数据 4	TX 数据 4	—	—
25	RX 数据 5	TX 数据 5	—	—
26	RX 数据 6	TX 数据 6	—	—
27	RX 数据 7	TX 数据 7	—	—
28	RX 数据 8	TX 数据 8	—	—
29	RX 信息计数器	—	RX 信息计数器	—

### 23.4.2 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 个弱勢位），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。

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
2	STM	1	RW	正常工作模式
1	LOM	1	RW	只听模式
0	RM	1	RW	复位模式

### 23.4.3 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 1111

1111

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	SRR	1	W	自接收请求
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

### 23.4.4 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

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

#### 23.4.5 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

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

#### 23.4.6 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

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

#### 23.4.7 仲裁丢失捕捉寄存器 (IER)

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

位域	位域名称	位宽	访问	描述
7: 5	—	3	R	保留
4	BITNO4	1	R	第四位
3	BITNO3	1	R	第三位
2	BITNO2	1	R	第二位
1	BITNO1	1	R	第一位
0	BITNO0	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RTR位

#### 23.4.8 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

位域	位域名称	位宽	访问	描述
7: 0	EML	8	RW	错误警报阈值

#### 23.4.9 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

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

#### 23.4.10 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

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

#### 23.4.11 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。

#### 23.4.12 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

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

## 23.5 公共寄存器

#### 23.5.1 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

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

#### 23.5.2 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

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

### 23.5.3 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

位域	位域名称	位宽	访问	描述
7: 0	OCR	8	RW	保留

## 24 ADC 接口

### 24.1 功能概述

龙芯 1C 集成了 ADC 控制器，用于控制 ADC 的通道的使用，实现一些特定的应用，如连续转换、单次转换、触摸屏应用和模拟看门狗等。

龙芯 1C 中的 ADC 控制器的主要特性参数如下：

- 4 通道 ADC 模拟输入，10bit 输出精度
- 测量电压范围为 0.15~0.99VREF，推荐模拟输入电压不超过 3.3V
- ADC 的工作频率可配置，从 0~16M
- ADC 的工作模式有单次、连续转换，触摸屏应用属于特殊的连续转换
- ADC 的连续转换采用 DMA 传输数据，但触摸屏的坐标不采用 DMA
- ADC 的连续转换时间间隔可配置，从 0~1M
- ADC 的一些通道在连续转换时，未被用到的通道可进行单次转换
- 触摸屏应用时，当触摸屏被按下时产生中断，当触摸屏被释放时也产生中断
- 可支持四线式触摸屏和五线式触摸屏，连接四线式触摸屏时有两路通用 ADC 可以使用，连接五线式触摸屏时有三路通用 ADC 可以使用
- 可支持多通道扫描式单次转换
- 可支持 analog watchdog 功能，上限下限阈值可配置，超过阈值时产生中断

### 24.2 专用寄存器

寄存器名称	地址	读/写(R/W)	功能描述	复位值
adc_cnt	0xbf74000	R/W	分频和采样间隔寄存器	0x400010

Adc_cnt	位	缺省值	描述
adc_pre	31:20	0x4	adc 转换时钟的分频值，从 0~16M
cc_gap	19:0	0x10	连续转换的时间间隔，从 0~1M

寄存器名称	地址	读/写(R/W)	功能描述	复位值
adc_s_ctrl	0xbfe74004	R/W	单次采样控制寄存器	0x0

adc_s_ctrl	位	缺省值	描述
reserved	31:7	0x0	
power_down	6	0x0	停止 ADC 转换，包含所有的 AD 转换
soft_reset	5	0x0	软复位，高电平有效，硬件自动清 0
soc	4	0x0	转换开始，高电平有效，硬件自动清 0
ch_s_sel	3:0	0x0	单次转换通道选择，4 位对应 4 通道

寄存器名称	地址	读/写(R/W)	功能描述	复位值
adc_c_ctrl	0xbfe74008	r/w	连续采样控制寄存器	0x0

adc_c_ctrl	位	缺省值	描述
reserved	31:8	0x0	
cc_en	7	0x0	连续转换使能，为 1 时有效，为 0 时不启动连续转换
touch_pad_sel	6	0x0	触摸屏应用使能，为 1 时使用触摸屏
mode5_or_mode4	5	0x0	四线式或者五线式触摸屏。 为 1 时，五线式触摸屏； 为 0 时，四线式触摸屏。
awatchdog_sel	4	0x0	模拟看门狗使能，为 1 时有效

ch_s_sel	3:0	0x0	单次转换通道选择，4 位对应 4 通道
----------	-----	-----	---------------------

寄存器名称	地址	读/写(R/W)	功能描述	复位值
x_range	0xbfe74010	r/w	触摸屏 X 方向阈值	0x3ff0000

x_range	位	缺省值	描述
reserved	31:26	0x0	
x_range_max	25:16	0x3ff	x 方向最大阈值，用于判断是否触摸屏是否被按下
reserved	15:10	0x0	
x_range_min	9:0	0x0	x 方向最小阈值，用于判断是否触摸屏是否被按下

寄存器名称	地址	读/写(R/W)	功能描述	复位值
y_range	0xbfe74014	r/w	触摸屏 X 方向阈值	0x3ff0000

y_range	位	缺省值	描述
reserved	31:26	0x0	
y_range_max	25:16	0x3ff	y 方向最大阈值，用于判断是否触摸屏是否被按下
reserved	15:10	0x0	
y_range_min	9:0	0x0	y 方向最小阈值，用于判断是否触摸屏是否被按下

寄存器名称	地址	读/写(R/W)	功能描述	复位值
awatchdog_range	0xbfe74018	r/w	模拟看门狗阈值	0x3f0000f

awatchdog_range	位	缺省值	描述
reserved	31:26	0x0	
awcg_range_max	25:16	0x3f0	模拟看门狗最大阈值，用于判断是否产生中断
reserved	15:10	0x0	
awcg_range_min	9:0	0xf	模拟看门狗最小阈值，用于判断是否产生中断

寄存器名称	地址	读/写(R/W)	功能描述	复位值
axis	0xbf7401c	ro	触摸屏坐标	0x0

axis	位	缺省值	描述
reserved	31:26	0x0	
x_axis	25:16	0x0	触摸屏 X 坐标
reserved	15:10	0x0	
y_axis	9:0	0x0	触摸屏 y 坐标

寄存器名称	地址	读/写(R/W)	功能描述	复位值
adc_s_dout0	0xbf74020	ro	单次转换结果，通道 0 和通道 1	0x0

adc_s_dout0	位	缺省值	描述
adc_busy	31	0x0	转换忙状态，写 soc 后置为 1，所有单次转换完成之后，硬件清 0
reserved	30:26	0x0	
adc_data1	25:16	0x0	通道 1 单次转换的结果
reserved	15:10	0x0	
adc_data0	9:0	0x0	通道 0 单次转换的结果

寄存器名称	地址	读/写(R/W)	功能描述	复位值
adc_s_dout1	0xbf74024	ro	单次转换结果，通道 0 和通道 1	0x0

adc_s_dout1	位	缺省值	描述
adc_busy	31	0x0	转换忙状态，写 soc 后置为 1，所有单次转换完成之后，硬件清 0
reserved	30:26	0x0	
adc_data3	25:16	0x0	通道 3 单次转换的结果
reserved	15:10	0x0	
adc_data2	9:0	0x0	通道 2 单次转换的结果

寄存器名称	地址	读/写(R/W)	功能描述	复位值
adc_c_dout	0xbf74028	ro	连续转换结果	0x0

adc_c_dout	位	缺省值	描述
reserved	31:26	0x0	
adc[i+1]	25:16	0x0	连续转换时 ADC[i+1]的转换结果
reserved	15:10	0x0	
adc[i]	9:0	0x0	连续转换时 ADC[i]的转换结果

寄存器名称	地址	读/写(R/W)	功能描述	复位值
adc_debounce_cnt	0xbf7402c	ro	触摸屏去抖动时间	0x640

adc_debounce_cnt	位	缺省值	描述
reserved	31:20	0x0	
debounce_cnt	19:0	0x640	去抖动延时时间

寄存器名称	地址	读/写(R/W)	功能描述	复位值
adc_int	0xbfe74030	ro	ADC 中断标志和控制	0x0

adc_int	位	缺省值	描述
reserved	31:10	0x0	
fifo_empty	9	0x0	连续转换时 fifo 为空状态
fifo_full_int_en	8	0x0	连续转换时 fifo 满中断使能, 为 1 时开启中断, 为 0 时屏蔽
awcg_int_en	7	0x0	模拟看门狗中断使能, 为 1 时开启中断, 为 0 时屏蔽
release_int_en	6	0x0	触摸屏被释放中断使能, 为 1 时开启中断, 为 0 时屏蔽
press_int_en	5	0x0	触摸屏被按下中断使能, 为 1 时开启中断, 为 0 时屏蔽
Fifo_full_int	4	0x0	连续转换时 fifo 满中断状态, 软件写 1 清 0
awcg_h_int	3	0x0	模拟看门狗电压超过最大阈值中断状态, 软件写 1 清 0
awcg_l_int	2	0x0	模拟看门狗电压低于最小阈值中断状态, 软件写 1 清 0
release_int	1	0x0	触摸屏被释放中断状态, 软件写 1 清 0
press_int	0	0x0	触摸屏被按下中断状态, 软件写 1 清 0

## 24.3 配置操作

ADC 接口在正常工作之前, 需要对各个寄存器进行配置。

配置 ADC 控制器接口的工作流程如下：

1. 配置 `adc_int` 寄存器，开启或者关闭 ADC 控制器的中断。
2. 配置时间参数寄存器 `adc_cnt`，设置分频系数和连续转换的间隔时间。
3. 如果有触摸屏应用，配置 `adc_debounce_cnt`，设置按下或释放的去抖动时间；另外配置 X 方向和 Y 方向的阈值，用于判断是否产生中断。
4. 如果有模拟看门狗应用，配置 `awatchdog_range` 寄存器，设置看门狗的阈值。
5. 配置 `adc_c_ctrl` 寄存器，设置应用模式（如触摸屏应用、模拟看门狗）和连续转换的通道。
6. 配置 `adc_s_ctrl` 寄存器，设置单次转换的通道，以及开始和停止转换。
7. 读结果寄存器。单次转换的结果存放在 `adc_s_dout0` 和 `adc_s_dout1`；触摸屏的坐标存放在 `axis`；连续转换的结果通过 DMA 读 `adc_c_dout` 寄存器。

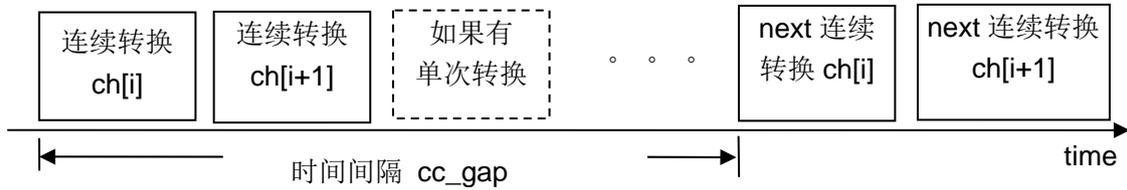
注：连续转换开始之后，除非配置 `adc_c_ctrl[7]` 为 0，否则连续转换不会停止。如果需要改变连续转换的通道，则需要先停止连续转换，重新配置完通道之后，再开始转换。如果连续转换过程中需要插入单次转换（正在使用的连续转换通道不能再做单次转换），不需要停止连续转换。每一次连续转换的结果会替换前一次的值，所以触摸屏坐标 `axis` 和 `adc_c_dout` 要及时读走。

另外通用 ADC 连续转换时，通道选择与 `adc_c_dout` 输出的对应关系如下表。如：`ch_c_sel[3:0]=b` 时，选择 ADC0,ADC1 和 ADC3 连续转换，则 ADC0 的数据放在第 1 个 word 的 `adc_c_dout[9:0]`，ADC1 的数据放在第 1 个 word 的 `adc_c_dout[25:16]`，ADC3 的数据放第 2 个 word 的 `adc_c_dout[9:0]`。

ch_c_sel[3:0]	ch[0]	ch[1]	ch[2]	ch[3]
1	Dout0[9:0]			
2		Dout0[9:0]		
3	Dout0[9:0]	Dout0[25:16]		
4			Dout0[9:0]	
5	Dout0[9:0]		Dout0[25:16]	
6		Dout0[9:0]	Dout0[25:16]	

7	Dout0[9:0]	Dout0[25:16]	Dout1[9:0]	
8				Dout0[9:0]
9	Dout0[9:0]			Dout0[25:16]
a		Dout0[9:0]		Dout0[25:16]
b	Dout0[9:0]	Dout0[25:16]		Dout1[9:0]
c			Dout0[9:0]	Dout0[25:16]
d	Dout0[9:0]		Dout0[25:16]	Dout1[9:0]
e		Dout0[9:0]	Dout0[25:16]	Dout1[9:0]
f	Dout0[9:0]	Dout0[25:16]	Dout1[9:0]	Dout1[25:16]

连续转换中插入单次转换的时序图如下：



如果 `cc_gap` 时间间隔不够插入一次单次转换，则会导致下一次连续转换时间点往后移，需要软件配置 `cc_gap` 时注意。

## 25 PWM

### 25.1 概述

龙芯 1C 里实现了四路脉冲宽度调节/计数控制器，以下简称 PWM。每一路 PWM 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 (pwm\_o)。系统时钟高达 100MHz，计数寄存器和参考寄存器均 24 位数据宽度，使得芯片非常适合高档电机的控制。

四路 PWM 控制器系统的基地址具体如下：

表 25-1 四路控制器描述

名称	基地址 (Base)	中断号
PWM0	0xbfe5_c000	18
PWM1	0xbfe5_c010	19
PWM2	0xbfe5_c020	20
PWM3	0xbfe5_c030	21

每路控制器共有四个控制寄存器，具体描述如下：

表 25-2 控制寄存器描述

名称	地址	宽度	访问	说明
CNTR	Base + 0x0	32	R/W	主计数器
HRC	Base + 0x4	32	R/W	高脉冲定时参考寄存器
LRC	Base + 0x8	32	R/W	低脉冲定时参考寄存器
CTRL	Base + 0xC	11	R/W	控制寄存器

### 25.2 PWM 寄存器说明

实现脉冲宽度功能

CNTR 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，CNTR 寄存器在系统时钟驱动下不断自加，当到达 LRC 寄存器的值后清 0。然后重新开始不断自加，控制器就产生连续不断的脉冲宽度输出。

表 25-3 主计数器设置

位域	访问	复位值	说明
31: 0	R/W	0x0	主计数器

HRC 寄存器由系统写入，当 CNTR 寄存器的值等于 HRC 的值的时候，控制器产生高脉冲电平。

表 25-4 高脉冲计数器设置

位域	访问	复位值	说明
31: 0	R/W	0x0	高脉冲计数器

LRC 寄存器由系统写入，当 CNTR 寄存器的值等于 LRC 的值的时候，控制器产生低脉冲电平。

表 25-5 低脉冲计数器设置

位域	访问	复位值	说明
31: 0	R/W	0x0	低脉冲计数器

例：如果要产生宽度为系统始终周期 50 倍的高脉宽和 90 倍的低脉宽，在 HRC 中应该配置初始值 $(90-1)=89$ ，在 LRC 寄存器中配置初始值 $(50+90-1)=139$ 。

当工作在定时器模式下，CNTR 记录内部系统时钟。HRC 和 LRC 寄存器的初始值系统编程写入，当 CNTR 寄存器的值等于 HRC 或者 LRC 的时候，芯片会产生一个中断，这样就实现了定时器功能。

CTRL 控制寄存器，在上面三种工作模式下，控制寄存器的功能不变，根据功能需求选择不同的配置。

表 25-6 控制寄存器设置

位域	访问	复位值	说明
0	R/W	0	EN, 主计数器使能位 置 1 时: CNTR 用来计数 置 0 时: CNTR 停止计数
2: 1	Reserved	2'b0	预留
3	R/W	0	OE, 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	R/W	0	SINGLE, 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	R/W	0	INTE, 中断使能位 置 1 时: 当 CNTR 计数到 LRC 和 CNTR 后送中断 置 0 时: 不产生中断
6	R/W	0	INT, 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	R/W	0	CNTR_RST,使得 CNTR 计数器清零 置 1 时: CNTR 计数器清零 置 0 时: CNTR 计数器正常工作
---	-----	---	--

## 26 看门狗 (WDT)

### 26.1 概述

在系统中看门狗定时器 (WDT, Watch Dog Timer) 实际上是一个计数器, 一般给看门狗一个大数, 程序开始运行后看门狗开始倒计时。如果程序运行正常, 过一段时间 CPU 应发出指令让看门狗复位, 重新开始倒计时。如果看门狗减到 0 就认为程序没有正常工作, 强制整个系统复位。下图是看门狗的实现, 系统对看门狗进行配置, 看门狗内部有个计数器, 同时看门狗里面的比较器比较计数器值是否为零, 如果为零就发出软复位信号让系统重启。

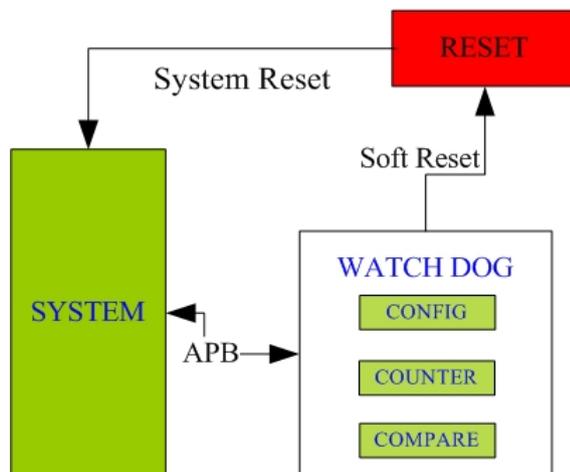


图 26-1 看门狗的结构图

### 26.2 WATCH DOG 寄存器描述

看门狗逻辑可编程寄存器主要有三个, 基地址为。这些寄存器描述如下:

#### 26.2.1 WDT\_EN 地址: (0x1fe7\_c060)

位	位域名	读写	描述
31:1			Reserved
0	WDT_EN	R/W	看门狗使能

#### 26.2.2 WDT\_SET (地址: 0x1fe7\_c068)

位	位域名	读写	描述
31:1			Reserved
0	WDT_SET	R/W	看门狗中计数器设置

### 26.2.3 WDT\_timer (地址: 0x1fe7\_c064)

位	位域名	读写	描述
31:0	WDT_timer	R/W	看门狗计数器计数值

系统这三个寄存器的设置顺序：系统先配置看门狗使能位 WDT\_EN；然后配置看门狗开始计数器的初始值 WDT\_TIMER，该值保持在一个特别的寄存器中；当系统设置 WDT\_SET 后，计数器开始计数。

看门狗不实现低功耗功能，看门狗功能的工作和硬件设计没有关系。如果需要看门狗工作，软件需要定时去更新器计数器的数值