

LOONGSON

龙芯 2P0300 处理器

用户手册

V1.0

2026 年 08 月

龙芯中科技股份有限公司

自主决定命运，创新成就未来

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



www.loongson.cn

阅读指南

《龙芯 2P0300 处理器用户手册》主要介绍龙芯 2P0300 的架构与寄存器描述，包括用户手册和片上设备使用指南两部分，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。

目 录

图目录.....	15
表目录.....	17
1 概述.....	1
1.1 体系结构框图.....	1
1.2 芯片主要功能.....	2
1.2.1 主系统 CPU.....	2
1.2.2 打印/扫描系统 CPU.....	2
1.2.3 DDR 控制器.....	2
1.2.4 USB 控制器.....	3
1.2.5 GMAC 控制器.....	3
1.2.6 SPI.....	3
1.2.7 UART.....	3
1.2.8 I ² C.....	3
1.2.9 打印接口.....	4
1.2.10 扫描接口.....	4
1.2.11 AD.....	4
1.2.12 PMIO.....	4
1.2.13 PWM.....	4
1.2.14 SDIO.....	4
1.2.15 HPET.....	4
1.2.16 GPIO.....	4
1.2.17 Watchdog.....	5
1.2.18 中断控制器.....	5
1.3 外设功能引脚复用.....	5
2 时钟结构.....	8
2.1 系统参考时钟.....	9
2.2 SYS-PLL 时钟.....	10
2.3 DDR-PLL 时钟.....	11
2.4 VID-PLL 时钟.....	13
2.5 内部 PLL 配置方法.....	14
2.5.1 硬件配置.....	14
2.5.2 软件配置.....	15
2.6 USB 参考时钟.....	15

2.7 时钟信号说明.....	15
3 芯片配置与控制.....	17
3.1 芯片初始化信号.....	17
3.2 地址空间分配.....	17
3.3 时钟与复位控制.....	20
3.3.1 时钟配置概要.....	20
3.3.2 展频 PLL 配置.....	21
3.3.3 复位控制.....	22
3.4 芯片配置寄存器.....	22
3.4.1 主系统通用配置寄存器 0.....	28
3.4.2 主系统通用配置寄存器 1.....	29
3.4.3 主系统通用配置寄存器 2.....	31
3.4.4 主系统通用配置寄存器 3.....	32
3.4.5 主系统通用配置寄存器 4.....	33
3.4.6 主系统通用配置寄存器 5.....	34
3.4.7 主系统通用配置寄存器 6.....	35
3.4.8 主系统通用配置寄存器 7.....	35
3.4.9 主系统通用配置寄存器 8.....	35
3.4.10 主系统通用配置寄存器 9.....	35
3.4.11 主系统采样参数寄存器.....	36
3.4.12 主系统高精度计数器 0.....	36
3.4.13 主系统高精度计数器 1.....	36
3.4.14 主系统 GPIO0~15 复用配置寄存器.....	37
3.4.15 主系统 GPIO16~31 复用配置寄存器.....	38
3.4.16 主系统 GPIO32~47 复用配置寄存器.....	39
3.4.17 USB PHY 配置寄存器 0.....	40
3.4.18 USB PHY 配置寄存器 1.....	40
3.4.19 USB PHY 配置寄存器 2.....	40
3.4.20 SYS PLL 时钟配置寄存器 0.....	41
3.4.21 SYSPLL 时钟配置寄存器 1.....	41
3.4.22 DDR PLL 时钟配置寄存器 0.....	42
3.4.23 DDR PLL 时钟配置寄存器 1.....	42
3.4.24 VID PLL 时钟配置寄存器 0.....	42
3.4.25 VID PLL 时钟配置寄存器 1.....	43

3.4.26 设备时钟分频配置寄存器 0.....	43
3.4.27 设备时钟分频配置寄存器 1.....	44
3.4.28 设备时钟分频配置寄存器 2.....	45
3.4.29 设备时钟输出配置寄存器.....	45
3.4.30 打印系统设备时钟门控寄存器.....	46
3.4.31 打印系统设备复位控制寄存器.....	47
3.4.32 打印系统 GPIO0~15 复用配置寄存器.....	48
3.4.33 打印系统 GPIO16~31 复用配置寄存器.....	49
3.4.34 设备时钟输出模式配置寄存器.....	50
3.4.35 打印系统通用配置寄存器 0.....	51
3.4.36 打印系统通用配置寄存器 1.....	52
3.4.37 打印系统采样参数寄存器.....	52
3.4.38 打印系统高精度计数器.....	53
3.4.39 打印系统设备 DMA 路由基址配置寄存器.....	53
3.4.40 打印系统启动入口配置寄存器.....	53
3.4.41 打印系统 0 号 MAILBOX 中断状态寄存器.....	54
3.4.42 打印系统 0 号 MAILBOX 中断使能寄存器.....	54
3.4.43 打印系统 0 号 MAILBOX 中断触发寄存器.....	54
3.4.44 打印系统 0 号 MAILBOX 中断清除寄存器.....	54
3.4.45 打印系统 0 号 MAILBOX 消息寄存器 0~7.....	54
3.4.46 打印系统 1 号 MAILBOX 中断状态寄存器.....	55
3.4.47 打印系统 1 号 MAILBOX 中断使能寄存器.....	55
3.4.48 打印系统 1 号 MAILBOX 中断触发寄存器.....	55
3.4.49 打印系统 1 号 MAILBOX 中断清除寄存器.....	55
3.4.50 打印系统 1 号 MAILBOX 消息寄存器 0~7.....	56
3.4.51 打印系统 2 号 MAILBOX 中断状态寄存器.....	56
3.4.52 打印系统 2 号 MAILBOX 中断使能寄存器.....	56
3.4.53 打印系统 2 号 MAILBOX 中断触发寄存器.....	56
3.4.54 打印系统 2 号 MAILBOX 中断清除寄存器.....	57
3.4.55 打印系统 2 号 MAILBOX 消息寄存器 0~7.....	57
3.4.56 打印系统 MAILBOX 中断路由寄存器.....	57
3.4.57 扫描系统设备时钟门控寄存器.....	58
3.4.58 扫描系统设备复位控制寄存器.....	58
3.4.59 扫描系统 GPIO0~15 复用配置寄存器.....	59

3.4.60	扫描系统 GPIO16~28 复用配置寄存器.....	60
3.4.61	扫描系统通用配置寄存器.....	61
3.4.62	扫描系统采样参数寄存器.....	61
3.4.63	扫描系统高精度计数器.....	62
3.4.64	扫描系统启动入口配置寄存器.....	62
3.4.65	扫描系统 0 号 MAILBOX 中断状态寄存器.....	62
3.4.66	扫描系统 0 号 MAILBOX 中断使能寄存器.....	62
3.4.67	扫描系统 0 号 MAILBOX 中断触发寄存器.....	63
3.4.68	扫描系统 0 号 MAILBOX 中断清除寄存器.....	63
3.4.69	扫描系统 0 号 MAILBOX 消息寄存器 0~7.....	63
3.4.70	扫描系统 1 号 MAILBOX 中断状态寄存器.....	63
3.4.71	扫描系统 1 号 MAILBOX 中断使能寄存器.....	64
3.4.72	扫描系统 1 号 MAILBOX 中断触发寄存器.....	64
3.4.73	扫描系统 1 号 MAILBOX 中断清除寄存器.....	64
3.4.74	扫描系统 1 号 MAILBOX 消息寄存器 0~7.....	64
3.4.75	扫描系统 2 号 MAILBOX 中断状态寄存器.....	65
3.4.76	扫描系统 2 号 MAILBOX 中断使能寄存器.....	65
3.4.77	扫描系统 2 号 MAILBOX 中断触发寄存器.....	65
3.4.78	扫描系统 2 号 MAILBOX 中断清除寄存器.....	65
3.4.79	扫描系统 2 号 MAILBOX 消息寄存器 0~7.....	66
3.4.80	扫描系统 MAILBOX 中断路由寄存器.....	66
3.5	中断配置及路由.....	66
3.5.1	各系统中断概述.....	66
3.5.2	中断触发类型.....	74
3.5.3	中断相关寄存器描述.....	74
3.5.4	中断路由寄存器描述.....	84
4	DDR 控制器.....	88
4.1	访问地址.....	88
4.2	DDR 控制器寄存器.....	88
5	GMAC 控制器.....	99
5.1	寄存器描述.....	99
5.2	软件编程向导.....	99
6	USB 控制器.....	101
6.1	总体概述.....	101

6.2 控制器寄存器.....	101
7 OTG 控制器.....	102
7.1 概述.....	102
7.2 寄存器描述.....	102
8 SPI-FLASH 控制器.....	103
8.1 SPI 控制器结构.....	103
8.2 配置寄存器.....	104
8.2.1 控制寄存器 (SPCR)	104
8.2.2 状态寄存器 (SPSR)	105
8.2.3 数据寄存器 (TxFIFO/RxFIFO)	105
8.2.4 外部寄存器 (SPER)	105
8.2.5 参数控制寄存器 (SFC_PARAM)	106
8.2.6 片选控制寄存器 (SFC_SOFTCS)	106
8.2.7 时序控制寄存器 (SFC_TIMING)	106
8.2.8 时序控制寄存器 2 (SFC_TIMING2)	107
8.2.9 自定义控制寄存器 (CTRL)	107
8.2.10 自定义命令寄存器 (CMD)	107
8.2.11 自定义数据寄存器 0 (BUF0)	107
8.2.12 自定义数据寄存器 1 (BUF1)	108
8.2.13 自定义时序寄存器 0 (TIMER0)	108
8.2.14 自定义时序寄存器 1 (TIMER1)	108
8.2.15 自定义时序寄存器 2 (TIMER2)	108
8.3 接口时序.....	109
8.3.1 SPI 主控制器接口时序.....	109
8.3.2 SPI Flash 访问时序.....	109
8.4 使用指南.....	110
8.4.1 SPI 主控制器的读写操作.....	110
8.4.2 硬件 SPI Flash 读.....	110
8.4.3 混合访问 SPI Flash 和 SPI 主控制器.....	111
9 SPI-I/O 控制器.....	112
9.1 SPI 控制器结构.....	112
9.2 配置寄存器.....	112
9.2.1 控制寄存器 1 (CR1)	112
9.2.2 控制寄存器 2 (CR2)	113

9.2.3 控制寄存器 3 (CR3)	113
9.2.4 控制寄存器 4 (CR4)	114
9.2.5 中断寄存器 (IER)	114
9.2.6 状态寄存器 1 (SR1)	114
9.2.7 状态寄存器 2 (SR2)	115
9.2.8 配置寄存器 1 (CFG1)	116
9.2.9 配置寄存器 2 (CFG2)	116
9.2.10 配置寄存器 3 (CFG3)	116
9.2.11 CRC 寄存器 1 (CRC1)	117
9.2.12 CRC 寄存器 2 (CRC2)	117
9.2.13 数据寄存器 (DR)	118
9.3 功能描述.....	118
9.3.1 PIN 脚配置.....	118
9.3.2 SS 模式.....	119
9.3.3 波特率.....	119
9.3.4 传输格式.....	120
9.3.5 FIFO 阈值.....	120
9.3.6 FIFO 中数据储存格式和访问格式.....	120
9.3.7 单次传输帧数.....	121
9.3.8 开始与挂起.....	121
10 I ² C 控制器.....	122
10.1 概述.....	122
10.2 主要特性.....	122
10.3 功能描述.....	123
10.3.1 模式选择.....	123
10.3.2 从模式.....	123
10.3.3 主模式.....	125
10.3.4 错误条件.....	128
10.3.5 DMA 请求.....	129
10.4 中断.....	130
10.5 寄存器定义.....	130
10.5.1 I ² C 控制寄存器 (I2C_CR1)	130
10.5.2 I ² C 控制寄存器 2 (I2C_CR2)	131
10.5.3 I ² C 从地址寄存器 (I2C_OAR)	132

10.5.4 I ² C 数据寄存器 (I2C_DR)	132
10.5.5 I ² C 状态寄存器 (I2C_SR1)	132
10.5.6 I ² C 状态寄存器 2 (I2C_SR2)	133
10.5.7 I ² C 时钟控制寄存器 (I2C_CCR)	134
10.5.8 I ² C 上升时间寄存器 (I2C_TRISE)	135
11 打印接口.....	136
11.1 概述.....	136
11.2 配置寄存器.....	136
12 扫描接口.....	137
12.1 概述.....	137
12.2 配置寄存器.....	137
13 PMIO.....	138
13.1 概述.....	138
13.2 配置寄存器.....	138
14 JPEG 编码器.....	139
14.1 概述.....	139
14.2 寄存器描述.....	139
14.2.1 命令寄存器 (CMD)	139
14.2.2 状态寄存器 (STS)	140
14.2.3 量化表地址寄存器 (Q_TABLE_ADDR)	140
14.2.4 块数量寄存器 (BLOCK_OF_IMAGE)	141
14.2.5 图像尺寸寄存器 (IMAGE_SIZE)	141
14.2.6 输入地址寄存器 (RGB_IN_ADDR)	141
14.2.7 输出地址寄存器 (JPEG_OUT_ADDR)	141
14.2.8 编码字节数寄存器 (ENCODING_BYTE)	141
14.2.9 末尾字有效位数寄存器 (TRAILING_BIT_CNT)	142
14.2.10 跟随字寄存器 (TRAILING_WORD)	142
14.2.11 格式转换命令寄存器 (IT_CMD)	142
14.2.12 格式转换状态寄存器 (IT_STS)	143
14.2.13 格式转换尺寸寄存器 (IT_SIZE)	143
14.2.14 格式转换输入地址寄存器 (IT_IN_ADDR)	143
14.2.15 格式转换命令寄存器 (IT_CMD)	143
14.3 编程示例.....	143
15 JBIG85 解码器.....	145

15.1 概述.....	145
15.2 寄存器描述.....	146
15.3 编程说明.....	149
16 UART 控制器.....	151
16.1 概述.....	151
16.2 控制器结构.....	151
16.3 寄存器描述.....	152
16.3.1 数据寄存器 (DAT)	152
16.3.2 中断使能寄存器 (IER)	153
16.3.3 中断标识寄存器 (IIR)	153
16.3.4 FIFO 控制寄存器 (FCR)	154
16.3.5 线路控制寄存器 (LCR)	154
16.3.6 MODEM 控制寄存器 (MCR)	155
16.3.7 线路状态寄存器 (LSR)	155
16.3.8 MODEM 状态寄存器 (MSR)	156
16.3.9 分频锁存器.....	157
16.3.10 采样点寄存器 (SPC)	157
17 eMMC 控制器.....	159
17.1 功能概述.....	159
17.2 访问地址及引脚复用.....	159
17.3 寄存器描述.....	159
17.4 专用 DMA 控制器.....	167
17.4.1 结构描述.....	167
17.4.2 DMA 描述符.....	168
17.5 软件配置流程.....	171
17.5.1 eMMC 正常读写流程.....	171
17.5.2 eMMC 初始化流程.....	172
17.5.3 DDR 模式设置.....	172
18 SDIO 控制器.....	173
18.1 功能概述.....	173
18.2 访问地址及引脚复用.....	173
18.3 寄存器描述.....	173
18.4 软件编程指南.....	181
18.4.1 SD Memory 卡软件编程说明.....	181

18.4.2	SDIO 卡软件编程说明.....	182
18.4.3	DDR 模式设置.....	183
18.5	支持 SDIO 型号.....	184
19	PWM 控制器.....	185
19.1	概述.....	185
19.2	访问地址及引脚复用.....	185
19.3	寄存器描述.....	185
19.4	功能说明.....	186
19.4.1	脉宽调制功能.....	186
19.4.2	脉冲测量功能.....	187
19.4.3	防死区功能.....	187
20	HPET 控制器.....	188
20.1	概述.....	188
20.2	访问地址.....	188
20.3	寄存器描述.....	188
20.3.1	General Capabilities and ID Register.....	189
20.3.2	General Configuration Register.....	189
20.3.3	General Interrupt Status Register.....	190
20.3.4	Main Counter Value Register.....	190
20.3.5	Timer N Configuration and Capabilities Register.....	190
20.3.6	Timer N Comparator Value Register.....	191
21	DMA 控制器.....	193
21.1	概述.....	193
21.2	访问地址.....	193
21.3	寄存器描述.....	193
21.3.1	DMA 中断状态寄存器 (DMA_ISR)	194
21.3.2	DMA 中断标志清除寄存器 (DMA_IFCR)	195
21.3.3	DMA 通道 x 配置寄存器 (DMA_CCRx)	195
21.3.4	DMA 通道 x 传输数量寄存器 (DMA_CNDTRx)	196
21.3.5	DMA 通道 x 外设地址寄存器 (DMA_CPARx)	197
21.3.6	DMA 通道 x 存储器地址寄存器 (DMA_CMARx)	197
21.4	功能描述.....	197
21.4.1	配置流程.....	197
21.4.2	宽度和对齐方式.....	197

21.4.3 通道映射.....	197
22 ADC 控制器.....	199
22.1 概述.....	199
22.2 工作模式.....	199
22.3 其他功能.....	199
22.3.1 注入通道管理.....	199
22.3.2 数据对齐.....	200
22.3.3 可编程的通道采样时间.....	200
22.3.4 模拟看门狗.....	200
22.3.5 DMA 请求.....	200
22.3.6 外部触发.....	201
22.4 寄存器描述.....	201
22.4.1 寄存器地址列表.....	201
22.4.2 ADC_SR.....	201
22.4.3 ADC_CR1.....	202
22.4.4 ADC_CR2.....	203
22.4.5 ADC_SMPR1.....	205
22.4.6 ADC_SMPR2.....	205
22.4.7 ADC_JOFRx (1-4).....	206
22.4.8 ADC_HTR.....	206
22.4.9 ADC_LTR.....	207
22.4.10 ADC_SQR1.....	207
22.4.11 ADC_SQR2.....	207
22.4.12 ADC_SQR3.....	208
22.4.13 ADC_JSQR.....	208
22.4.14 ADC_JDRx (1-4).....	209
22.4.15 ADC_DR.....	209
23 功耗管理模块.....	210
23.1 概述.....	210
23.2 动态功耗管理.....	210
23.2.1 DFS 功能描述.....	210
23.2.2 设备唤醒功能描述.....	212
23.3 寄存器描述.....	213
23.3.1 DFS 寄存器描述.....	214

23.3.2 WDT 寄存器描述.....	215
24 GPIO.....	217
24.1 概述.....	217
24.2 寄存器描述.....	217
24.3 访问地址.....	218
24.4 控制寄存器.....	219
24.4.1 GPIO 方向控制.....	219
24.4.2 GPIO 输出值.....	219
24.4.3 GPIO 输入值.....	219
24.4.4 GPIO 中断使能.....	220
24.4.5 GPIO 中断极性.....	220
24.4.6 GPIO 中断边沿.....	221
24.4.7 GPIO 中断清除.....	221
24.4.8 GPIO 中断状态.....	222
24.4.9 GPIO 中断双沿模式.....	222

图目录

图1-1 龙芯 2P0300 芯片结构图.....	2
图2-1 时钟结构图.....	9
图2-2 系统参考时钟.....	10
图2-3 SYS_PLL 时钟输出.....	10
图2-4 NODE_PLLCLK 时钟分频结构.....	11
图2-5 LA132_PLLCLK 时钟分频结构.....	11
图2-6 SYS0_PLLCLK 设备模块时钟分频结构.....	11
图2-7 DDR_PLL 时钟输出.....	12
图2-8 DDR_PLLCLK 时钟结构.....	12
图2-9 NETWORK_PLLCLK 时钟结构.....	12
图2-10 SYS1_PLLCLK 单元时钟分频结构.....	13
图2-11 VID_PLL 时钟输出.....	13
图2-12 PRTVID_PLL 时钟分频结构.....	14
图2-13 SCVID_PLLCLK 时钟结构.....	14
图2-14 PRTSYS 时钟结构.....	14
图3-1 展频 PLL 概念性结构.....	21
图3-2 主系统传统 IO 中断路由示意图.....	67
图3-3 主系统扩展 IO 中断路由示意图.....	69
图3-4 打印系统 IO 中断路由示意图.....	71
图3-5 扫描系统 IO 中断路由示意图.....	73
图8-1 SPI 控制器结构.....	103
图8-2 SPI 主控制器接口时序.....	109
图8-3 SPI Flash 标准读时序.....	109
图8-4 SPI Flash 快速读时序.....	109
图8-5 SPI Flash 双向 I/O 读时序.....	110
图10-1 从发送模式示意图.....	124
图10-2 从接收模式示意图.....	124
图10-3 主发送模式示意图.....	126
图10-4 主接收模式（及时）示意图.....	127
图10-5 主接收模式（非及时、3 字节）示意图.....	128
图10-6 主接收模式（非及时、2 字节）示意图.....	128
图15-1 JBIG85 解码结构示意图.....	145
图16-1 UART 控制器结构	151

图17-1 eMMC 初始化流程图.....	172
图19-1 防死区功能.....	187
图22-1 数据对齐方式.....	200
图23-1 处理器核 DFS 控制结构框图.....	211
图23-2 处理器核 DFS 操作流程图中.....	211
图23-3 芯片待机唤醒操作流程图中.....	212
图23-4 芯片休眠唤醒操作流程图中.....	213

表目录

表 1-1	主系统功能引脚复用关系表.....	5
表 1-2	打印系统功能引脚复用关系表.....	6
表 1-3	扫描系统功能引脚复用关系表.....	7
表 3-1	初始化配置信号.....	17
表 3-2	地址空间分配之主系统视角.....	18
表 3-3	地址空间分配之打印系统视角.....	18
表 3-4	地址空间分配之扫描系统视角.....	19
表 3-5	地址空间分配之 DMA 视角.....	20
表 3-6	PLL 相关配置信号说明表.....	21
表 3-7	主系统芯片配置寄存器列表.....	22
表 3-8	打印系统芯片配置寄存器列表.....	24
表 3-9	扫描系统芯片配置寄存器列表.....	26
表 3-10	通用配置寄存器 0.....	28
表 3-11	通用配置寄存器 1.....	29
表 3-12	通用配置寄存器 2.....	31
表 3-13	通用配置寄存器 3.....	32
表 3-14	通用配置寄存器 4.....	33
表 3-15	通用配置寄存器 5.....	34
表 3-16	通用配置寄存器 6.....	35
表 3-17	通用配置寄存器 7.....	35
表 3-18	通用配置寄存器 8.....	35
表 3-19	通用配置寄存器 9.....	35
表 3-20	主系统采样参数寄存器.....	36
表 3-21	主系统高精度计数器 0.....	36
表 3-22	主系统高精度计数器 1.....	37
表 3-23	主系统 GPIO0~15 复用配置寄存器.....	37
表 3-24	主系统 GPIO16~31 复用配置寄存器.....	38
表 3-25	主系统 GPIO32~47 复用配置寄存器.....	39
表 3-26	USB PHY 配置寄存器 2.....	40
表 3-27	SYS-PLL 时钟配置寄存器 0.....	41
表 3-28	SYS-PLL 时钟配置寄存器 1.....	41
表 3-29	DDR-PLL 时钟配置寄存器 0.....	42
表 3-30	DDR-PLL 时钟配置寄存器 1.....	42

表 3-31	VID-PLL 时钟配置寄存器 0.....	42
表 3-32	VID-PLL 时钟配置寄存器 1.....	43
表 3-33	设备时钟分频配置寄存器.....	43
表 3-34	设备时钟分频配置寄存器.....	44
表 3-35	设备时钟分频配置寄存器.....	45
表 3-36	设备时钟输出使能配置寄存器.....	45
表 3-37	打印系统设备时钟门控寄存器.....	46
表 3-38	打印系统设备复位控制寄存器.....	47
表 3-39	打印系统 GPIO0~15 复用配置寄存器.....	48
表 3-40	打印系统 GPIO16~31 复用配置寄存器.....	49
表 3-41	设备时钟输出模式配置寄存器.....	50
表 3-42	通用配置寄存器 0.....	51
表 3-43	通用配置寄存器 1.....	52
表 3-44	芯片采样参数寄存器.....	52
表 3-45	打印系统高精度计数器.....	53
表 3-46	打印系统设备 DMA 路由基址配置寄存器.....	53
表 3-47	打印系统启动入口配置寄存器.....	53
表 3-48	打印系统 0 号 MAILBOX 中断状态寄存器.....	54
表 3-49	打印系统 0 号 MAILBOX 中断使能寄存器.....	54
表 3-50	打印系统 0 号 MAILBOX 中断触发寄存器.....	54
表 3-51	打印系统 0 号 MAILBOX 中断清除寄存器.....	54
表 3-52	打印系统 0 号 MAILBOX 消息寄存器 0~7.....	54
表 3-53	打印系统 1 号 MAILBOX 中断状态寄存器.....	55
表 3-54	打印系统 1 号 MAILBOX 中断使能寄存器.....	55
表 3-55	打印系统 1 号 MAILBOX 中断触发寄存器.....	55
表 3-56	打印系统 1 号 MAILBOX 中断清除寄存器.....	56
表 3-57	打印系统 1 号 MAILBOX 消息寄存器 0~7.....	56
表 3-58	打印系统 2 号 MAILBOX 中断状态寄存器.....	56
表 3-59	打印系统 2 号 MAILBOX 中断使能寄存器.....	56
表 3-60	打印系统 2 号 MAILBOX 中断触发寄存器.....	57
表 3-61	打印系统 2 号 MAILBOX 中断清除寄存器.....	57
表 3-62	打印系统 0 号 MAILBOX 消息寄存器 0~7.....	57
表 3-63	打印系统 MAILBOX 中断路由寄存器.....	57
表 3-64	扫描系统设备时钟门控寄存器.....	58

表 3-65	扫描系统设备复位控制寄存器.....	58
表 3-66	扫描系统 GPIO0~15 复用配置寄存器.....	59
表 3-67	扫描系统 GPIO16~28 复用配置寄存器.....	60
表 3-68	扫描系统通用配置寄存器 0.....	61
表 3-69	扫描系统采样参数寄存器.....	61
表 3-70	扫描系统高精度计数器.....	62
表 3-71	扫描系统启动入口配置寄存器.....	62
表 3-72	扫描系统 0 号 MAILBOX 中断状态寄存器.....	62
表 3-73	扫描系统 0 号 MAILBOX 中断使能寄存器.....	63
表 3-74	扫描系统 0 号 MAILBOX 中断触发寄存器.....	63
表 3-75	扫描系统 0 号 MAILBOX 中断清除寄存器.....	63
表 3-76	扫描系统 0 号 MAILBOX 消息寄存器 0~7.....	63
表 3-77	扫描系统 1 号 MAILBOX 中断状态寄存器.....	64
表 3-78	扫描系统 1 号 MAILBOX 中断使能寄存器.....	64
表 3-79	扫描系统 1 号 MAILBOX 中断触发寄存器.....	64
表 3-80	扫描系统 1 号 MAILBOX 中断清除寄存器.....	64
表 3-81	扫描系统 1 号 MAILBOX 消息寄存器 0~7.....	64
表 3-82	扫描系统 2 号 MAILBOX 中断状态寄存器.....	65
表 3-83	扫描系统 2 号 MAILBOX 中断使能寄存器.....	65
表 3-84	扫描系统 2 号 MAILBOX 中断触发寄存器.....	65
表 3-85	扫描系统 2 号 MAILBOX 中断清除寄存器.....	65
表 3-86	扫描系统 0 号 MAILBOX 消息寄存器 0~7.....	66
表 3-87	打印系统 MAILBOX 中断路由寄存器.....	66
表 3-88	主系统传统 IO 中断寄存器列表.....	68
表 3-89	主系统 IO 扩展中断寄存器列表.....	70
表 3-90	打印系统 IO 中断寄存器列表.....	72
表 3-91	扫描系统 IO 中断寄存器列表.....	74
表 3-92	主系统传统中断控制寄存器属性.....	74
表 3-93	传统中断控制寄存器地址.....	76
表 3-94	扩展中断控制寄存器属性.....	77
表 3-95	打印系统 IO 中断控制寄存器属性.....	78
表 3-96	打印系统中断控制寄存器描述.....	80
表 3-97	扫描系统 IO 中断控制寄存器属性.....	81
表 3-98	扫描系统中断控制寄存器描述.....	83

表 3-99	传统中断路由寄存器的说明.....	84
表 3-100	传统中断路由寄存器地址.....	84
表 3-101	扩展中断路由寄存器的说明.....	86
表 3-102	打印系统中断路由寄存器的说明.....	86
表 3-103	扫描系统中断路由寄存器的说明.....	87
表 4-1	内存控制器地址空间分配.....	88
表 4-2	DDR SDRAM 配置参数寄存器.....	89
表 6-1	USB 控制器地址空间分布.....	101
表 7-1	OTG 控制器地址空间分布.....	102
表 8-1	SPI 控制器地址空间分布.....	103
表 8-2	SPI0 配置寄存器列表.....	104
表 8-3	SPI 控制寄存器 (SPCR)	104
表 8-4	SPI 状态寄存器 (SPSR)	105
表 8-5	SPI 数据寄存器 (TxFIFO/RxFIFO)	105
表 8-6	SPI 外部寄存器 (SPER)	105
表 8-7	SPI 分频系数.....	106
表 8-8	SPI 参数控制寄存器 (SFC_PARAM)	106
表 8-9	SPI 片选控制寄存器 (SFC_SOFTCS)	106
表 8-10	SPI 时序控制寄存器 (SFC_TIMING)	106
表 8-11	SPI 时序控制寄存器 2 (SFC_TIMING2)	107
表 8-12	自定义控制寄存器 (CTRL)	107
表 8-13	自定义命令寄存器 (CMD)	107
表 8-14	自定义数据寄存器 0 (BUF0)	108
表 8-15	自定义数据寄存器 1 (BUF1)	108
表 8-16	自定义时序寄存器 0 (TIMER0)	108
表 8-17	自定义时序寄存器 1 (TIMER1)	108
表 8-18	自定义时序寄存器 2 (TIMER2)	108
表 9-1	SPI 控制器地址空间分布.....	112
表 9-2	SPI-I/O 配置寄存器列表.....	112
表 9-3	SPI 控制寄存器 1 (CR1)	112
表 9-4	SPI 控制寄存器 2 (CR2)	113
表 9-5	SPI 控制寄存器 3 (CR3)	113
表 9-6	SPI 控制寄存器 4 (CR4)	114
表 9-7	SPI 中断寄存器 (IER)	114

表 9-8	SPI 状态寄存器 1(SR1)	114
表 9-9	SPI 状态寄存器 2(SR2)	115
表 9-10	SPI 配置寄存器 1(CFG1)	116
表 9-11	SPI 配置寄存器 2(CFG2)	116
表 9-12	SPI 配置寄存器 3(CFG3)	116
表 9-13	SPI CRC 寄存器 1(CRC1)	117
表 9-14	SPI CRC 寄存器 2(CRC2)	117
表 9-15	SPI 数据寄存器(DR)	118
表 9-16	PIN 脚配置位和流动方向关系	118
表 9-17	PIN 脚常见配置	119
表 9-18	PIN 脚配置位和流动方向关系	119
表 9-19	FIFO 访问格式	121
表 10-1	I ² C 控制器地址空间分布	122
表 10-2	I ² C 中断请求	130
表 10-3	I ² C 控制器寄存器列表	130
表 10-4	I ² C 控制寄存器	130
表 10-5	I ² C 控制寄存器 2	131
表 10-6	I ² C 从地址寄存器	132
表 10-7	I ² C 数据寄存器	132
表 10-8	I ² C 状态寄存器	132
表 10-9	I ² C 状态寄存器 2	134
表 10-10	I ² C 时钟控制寄存器	134
表 10-11	I ² C 上升时间寄存器	135
表 11-1	打印控制器地址空间分布	136
表 12-1	SCAN 控制器地址空间分布	137
表 13-1	各系统 PMIO 控制器地址空间分布	138
表 14-1	JPEG 编码器寄存器列表	139
表 14-2	命令寄存器	140
表 14-3	状态寄存器	140
表 14-4	量化表地址寄存器	140
表 14-5	块数量寄存器	141
表 14-6	寄存器	141
表 14-7	输入地址寄存器	141
表 14-8	输出地址寄存器	141

表 14-9	编码字节数寄存器.....	142
表 14-10	末尾字节位数寄存器.....	142
表 14-11	跟随字节寄存器.....	142
表 14-12	格式转换命令寄存器.....	142
表 14-13	格式转换命令寄存器.....	143
表 14-14	格式转换尺寸寄存器.....	143
表 14-15	格式转换输入地址寄存器.....	143
表 14-16	格式转换输出地址寄存器.....	143
表 15-1	JBIG 配置地址.....	146
表 15-2	JBIG 配置寄存器.....	146
表 15-3	jbig_top_cfg 寄存器.....	146
表 15-4	start 寄存器.....	147
表 15-5	error_code_reg 寄存器.....	147
表 15-6	int_reg_clear 寄存器.....	148
表 15-7	int_reg 寄存器.....	148
表 15-8	state_for_int 寄存器.....	149
表 15-9	line_for_int 寄存器.....	149
表 16-1	UART 配置寄存器列表.....	152
表 16-2	数据传输寄存器.....	153
表 16-3	中断使能寄存器.....	153
表 16-4	中断源寄存器.....	153
表 16-5	中断控制功能表.....	153
表 16-6	FIFO 控制寄存器.....	154
表 16-7	线路控制寄存器.....	154
表 16-8	Modem 控制寄存器.....	155
表 16-9	线路状态寄存器.....	155
表 16-10	Modem 状态寄存器.....	156
表 16-11	分频锁存器低 8 位寄存器.....	157
表 16-12	分频锁存器高 8 位寄存器.....	157
表 16-13	分频锁存器小数位寄存器.....	157
表 16-14	采样点寄存器.....	158
表 17-1	eMMC 内部寄存器物理地址构成.....	159
表 17-2	EMMC_CON 寄存器.....	159
表 17-3	EMMC_CON 寄存器位域描述.....	159

表 17-4	EMMC_PRE 寄存器.....	160
表 17-5	EMMC_PRE 寄存器位域描述.....	160
表 17-6	EMMC_CMD_ARG 寄存器.....	160
表 17-7	EMMC_CMD_ARG 寄存器位域描述.....	160
表 17-8	EMMC_CMD_CON 寄存器.....	160
表 17-9	EMMC_CMD_CON 寄存器位域描述.....	160
表 17-10	EMMC_CMD_STA 寄存器.....	161
表 17-11	EMMC_CMD_STA 寄存器位域描述.....	161
表 17-12	EMMC_RSP0 寄存器.....	161
表 17-13	EMMC_RSP0 寄存器位域描述.....	161
表 17-14	EMMC_RSP1 寄存器.....	162
表 17-15	EMMC_RESP1 寄存器位域描述.....	162
表 17-16	EMMC_RSP2 寄存器.....	162
表 17-17	EMMC_RSP2 寄存器位域描述.....	162
表 17-18	EMMC_RSP3 寄存器.....	162
表 17-19	EMMC_RSP3 寄存器位域描述.....	162
表 17-20	EMMC_DTIMER 寄存器.....	162
表 17-21	EMMC_DTIMER 寄存器位域描述.....	162
表 17-22	EMMC_BSIZE 寄存器.....	162
表 17-23	EMMC_BSIZE 寄存器位域描述.....	163
表 17-24	EMMC_DAT_CON 寄存器.....	163
表 17-25	EMMC_DAT_CON 寄存器位域描述.....	163
表 17-26	EMMC_DAT_CNT 寄存器.....	163
表 17-27	EMMC_DAT_CNT 寄存器位域描述.....	164
表 17-28	EMMC_DAT_STA 寄存器.....	164
表 17-29	EMMC_DAT_STA 寄存器位域描述.....	164
表 17-30	EMMC_FIFO_STA 寄存器.....	164
表 17-31	EMMC_FIFO_STA 寄存器位域描述.....	164
表 17-32	EMMC_INT_MASK 寄存器.....	165
表 17-33	EMMC_INT_MASK 寄存器位域描述.....	165
表 17-34	EMMC_DAT 寄存器.....	165
表 17-35	EMMC_DAT 寄存器位域描述.....	165
表 17-36	EMMC_INT_EN 寄存器.....	165
表 17-37	EMMC_INT_EN 寄存器位域描述.....	165

表 17-38	DLL_MASTER_VAL 寄存器.....	166
表 17-39	DLL_MASTER_VAL 寄存器位域描述.....	166
表 17-40	DLL_CON 寄存器.....	166
表 17-41	DLL_CON 寄存器位域描述.....	166
表 17-42	PARAM_DELAY 寄存器.....	167
表 17-43	PARAM_DELAY 寄存器位域描述.....	167
表 17-44	SDIO_EMMC_SEL 寄存器.....	167
表 17-45	SDIO_EMMC_SEL 寄存器位域描述.....	167
表 17-46	DMA_ORDER_ADDR_LOW 寄存器.....	168
表 17-47	DMA_SADDR 寄存器.....	168
表 17-48	DMA_DADDR 寄存器.....	168
表 17-49	DMA_LENGTH 寄存器.....	169
表 17-50	DMA_STEP_LENGTH 寄存器.....	169
表 17-51	DMA_STEP_TIMERS 寄存器.....	169
表 17-52	DMA_CMD 寄存器.....	169
表 17-53	DMA 写状态描述.....	170
表 17-54	DMA 读状态描述.....	170
表 17-55	DMA_ADDR_HIGH 寄存器.....	171
表 17-56	DMA_SADDR_HIGH 寄存器.....	171
表 18-1	SDIO 内部寄存器物理地址构成.....	173
表 18-2	SDI_CON 寄存器.....	173
表 18-3	SDI_CON 寄存器位域描述.....	173
表 18-4	SDI_PRE 寄存器.....	174
表 18-5	SDI_PRE 寄存器位域描述.....	174
表 18-6	SDI_CMD_ARG 寄存器.....	174
表 18-7	SDI_CMD_ARG 寄存器位域描述.....	174
表 18-8	SDI_CMD_CON 寄存器.....	174
表 18-9	SDI_CMD_CON 寄存器位域描述.....	174
表 18-10	SDI_CMD_STA 寄存器.....	175
表 18-11	SDI_CMD_STA 寄存器位域描述.....	175
表 18-12	SDI_RSP0 寄存器.....	175
表 18-13	SDI_RSP0 寄存器位域描述.....	175
表 18-14	SDI_RSP1 寄存器.....	176
表 18-15	SDI_RSP1 寄存器位域描述.....	176

表 18-16	SDI_RSP2 寄存器.....	176
表 18-17	SDI_RSP2 寄存器位域描述.....	176
表 18-18	SDI_RSP3 寄存器.....	176
表 18-19	SDI_RSP3 寄存器位域描述.....	176
表 18-20	SDI_DTIMER 寄存器.....	176
表 18-21	SDI_DTIMER 寄存器位域描述.....	176
表 18-22	SDI_BSIZE 寄存器.....	176
表 18-23	SDI_BSIZE 寄存器位域描述.....	177
表 18-24	SDI_DAT_CON 寄存器.....	177
表 18-25	SDI_DAT_CON 寄存器位域描述.....	177
表 18-26	SDI_DAT_CNT 寄存器.....	177
表 18-27	SDI_DAT_CNT 寄存器位域描述.....	177
表 18-28	SDI_DAT_STA 寄存器.....	178
表 18-29	SDI_DAT_STA 寄存器位域描述.....	178
表 18-30	SDI_FIFO_STA 寄存器.....	178
表 18-31	SDI_FIFO_STA 寄存器位域描述.....	178
表 18-32	SDI_INT_MASK 寄存器.....	179
表 18-33	SDI_INT_MASK 寄存器位域描述.....	179
表 18-34	SDI_DAT 寄存器.....	179
表 18-35	SDI_DAT 寄存器位域描述.....	179
表 18-36	SDI_INT_EN 寄存器.....	179
表 18-37	SDI_INT_EN 寄存器位域描述.....	179
表 18-38	dll_master_val 寄存器.....	180
表 18-39	dll_master_val 寄存器位域描述.....	180
表 18-40	dll_con 寄存器.....	180
表 18-41	dll_con 寄存器位域描述.....	180
表 18-42	param_delay 寄存器.....	181
表 18-43	param_delay 寄存器位域描述.....	181
表 18-44	sdio_emmc_sel 寄存器.....	181
表 18-45	sdio_emmc_sel 寄存器位域描述.....	181
表 19-1	PWM 寄存器地址.....	185
表 19-2	PWM 寄存器列表.....	185
表 19-3	PWM 控制寄存器设置.....	185
表 20-1	HPET 寄存器地址.....	188

表 20-2	HPET 寄存器.....	188
表 20-3	General Capabilities ID 寄存器.....	189
表 20-4	General Configuration 寄存器.....	189
表 20-5	General Interrupt Status 寄存器.....	190
表 20-6	Main Counter Value 寄存器.....	190
表 20-7	Timer N Configuration Capabilities 寄存器.....	190
表 20-8	Timer N Comparator Value 寄存器.....	191
表 21-1	DMA 寄存器地址.....	193
表 21-2	DMA 寄存器列表.....	193
表 21-3	DMA 中断状态寄存器.....	195
表 21-4	DMA 中断标志清除寄存器.....	195
表 21-5	DMA 通道 x 配置寄存器.....	195
表 21-6	DMA 通道 x 传输数量寄存器.....	196
表 21-7	DMA 通道 x 外设地址寄存器.....	197
表 21-8	DMA 通道 x 存储器地址寄存器.....	197
表 21-9	MAIN-APB-DMA 外设请求的通道映射.....	198
表 21-10	PRT-APB-DMA 外设请求的通道映射.....	198
表 22-1	工作模式配置.....	199
表 22-2	ADC 寄存器列表.....	201
表 22-3	ADC 状态寄存器.....	201
表 22-4	ADC 控制寄存器 1.....	202
表 22-5	ADC 控制寄存器 1.....	203
表 22-6	ADC 采样时间寄存器 1.....	205
表 22-7	ADC 采样时间寄存器 2.....	206
表 22-8	ADC 控制寄存器 1.....	206
表 22-9	ADC 看门狗高阈值寄存器.....	206
表 22-10	ADC 看门狗低阈值寄存器.....	207
表 22-11	ADC 看门狗低阈值寄存器.....	207
表 22-12	ADC 看门狗低阈值寄存器.....	208
表 22-13	ADC 看门狗低阈值寄存器.....	208
表 22-14	ADC 注入序列寄存器.....	208
表 22-15	ADC 注入数据寄存器.....	209
表 22-16	ADC 规则数据寄存器.....	209
表 23-1	芯片各功能模块功耗管理功能划分.....	210

表 23-2	LA264 DVFS Control 寄存器描述.....	214
表 23-3	LA132 DVFS Control 寄存器描述.....	214
表 23-4	WDT 寄存器地址.....	215
表 23-5	WD_EN 寄存器描述.....	215
表 23-6	WD_SET 寄存器描述.....	215
表 23-7	WD_TIMERL 寄存器描述.....	216
表 23-8	WD_TIMER 寄存器描述.....	216
表 23-9	PLTRST_SET 寄存器描述.....	216
表 24-1	GPIO 寄存器地址.....	217
表 24-2	GPIO 控制寄存器.....	217
表 24-3	GPIO 模块内部寄存器物理地址.....	218
表 24-4	按位控制 GPIO 配置寄存器地址.....	218
表 24-5	按字节控制 GPIO 配置寄存器地址.....	218
表 24-6	GPIO 方向按位控制寄存器.....	219
表 24-7	GPIO 方向按字节控制寄存器.....	219
表 24-8	GPIO 按位输出寄存器.....	219
表 24-9	GPIO 按字节输出寄存器.....	219
表 24-10	GPIO 按位输入寄存器.....	219
表 24-11	GPIO 按字节输入寄存器.....	220
表 24-12	GPIO 按位中断使能寄存器.....	220
表 24-13	GPIO 按字节中断使能寄存器.....	220
表 24-14	GPIO 按位中断极性寄存器.....	220
表 24-15	GPIO 按字节中断极性寄存器.....	220
表 24-16	GPIO 按位中断边沿寄存器.....	221
表 24-17	GPIO 按字节中断边沿寄存器.....	221
表 24-18	GPIO 按位中断清除寄存器.....	221
表 24-19	GPIO 按字节中断清除寄存器.....	222
表 24-20	GPIO 按位中断状态寄存器.....	222
表 24-21	GPIO 按字节中断状态寄存器.....	222
表 24-22	GPIO 按位中断双沿寄存器.....	222
表 24-23	GPIO 按字节中断双沿寄存器.....	222

1 概述

龙芯 2P0300 芯片是一款适用于单/多功能打印机主控 SOC 芯片，支持打印、扫描、复印、网络等常用政企办公需求，支持 A4 常用打印纸张幅面的图像处理。龙芯 2P0300 打印主控芯片是打印整机中核心控制部件，主要用于打印数据接收、解析和处理，打印引擎控制；扫描时序控制，扫描图像处理及马达控制等。

龙芯 2P0300 具有以下关键特性：

- 集成一个 64 位双发射 LA264 龙芯处理器核，L1 Cache(I/D) 32KB，L2 Cache 256KB，最高主频 1.25GHz
- 集成两个 32 位单发射 LA132 龙芯处理器核，最高主频 400MHz
- 集成 1 个 8 位 DDR3/4 控制器，支持 DDR3-800、DDR4-1600 速率
- 集成 1 个 10M/100M/1000M 自适应 GMAC，支持 RGMII/MII
- 集成 1 个 USB2.0 HOST 接口，1 个 OTG 接口
- 集成 3 个 SPI 控制器，1 路（SPI0）支持系统启动、QSPI 模式
- 集成 1 个打印控制器，支持 2 路 LSU 机芯控制
- 集成 1 个扫描控制器，支持 1 路 AFE/CIS 控制
- 集成 1 个 8 路 12 位 ADC 控制器
- 集成 2 路 I²C 控制器
- 集成 5 路 UART 控制器，其中 2 个四线串口，3 个两线串口
- 集成 2 个 SDIO/eMMC 控制器
- 集成 16 路 PWM 控制器
- 集成 3 组 PMIO 控制器
- 集成 109 路复用 GPIO
- 集成 HPET 高精度计数
- 集成看门狗电路
- 集成低功耗管理，支持待机、睡眠等低功耗模式
- 集成中断控制器，支持灵活的中断设置

1.1 体系结构框图

龙芯 2P0300 内部采用多级总线结构。一级交叉开关连接一个处理器核、一个二级 Cache 以及 IO 子网络（Cache 访问路径）。二级 Cache 及 IODMA、内存控制器、GMAC、USB、IMAGE、PRINT、SCAN 等设备共享系统互连网络。低速外设（I²C、UART 等）作为一个集合加在低速总线上。

龙芯 2P0300 芯片结构图如图 1-1 所示：

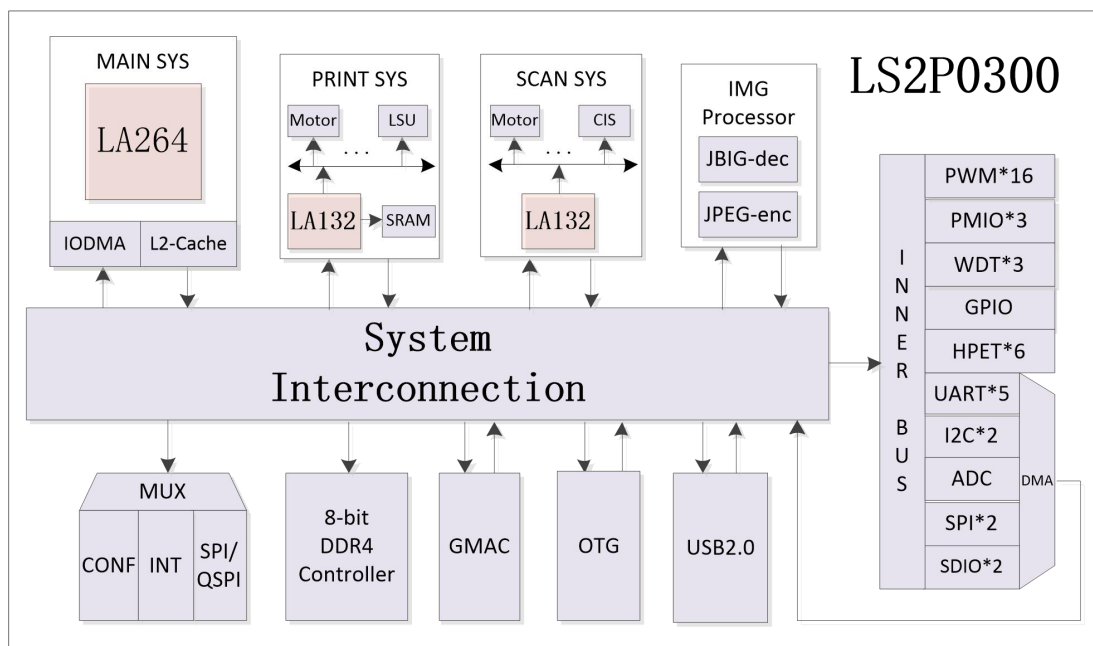


图1-1 龙芯 2P0300 芯片结构图

1.2 芯片主要功能

1.2.1 主系统 CPU

LA264, 64 位双发射、乱序发射、乱序执行

LoongArch 自主指令集

32KB 指令 Cache+32KB 数据 Cache

256KB 二级 Cache, 支持锁定

硬件维护 IO DMA 一致性

全流水的 64 位浮点加法和浮点乘法运算, 硬件除法

支持 JTAG 调试

1.2.2 打印/扫描系统 CPU

LA132, 32 位单发射、顺序执行

LoongArch 自主指令集

4KB 指令 Cache+4KB 数据 Cache

支持 JTAG 调试

1.2.3 DDR 控制器

8 位数据线, 支持 DDR3-800、DDR4-1600

遵守 DDR3/4 行业标准

- 支持命令调度

1.2.4 USB 控制器

- 1 个独立的 USB2.0 端口, 1 个独立的 OTG 端口
- 兼容 USB1.1、USB2.0
- 内部 EHCI 控制和实现高速传输可达 480Mbps
- 内部 OHCI 控制和实现全速和低速传输 12Mbps 和 1.5Mbps

1.2.5 GMAC 控制器

- 1 路 10M/100M/1000Mbps 自适应以太网控制器
- 兼容 IEEE 802.3
- 对外部 PHY 实现 RGMII 和 MII 接口
- 半双工/全双工自适应

1.2.6 SPI

- 支持 3 路 SPI 接口
- SPI0 支持系统启动、QSPI 模式
- SPI1/2 支持主从设备模式
- 极性和相位可编程的串行时钟

1.2.7 UART

- 2 个四线 UART 和流控 TXD, RXD, CTS, RTS
- 最多 5 个两线 UART 接口
- 两路全双工异步数据接收/发送
- 可编程的数据格式
- 16 位可编程时钟计数器
- 支持接收超时检测
- 带仲裁的多中断系统

1.2.8 I²C

- 支持 2 路 I²C 接口
- 履行双向同步串行协议
- 支持主从设备操作
- 能够支持多主设备的总线
- 总线的时钟频率可编程
- 可以产生开始/停止/应答等操作

- 能够对总线的状态进行探测
- 支持低速和快速模式
- 支持 7 位寻址
- 支持时钟延伸和等待状态

1.2.9 打印接口

- 支持 A4 及以下纸张尺寸
- 支持 2 路独立 LSU 机芯控制
- 支持 Laser Video (LVDS/TTL 双模式)

1.2.10 扫描接口

- 支持 1 路 AFE、CIS 接口控制
- 支持灰度、彩色扫描

1.2.11 AD

- 8 路 12 位 AD 接口采样

1.2.12 PMIO

- 3 组 PMIO 控制器

1.2.13 PWM

- 提供 16 路可配置 PWM 输出
- 数据宽度 32 位
- 定时器功能
- 脉冲捕获功能支持脉冲生成及捕获

1.2.14 SDIO

- 2 路独立 SDIO 控制器 (均可配置为 eMMC 模式)
- 1 路支持 SDIO 系统启动 (SDIO0/eMMC0 支持启动)

1.2.15 HPET

- 6 路 32 位计数器
- 支持 1 个周期性中断
- 支持 2 个非周期性中断

1.2.16 GPIO

- 109 位复用 GPIO 引脚
- 支持外部中断输入 (支持电平/沿触发)

- 与其他接口功能复用

1.2.17 Watchdog

- 64 比特计数器及初始化寄存器

1.2.18 中断控制器

- 支持软件设置中断
- 支持电平与边沿触发
- 支持中断屏蔽与使能

1.3 外设功能引脚复用

各信号引脚的功能复用关系如下表所示：

表 1-1 主系统功能引脚复用关系表

芯片引脚	GPIO 复用 (默认功能)	芯片主功能	第一复用	第二复用
UART_RX	MAIN_GPIO00	uart0_rx	gmac_col	sdio1_d[4]
UART_TX	MAIN_GPIO01	uart0_tx	gmac_crs	sdio1_d[5]
I2C_SCL	MAIN_GPIO02	i2c_scl	sca_uart_rx	sdio1_d[6]
I2C_SDA	MAIN_GPIO03	i2c_sda	sca_uart_tx	sdio1_d[7]
SPI_CLK	MAIN_GPIO04	spi0_clk	i2c_scl	prt_pmio[0]
SPI_MISO	MAIN_GPIO05	spi0_miso	i2c_sda	prt_pmio[1]
SPI_MOSI	MAIN_GPIO06	spi0_mosi	prt_i2c_scl	prt_pmio[2]
SPI_CS	MAIN_GPIO07	spi0_cs[0]	prt_i2c_sda	prt_pmio[3]
GMAC_RX_CTL	MAIN_GPIO08	gmac_rx_ctl	uart1_rx	prt_pmio[4]
GMAC_RX0	MAIN_GPIO09	gmac_rx[0]	uart1_tx	prt_pmio[5]
GMAC_RX1	MAIN_GPIO10	gmac_rx[1]	uart1_rts	prt_pmio[6]
GMAC_RX2	MAIN_GPIO11	gmac_rx[2]	uart1_cts	prt_pmio[7]
GMAC_RX3	MAIN_GPIO12	gmac_rx[3]	sys_pwm[0]	prt_pmio[8]
GMAC_TX_CTL	MAIN_GPIO13	gmac_tx_ctl	sys_pwm[1]	prt_pmio[9]
GMAC_TX0	MAIN_GPIO14	gmac_tx[0]	sys_pwm[2]	prt_pmio[10]
GMAC_TX1	MAIN_GPIO15	gmac_tx[1]	sys_pwm[3]	prt_pmio[11]
GMAC_TX2	MAIN_GPIO16	gmac_tx[2]	sys_pwm[4]	prt_pmio[12]
GMAC_TX3	MAIN_GPIO17	gmac_tx[3]	sys_pwm[5]	prt_pmio[13]
GMAC_MDCK	MAIN_GPIO18	gmac_mdck	sys_pwm[6]	prt_pmio[14]
GMAC_MDIO	MAIN_GPIO19	gmac_mdio	sys_pwm[7]	prt_pmio[15]
SDIO0_CLK	MAIN_GPIO20	sdio0_clk	sys_pmio[0]	prt_pwm[0]
SDIO0_CMD	MAIN_GPIO21	sdio0_cmd	sys_pmio[1]	prt_pwm[1]
SDIO0_D0	MAIN_GPIO22	sdio0_d[0]	sys_pmio[2]	prt_pwm[2]
SDIO0_D1	MAIN_GPIO23	sdio0_d[1]	sys_pmio[3]	prt_pwm[3]
SDIO0_D2	MAIN_GPIO24	sdio0_d[2]	sys_pmio[4]	prt_pwm[4]
SDIO0_D3	MAIN_GPIO25	sdio0_d[3]	sys_pmio[5]	prt_pwm[5]
SDIO0_D4	MAIN_GPIO26	sdio0_d[4]	sys_pmio[6]	prt_pwm[6]
SDIO0_D5	MAIN_GPIO27	sdio0_d[5]	sys_pmio[7]	prt_pwm[7]
SDIO0_D6	MAIN_GPIO28	sdio0_d[6]	sys_pmio[8]	uart0_rts
SDIO0_D7	MAIN_GPIO29	sdio0_d[7]	sys_pmio[9]	uart0_cts

芯片引脚	GPIO 复用 (默认功能)	芯片主功能	第一复用	第二复用
SDIO1_CLK	MAIN_GPIO30	sdio1_clk	sys_pmio[10]	gmac_col
SDIO1_CMD	MAIN_GPIO31	sdio1_cmd	sys_pmio[11]	gmac_crs
SDIO1_D0	MAIN_GPIO32	sdio1_d[0]	sys_pmio[12]	prt_uart0_rx
SDIO1_D1	MAIN_GPIO33	sdio1_d[1]	sys_pmio[13]	prt_uart0_tx
SDIO1_D2	MAIN_GPIO34	sdio1_d[2]	sys_pmio[14]	prt_uart1_rx
SDIO1_D3	MAIN_GPIO35	sdio1_d[3]	sys_pmio[15]	prt_uart1_tx
SYS_PMIO0	MAIN_GPIO36	sys_pmio[0]	sys_pwm[0]	prt_pmio[24]
SYS_PMIO1	MAIN_GPIO37	sys_pmio[1]	sys_pwm[1]	prt_pmio[25]
SYS_PMIO2	MAIN_GPIO38	sys_pmio[2]	sys_pwm[2]	prt_pmio[26]
SYS_PMIO3	MAIN_GPIO39	sys_pmio[3]	sys_pwm[3]	prt_pmio[27]
SYS_PMIO4	MAIN_GPIO40	sys_pmio[4]	sys_pwm[4]	prt_pmio[28]
SYS_PMIO5	MAIN_GPIO41	sys_pmio[5]	sys_pwm[5]	prt_pmio[29]
SYS_PMIO6	MAIN_GPIO42	sys_pmio[6]	sys_pwm[6]	prt_pmio[30]
SYS_PMIO7	MAIN_GPIO43	sys_pmio[7]	sys_pwm[7]	prt_pmio[31]
SYS_PMIO8	MAIN_GPIO44	sys_pmio[8]	-	uart1_tx
SYS_PMIO9	MAIN_GPIO45	sys_pmio[9]	spi0_cs[1]	uart1_rx
SYS_PMIO10	MAIN_GPIO46	sys_pmio[10]	spi0_cs[2]	uart1_rts
SYS_PMIO11	MAIN_GPIO47	sys_pmio[11]	spi0_cs[3]	uart1_cts

表 1-2 打印系统功能引脚复用关系表

芯片引脚	GPIO 复用 (默认功能)	芯片主功能	第一复用	第二复用
PRT_UART_RX	PRT_GPIO00	prt_uart0_rx	spi1_clk	gmac_ptp_trig
PRT_UART_TX	PRT_GPIO01	prt_uart0_tx	spi1_miso	gmac_ptp_pps
PRT_I2C_SCL	PRT_GPIO02	prt_i2c_scl	spi1_mosi	prt_uart1_rx
PRT_I2C_SDA	PRT_GPIO03	prt_i2c_sda	spi1_cs	prt_uart1_tx
PRT_PMIO0	PRT_GPIO04	prt_pmio[0]	prt_ovlin[0]	prt_pwm[0]
PRT_PMIO1	PRT_GPIO05	prt_pmio[1]	prt_ovlin[1]	prt_pwm[1]
PRT_PMIO2	PRT_GPIO06	prt_pmio[2]	prt_ovlin[2]	prt_pwm[2]
PRT_PMIO3	PRT_GPIO07	prt_pmio[3]	prt_ovlin[3]	prt_pwm[3]
PRT_PMIO4	PRT_GPIO08	prt_pmio[4]	prt_vid_en[0]	prt_pwm[4]
PRT_PMIO5	PRT_GPIO09	prt_pmio[5]	prt_vid_en[1]	prt_pwm[5]
PRT_PMIO6	PRT_GPIO10	prt_pmio[6]	vid_clk[0]	prt_pwm[6]
PRT_PMIO7	PRT_GPIO11	prt_pmio[7]	vid_clk[1]	prt_pwm[7]
PRT_PMIO8	PRT_GPIO12	prt_pmio[8]	prt_vid_out[4]	sca_pmio[0]
PRT_PMIO9	PRT_GPIO13	prt_pmio[9]	prt_vid_out[5]	sca_pmio[1]
PRT_PMIO10	PRT_GPIO14	prt_pmio[10]	prt_vid_out[6]	sca_pmio[2]
PRT_PMIO11	PRT_GPIO15	prt_pmio[11]	prt_vid_out[7]	sca_pmio[3]
PRT_PMIO12	PRT_GPIO16	prt_pmio[12]	prt_vid_out[8]	sca_pmio[4]
PRT_PMIO13	PRT_GPIO17	prt_pmio[13]	prt_vid_out[9]	sca_pmio[5]
PRT_PMIO14	PRT_GPIO18	prt_pmio[14]	prt_vid_out[10]	sca_pmio[6]
PRT_PMIO15	PRT_GPIO19	prt_pmio[15]	prt_vid_out[11]	sca_pmio[7]
PRT_PMIO16	PRT_GPIO20	prt_pmio[16]	prt_vid_out[12]	sca_pmio[8]
PRT_PMIO17	PRT_GPIO21	prt_pmio[17]	prt_vid_out[13]	sca_pmio[9]
PRT_PMIO18	PRT_GPIO22	prt_pmio[18]	prt_vid_out[14]	sca_pmio[10]
PRT_PMIO19	PRT_GPIO23	prt_pmio[19]	prt_vid_out[15]	sca_pmio[11]

芯片引脚	GPIO 复用 (默认功能)	芯片主功能	第一复用	第二复用
PRT_PMIO20	PRT_GPIO24	prt_pmio[20]	prt_pwm[0]	sca_pmio[12]
PRT_PMIO21	PRT_GPIO25	prt_pmio[21]	prt_pwm[1]	sca_pmio[13]
PRT_PMIO22	PRT_GPIO26	prt_pmio[22]	prt_pwm[2]	sca_pmio[14]
PRT_PMIO23	PRT_GPIO27	prt_pmio[23]	prt_pwm[3]	sca_pmio[15]
PRT_VIDOUT0	PRT_GPIO28	prt_vid_out[0]	prt_pwm[4]	prt_uart0_rx
PRT_VIDOUT1	PRT_GPIO29	prt_vid_out[1]	prt_pwm[5]	prt_uart0_tx
PRT_VIDOUT2	PRT_GPIO30	prt_vid_out[2]	prt_pwm[6]	prt_uart1_rx
PRT_VIDOUT3	PRT_GPIO31	prt_vid_out[3]	prt_pwm[7]	prt_uart1_tx

表 1-3 扫描系统功能引脚复用关系表

芯片引脚	GPIO 功能 (默认功能)	芯片主功能	第一复用	第二复用
SCA_UART_RX	SCA_GPIO00	sca_uart_rx	sca_pmio[8]	prt_pmio[0]
SCA_UART_TX	SCA_GPIO01	sca_uart_tx	sca_pmio[9]	prt_pmio[1]
SCA_SPI_CLK	SCA_GPIO02	sca_spi_clk	sca_pmio[10]	prt_pmio[2]
SCA_SPI_MISO	SCA_GPIO03	sca_spi_miso	sca_pmio[11]	prt_pmio[3]
SCA_SPI_MOSI	SCA_GPIO04	sca_spi_mosi	sca_pmio[12]	prt_pmio[4]
SCA_SPI_CS	SCA_GPIO05	sca_spi_cs	sca_pmio[13]	prt_pmio[5]
SCA_PMIO0	SCA_GPIO06	sca_pmio[0]	spil_clk	prt_pmio[6]
SCA_PMIO1	SCA_GPIO07	sca_pmio[1]	spil_miso	prt_pmio[7]
SCA_PMIO2	SCA_GPIO08	sca_pmio[2]	spil_mosi	prt_pmio[8]
SCA_PMIO3	SCA_GPIO09	sca_pmio[3]	spil_cs	prt_pmio[9]
SCA_PMIO4	SCA_GPIO10	sca_pmio[4]	sca_spi_clk	prt_pmio[10]
SCA_PMIO5	SCA_GPIO11	sca_pmio[5]	sca_spi_miso	prt_pmio[11]
SCA_PMIO6	SCA_GPIO12	sca_pmio[6]	sca_spi_mosi	prt_pmio[12]
SCA_PMIO7	SCA_GPIO13	sca_pmio[7]	sca_spi_cs	prt_pmio[13]
SCA_AFECLK	SCA_GPIO14	sca_afeclk	sca_pmio[8]	prt_pmio[14]
SCA_AFEPLS	SCA_GPIO15	sca_afepls[0]	sca_pmio[9]	prt_pmio[15]
SCA_AFED0	SCA_GPIO16	sca_afesd[0]	sca_pmio[10]	prt_pmio[16]
SCA_AFED1	SCA_GPIO17	sca_afesd[1]	sca_pmio[11]	prt_pmio[17]
SCA_AFED2	SCA_GPIO18	sca_afesd[2]	sca_pmio[12]	prt_pmio[18]
SCA_AFED3	SCA_GPIO19	sca_afesd[3]	sca_pmio[13]	prt_pmio[19]
SCA_AFED4	SCA_GPIO20	sca_afesd[4]	sca_pmio[14]	prt_pmio[20]
SCA_AFED5	SCA_GPIO21	sca_afesd[5]	sca_pmio[15]	prt_pmio[21]
SCA_AFED6	SCA_GPIO22	sca_afesd[6]	sca_pmio[0]	prt_pmio[22]
SCA_AFED7	SCA_GPIO23	sca_afesd[7]	sca_pmio[1]	prt_pmio[23]
SCA_CISCLK0	SCA_GPIO24	sca_cisclk[0]	sca_pmio[2]	prt_pmio[24]
SCA_CISCLK1	SCA_GPIO25	sca_cisclk[1]	sca_pmio[3]	prt_pmio[25]
SCA_CISCLK2	SCA_GPIO26	sca_cisclk[2]	sca_pmio[4]	prt_pmio[26]
SCA_CISPLS	SCA_GPIO27	sca_cispls[0]	sca_pmio[5]	prt_pmio[27]
SCA_CISPWM	SCA_GPIO28	sca_cispwm	sca_pmio[6]	prt_pmio[28]

注：除芯片启动相关的引脚(SPI0 或 SDIO0/eMMC0 在相应启动模式下对应引脚为主功能)外，以上复用引脚上电默认状态都复用为 GPIO 功能，且均默认为输入状态。

2 时钟结构

龙芯 2P0300 的时钟结构如图2-1 所示，片内主要由一路外部系统时钟输入，最高 100MHz 频率单端时钟（晶振模式）作为系统参考时钟输入，供片内 PLL 使用。芯片内部共有 3 个独立的 PLL，其中每个 PLL 最多可以提供 3 组频率相互依赖的时钟输出。这 3 个 PLL 的用途分别为：

一个 SYS PLL 用于产生 SYS 系统时钟，该时钟为芯片的主要设备时钟，分为三组频率时钟：一路供主 CPU 核、二级 Cache、一级交叉开关以及 IODMA 模块使用；一路供打印、扫描小核使用；一路供 USB、GMAC、BOOT、APB 设备、SDIO、IMAGE、打印/扫描子系统设备模块等设备控制器其中一个时钟源选择使用。

一个 DDR PLL 同时产生 DDR、NETWORK 以及 SYS1 设备的时钟，其中 DDR 时钟源供内部 DDR 控制器使用；NETWORK 时钟源供主系统内部系统互连网络使用；SYS1 设备时钟源供 USB、GMAC、BOOT、APB 设备、SDIO、IMAGE、打印/扫描子系统设备模块等设备控制器另一个时钟源选择使用。

一个 VID PLL 同时产生 Printer 控制器、Scanner 控制器和打印/扫描子系统设备模块及 GMAC 控制器备选的内部时钟。

除了内部的 PLL 之外，对于 USB 采用 PHY 自身产生时钟的模块，也使用外部输入的参考时钟进行参考时钟通路设计。

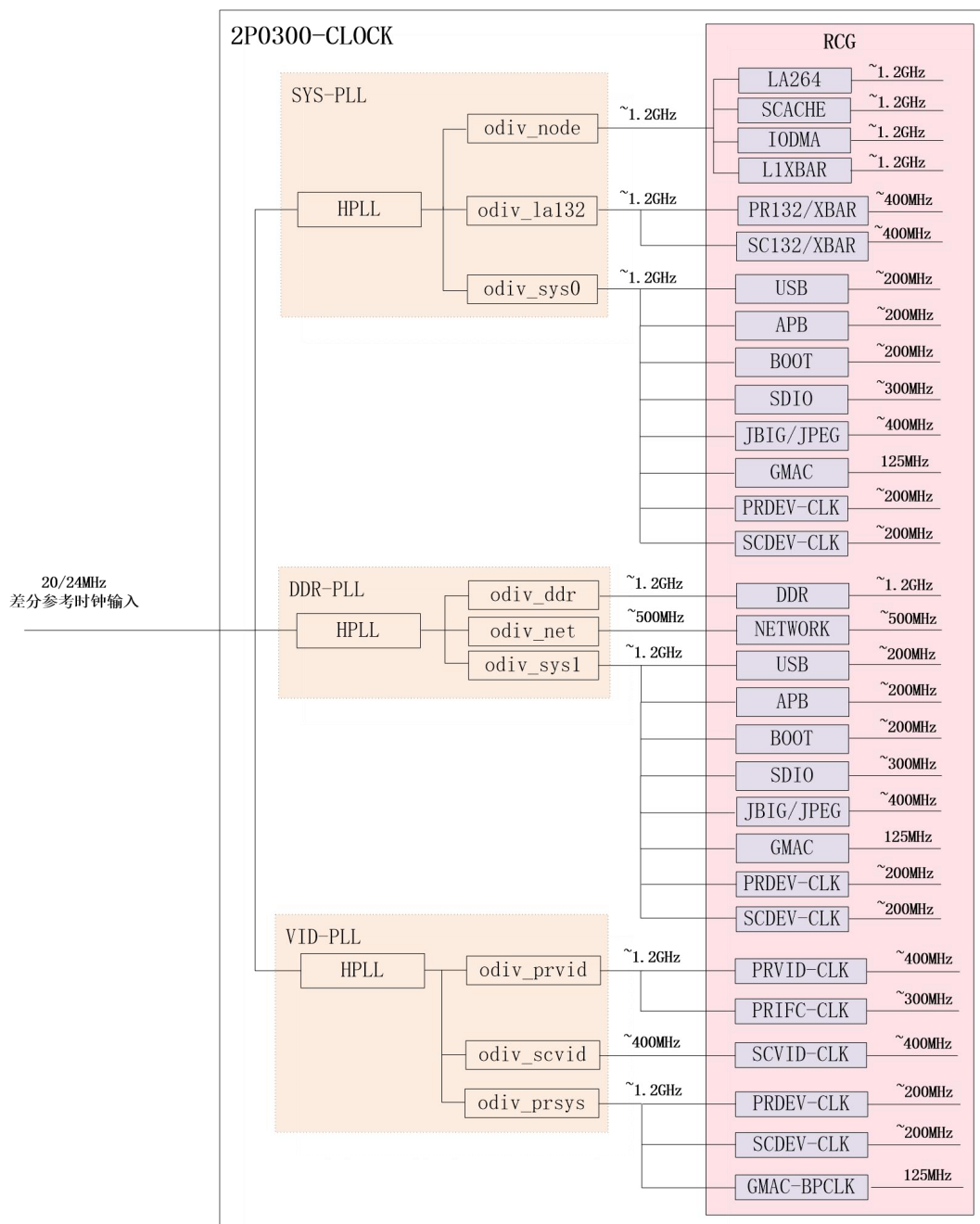


图2-1 时钟结构图

2.1 系统参考时钟

系统输入时钟一路由有源晶振（时钟最高频率 100MHz，由 XTALI 输入）两种方式产生，供内部系统结构使用。

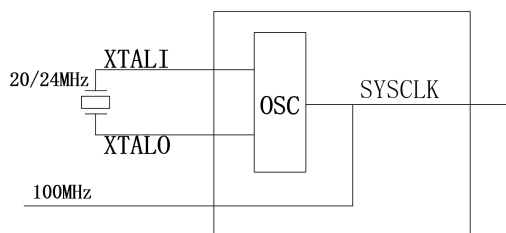


图2-2 系统参考时钟

2.2 SYS-PLL 时钟

SYS PLL 时钟的产生结构图如 2-3 所示：

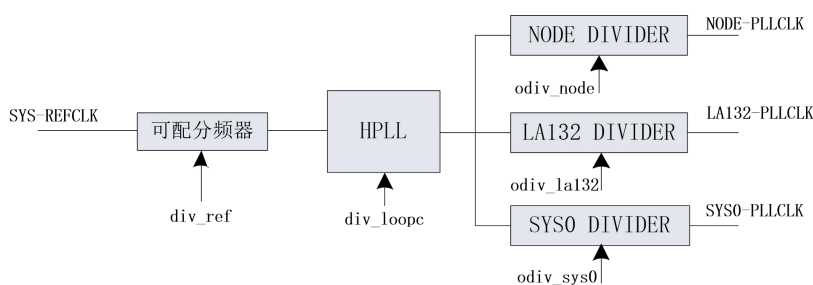


图2-3 SYS PLL 时钟输出

SYS PLL 会输出三个时钟，分别为：

- 1) NODE_PLLCLK 用于 NODE 模块（主 CPU/SCache 等），频率范围：~1.2GHz；
- 2) LA132_PLLCLK 用于打印/扫描 LA132 核，频率范围：~400MHz；
- 3) SYS0_PLLCLK 用于 USB、GMAC、BOOT、APB 设备、SDIO、IMAGE、打印/扫描子系统设备模块等设备控制器其中一路时钟选择源，频率范围：~400MHz。

输出时钟频率的计算方式如下：

- 1) $NODE_PLLCLK = \text{refclk} / \text{div_ref} * \text{div_loopc} / \text{odiv_node}$;
- 2) $LA132_PLLCLK = \text{refclk} / \text{div_ref} * \text{div_loopc} / \text{odiv_la132}$;
- 3) $SYS0_PLLCLK = \text{refclk} / \text{div_ref} * \text{div_loopc} / \text{odiv_sys0}$ 。

NODE_PLLCLK 的工作频率在~1.2GHz 范围，其 PLL 的分频系数以及倍频系数可以任意配置，但是需要保证可配分频器的输出 $\text{refclk} / \text{div_ref}$ 在 20~40MHz 范围内，PLL 倍频值 $\text{refclk} / \text{div_ref} * \text{div_loopc}$ 需要在 1.0GHz~3.2GHz。该限制对其他 2 个内部 PLL 也适用，所以下文不再赘述。

以上三个时钟共用一个 PLL，只是通过设置各自的 divout 值来实现不同的频率输出。所以在调整其中一个模块时钟时，如果对公用 PLL 的倍频系数进行了调整，那么需要注意对其他时钟的影响。

每个输出的时钟还可以经由 freq_scale 模块进行细粒度分频控制。具体分频寄存器配

置请参考 3.4.26~3.4.28 节。

NODE_PLLCLK 时钟，如图 2-4 所示，PLL 输出 NODE_PLLCLK 时钟后通过 NODE_RCG 模块中 freq_scale 模块进行细粒度分频控制，选择 1~8 分频输出。

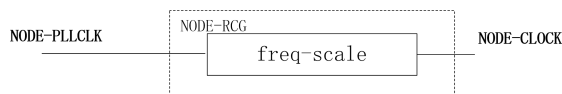


图2-4 NODE_PLLCLK 时钟分频结构

LA132_PLLCLK 时钟，如图 2-5 所示，PLL 输出 LA132_PLLCLK 时钟后通过两组 RCG 模块控制时钟分频输出，用于打印系统小核 la132/prxbar、扫描系统小核 la132/scxbar 时钟。

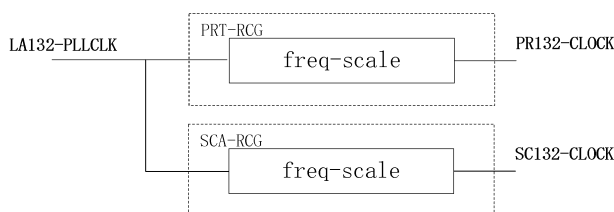


图2-5 LA132_PLLCLK 时钟分频结构

SYS0_PLLCLK 时钟，如图 2-6 所示，PLL 输出 SYS0_PLLCLK 时钟后通过各自 RCG 模块中 clk_divider 模块进行设备分频控制，选择 1~15 分频控制输出时钟，分别供 USB、CMAC、BOOT (CONFBUS/SPI0 控制器)、APB、SDIO、JBIG、JPEG 等设备模块使用。

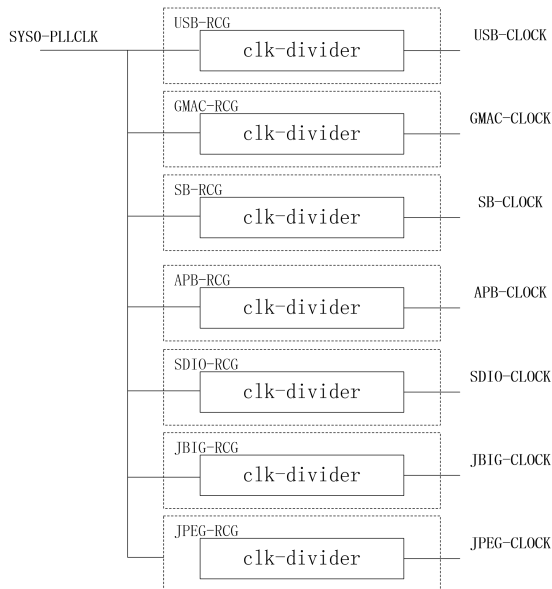


图2-6 SYS0_PLLCLK 设备模块时钟分频结构

2.3 DDR-PLL 时钟

DDR PLL 结构与 SYS PLL 结构基本相同，输出三个时钟，分别为：

- 1) DDR_PLLCLK 用于内存控制器，频率范围：~1.2GHz；

- 2) NETWORK_PLLCLK 用于主系统内部互连网络，频率范围：~500MHz；
- 3) SYS1_PLLCLK 用于 USB、GMAC、BOOT、APB 设备、SDIO、IMAGE、打印/扫描子系统设备模块等设备控制器其中一路时钟选择源，频率范围：~400MHz。

以上三个时钟共用一个 PLL，通过设置各自的 divout 值来实现不同的频率输出。同样在调整其中一个模块时钟时，如果对公用 PLL 的倍频系数进行调整，需要注意对其他时钟的影响。

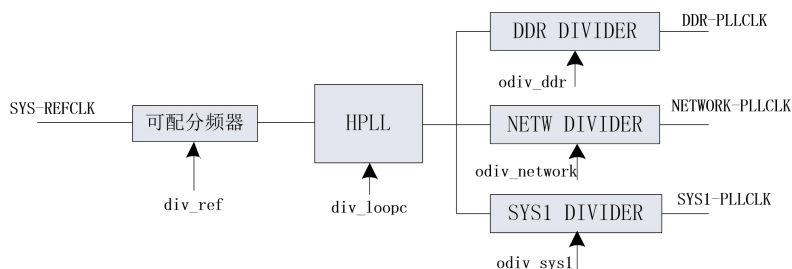


图2-7 DDR PLL 时钟输出

DDR_PLLCLK 时钟，如图 2-8 所示，PLL 输出 DDR-PLLCLK 时钟后通过 DDR_RCG 模块控制最终 DDR 控制器时钟输出。

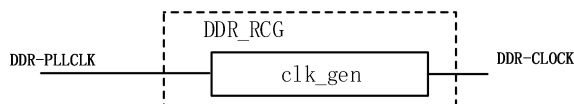


图2-8 DDR_PLLCLK 时钟结构

NETWORK_PLLCLK 互连网络时钟，如图 2-9 所示，PLL 输出互连网络 NET-PLLCLK 时钟后通过 NETWORK_RCG 模块控制最终 NETWORK 时钟输出。

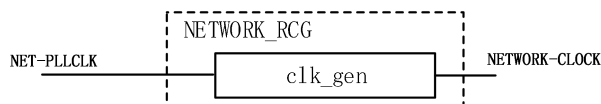


图2-9 NETWORK_PLLCLK 时钟结构

SYS1_PLLCLK 单元时钟，如图 2-10 所示，PLL 输出 SYS1-PLLCLK 时钟后通过各自 RCG 模块中 clk_divider 模块进行设备分频控制，选择 1~15 分频控制输出时钟，分别供 USB、CMAC、BOOT(CONFBUS/SPIO 控制器)、APB、SDIO、JBIG、JPEG 等单元模块使用。

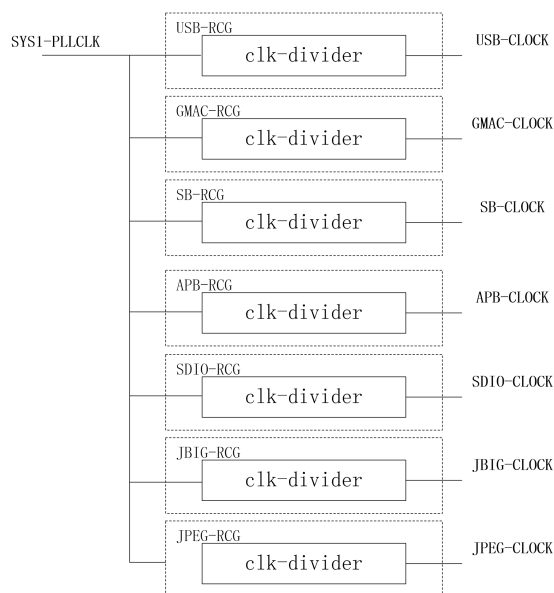


图2-10 SYS1_PLLCLK 单元时钟分频结构

2.4 VID-PLL 时钟

VID PLL 结构与其他 PLL 结构类似，输出三个时钟，分别为：

- 1) PRVID_PLLCLK 用于打印接口控制器，频率范围：~400MHz；
- 2) SCVID_PLLCLK 用于扫描接口控制器，频率范围：~300MHz；
- 3) PRSYS_PLLCLK 用于打印/扫描子系统内部设备模块及 GMAC 控制器备选的时钟，频率范围：~200MHz。

三个时钟共用一个 PLL，可通过设置各自的 divout 值来实现不同的频率输出。同样在调整其中一个模块时钟时，如果对公用 PLL 的倍频系数进行了调整，需要注意对其他时钟的影响。

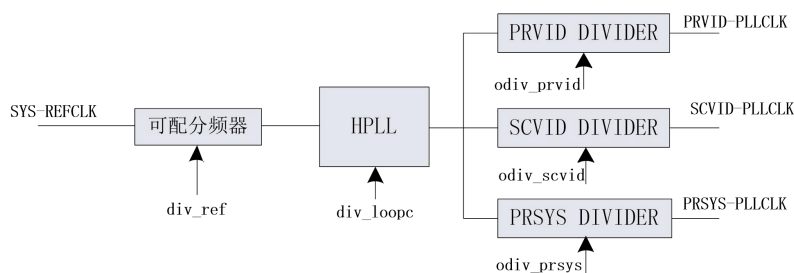


图2-11 VID PLL 时钟输出

PRVID_PLLCLK 时钟，如图 2-12 所示，PLL 输出 PRVID-PLLCLK 时钟后通过各自 RCG 模块中 clk_divider 模块进行细粒度分频控制，选择 1~15 分频控制输出时钟，分别供 PRTVID、PRTIFC 等设备模块使用。

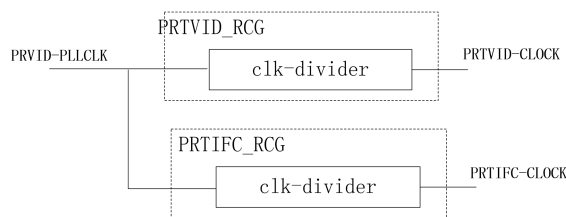


图2-12 PRTVID_PLL 时钟分频结构

SCVID_PLLCLK 时钟，如图 2-13 所示，PLL 输出 SCVID-PLLCLK 时钟后通过 RCG 模块控制最终扫描接口控制器输出时钟。

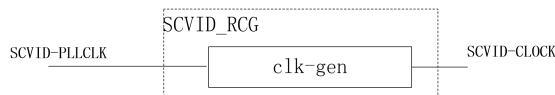


图2-13 SCVID_PLLCLK 时钟结构

PRSYS_PLL 时钟，如图 2-14 所示，PLL 输出 PRSYS-PLLCLK 时钟后通过各自 RCG 模块中 clk_divider 模块进行细粒度分频控制，选择 1~15 分频控制输出时钟，供 PRTDEV、SCADEV 等设备模块使用。

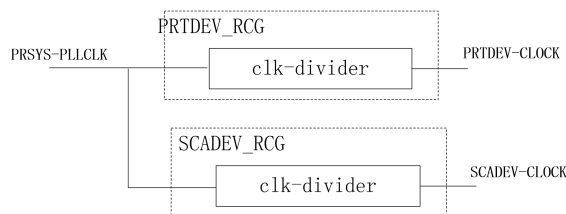


图2-14 PRTSYS 时钟结构

2.5 内部 PLL 配置方法

以上 3 个内部 PLL 提供硬件配置和软件配置两种配置方法。这两种配置方法通过 SYS_CLKSEL[1:0]的设置来区分。

2.5.1 硬件配置

具体如下表所示。

表 2-1 PLL 硬件配置

SYS_CLKSEL	00(硬件低频)	01 (硬件高频)	10	11
NODE	800M	1.2G	软件配置	硬件 bypass 所有 PLL，所有时钟频率与参考时钟相同
LA132	200M	400M		
SYS0	100M	200M		
DDR	800M	1.2G		
NETWORK	266M	400M		
SYS1	100M	150M		
PRVID	100M	300M		
SCVID	100M	300M		
PRSYS	100M	200M		

2.5.2 软件配置

当 SYS_CLKSEL 设置为 2' b10 时表示 PLL 频率通过软件配置。这种配置下，默认对应的时钟频率为外部参考时钟频率，即所有 PLL 输出都是 SYS_REFCLK，需要在处理器启动过程中对时钟进行软件配置。各个时钟设置的过程应该按照以下方式：

1. 将对应的 PLL 的 PD 信号设置为 1；
2. 设置寄存器除了 sel_pll_*及 soft_set_pll 之外的其它寄存器，即这两个寄存器在设置的过程中写为 0；
3. 将对应的 PLL 的 PD 信号设置为 0；
4. 其他寄存器值不变，将 soft_set_pll 设置为 1；
5. 等待寄存器中的锁定信号 locked_*为 1；
6. 设置 sel_pll_*为 1，此时对应的时钟频率将切换为软件设置的频率。

另外，芯片内部配有设备时钟分频配置参数 freqscale，可供软件对部分设备时钟进行再次分频选择。

具体的配置寄存器说明请参考第五章。

2.6 USB 参考时钟

USB PHY 为 1 个独立的单端口 PHY，参考时钟输入提供以下 2 种方式供选择：

使用 1 个外部 20/24MHz 晶振时钟输入；

使用内部生成 20/24MHz（由 SYSCLK 系统时钟产生）的参考时钟。

2.7 时钟信号说明

表 2-2 统一介绍了龙芯 2P0300 的所有时钟信号。

表 2-2 2P0300 时钟信号说明

信号名称	推荐工作频率(MHz)	类型	描述
SYS_XTALI SYS_XTALO	100	I	外接系统参考时钟（可支持直通晶振模式，由 XTALI 输入，最高频率 100MHz）
TESTCLK	100	I	外接系统测试时钟
GMAC_RX_CLK	125	I	GMAC 接收输入时钟
GMAC_TX_CLK_0	125	O	GMAC 发送输出时钟
SPI0_CLK	~100	O	SPI0 输出时钟
MAIN_SPI_CLK	~100	I/O	MAIN_SPI 输出（主）/输入（从）时钟
SCA_SPI_CLK	~100	I/O	SCA_SPI 输出（主）/输入（从）时钟
SDIO0/1_CLK	~200	O	SDIO0/1 输出时钟
JTAG_TCK	~33	I	JTAG 时钟

信号名称		推荐工作 频率 (MHz)	类型	描述
USB_REFCLK		24/20	I	USB 参考时钟输入 如果使用内部参考时钟，则可悬空。
DDR_CKp DDR_CKn		~1200	DIFF OUT	DDR3/4 SDRAM 差分时钟输出
内部时钟	NODE_CLOCK	~1250	-	主系统 NODE 模块时钟，供 LA264、SCACHE、IODMA、L1-XBAR 模块使用
	STABLE_CLOCK	100	-	LA264 核内 STABLE 计数时钟（同 SYSCLK 系统时钟）
	PRI32_CLOCK	~400	-	打印系统 LA132 核时钟，供 LA132 处理器核内部使用
	SC132_CLOCK	~400	-	扫描系统 LA132 核时钟，供 LA132 处理器核内部使用
	USB_CLOCK0	~200	-	USB 模块时钟源 0
	GMAC_CLOCK0	125	-	GMAC 模块时钟源 0
	BOOT_CLOCK0	~200	-	BOOT 设备时钟源 0 (BOOT/SPI/CONFBUS 等)
	APB_CLOCK0	~200	-	主系统 APB 设备时钟源 0
	JBIG_CLOCK0	~400	-	主系统 JBIG 设备时钟源 0
	JPEG_CLOCK0	~400	-	主系统 JPEG 设备时钟源 0
	SDIO_CLOCK0	~300	-	SDIO/eMMC 模块时钟源 0
	DDR_CLOCK	~1.2G	-	DDR3/4 控制器时钟
	NETWORK_CLOCK	~500	-	供 NETWORK 互联结构使用
	USB_CLOCK1	~200	-	USB 模块时钟源 1
	GMAC_CLOCK1	125	-	GMAC 模块时钟源 1
	BOOT_CLOCK1	~200	-	BOOT 设备时钟源 1 (BOOT/SPI/CONFBUS 等)
	APB_CLOCK1	~200	-	主系统 APB 设备时钟源 1
	JBIG_CLOCK1	~400	-	主系统 JBIG 设备时钟源 1
	JPEG_CLOCK1	~400	-	主系统 JPEG 设备时钟源 1
	SDIO_CLOCK1	~300	-	SDIO/eMMC 模块时钟源 1
	PRVID_CLOCK	~400	-	供 PRINTeR 打印数据处理单元使用
	PRIFC_CLOCK	~300	-	供 PRINTeR 打印模块 AXI 总线使用
	SCVID_CLOCK	~300	-	供扫描接口模块使用
	PRTDEV_CLOCK	~200	-	供打印系统中各外围设备使用
	SCADEV_CLOCK	~200	-	供扫描系统中各外围设备使用
	GMACBP_CLOCK	125	-	GMAC 备份时钟

3 芯片配置与控制

3.1 芯片初始化信号

龙芯 2P0300 初始化信号：复用芯片功能引脚，通过在系统复位期间采样外部上下拉的值得到配置信息。部分配置信息编码成 bootcfg，供软件判定上电状态。

表 3-1 初始化配置信号

信号名称	类型	描述
SYS_PM0IO[0]	I	启动选择输入 0=SPI0 1=SDIO0/eMMC0
SYS_PM0IO[2:1]	I	PLL 时钟配置输入 00=低频模式 01=高频模式 10=软件模式 11=bypass 模式
SYS_PM0IO[3]	I	SDIO0 模式配置输入 0=SDIO 模式 1=eMMC 模式
SYS_PM0IO[4]	I	SDIO1 模式配置输入 0=SDIO 模式 1=eMMC 模式
SYS_PM0IO[5]	I	USB 参考时钟模式输入 0=内部参考时钟输入 1=外部参考时钟输入
SYS_PM0IO[6]	I	eMMC0 引脚（4 线模式涉及 IO：CLK/CMD/D0~D3）电平类型 0=3.3V IO（对应 IO_3V3_1V8 供电需选择 3.3V 供电） 1=1.8V IO（对应 IO_3V3_1V8 供电需选择 1.8V 供电）
SYS_PM0IO[7]	I	eMMC 引脚（8 线模式涉及 IO：D4~D7）电平类型 0=3.3V IO（对应 IO_3V3_1V8 供电需选择 3.3V 供电） 1=1.8V IO（对应 IO_3V3_1V8 供电需选择 1.8V 供电）
SYS_PM0IO[8]	I	QSPI 0 引脚（涉及 IO：CLK/CS/D0~D3）电平类型 0=3.3V IO（对应 IO_3V3_1V8 供电需选择 3.3V 供电） 1=1.8V IO（对应 IO_3V3_1V8 供电需选择 1.8V 供电）

3.2 地址空间分配

龙芯 2P0300 的地址空间根据各个系统应用不同，分为主系统、打印系统、扫描系统和 DMA 四个视角地址空间：所有主系统处理器核可访问的设备编址在主系统地址空间上，谓之主系统视角；打印系统、扫描系统类似，可通过 DMA 直接访问系统内存的主设备所见到

的空间为 DMA 视角。此外，各个系统之间存在可以共享访问设备地址空间，不同系统访问同一设备，其设备地址均相同，可通过不同系统之间共享设备操作。表 3-2~表 3-4 分别给出了这四个视角的具体定义。表格中未包含的地址空间均为系统保留，软件错误地访问保留空间将导致不可预知的后果。

表 3-2 地址空间分配之主系统视角

地址空间 (unmapped)	大小	功能	拓扑
0x0000_0000 - 0x0fff_ffff	256MB	L-DDR	DDR 低地址空间
0x1000_0000 - 0x13ff_ffff	64MB	SPI0 MEM	MEM
0x1400_0000 - 0x1401_ffff	128KB	SYS-CONFBUS	CONFBUS (主/打印/扫描系统可共享)
0x1402_0000 - 0x141f_ffff	2MB-128KB	AXI-DEVs	DEVs (USB/GMAC1/IMGF)
0x1420_0000 - 0x142f_ffff	1MB	DEVs (APB)	MAIN-DEVs (主/打印/扫描系统可共享)
		Reserved	
0x1500_0000 - 0x1501_ffff	128KB	PR-SRAM	PR-SRAM (主/打印/扫描系统可共享)
0x1510_0000 - 0x151f_ffff	1MB	PR-DEVs	PR-DEVs (主/打印/扫描系统可共享)
0x1520_0000 - 0x152f_ffff	1MB	SC-DEVs	SC-DEVs (主/打印/扫描系统可共享)
		Reserved	
0x1c00_0000 - 0x1c0f_ffff	1MB	BOOT	Boot (主/打印/扫描系统可共享)
		Reserved	
0x8000_0000 - 0xffff_ffff	2GB	H-DDR	DDR 高地址空间
地址空间 (AXI-DEVs) [19:0] Base 0x1400_0000	大小 1MB	功能	拓扑
0x0_0000	128KB	CONF/INT	CLK/INT/IOMUX...
0x2_0000	64KB	GMAC	(可共享)
0x3_0000	64KB	USB	(可共享)
0x4_0000	64KB	JBIG	包含两路 JBIG，每路分配 16KB 空间，仅主核可访问
0x5_0000	64KB	JPEG	仅主核可访问
0x10_0000	256KB	OTG	(可共享)
地址空间 (APB) [19:0] Base 0x1420_0000	大小 1MB	功能	拓扑
0x0_0000	4KB	UARTx2	(主/打印/扫描系统可共享)，每路平均分配 2KB 空间
0x0_1000	4KB	GPIOx48	(主/打印/扫描系统可共享)，分为两种配置方式
0x0_2000	4KB	I ² Cx1	(主/打印/扫描系统可共享)
0x0_3000	4KB	SPIx1	(主/打印/扫描系统可共享)
0x0_4000	4KB	HPETx2	(主/打印/扫描系统可共享)，每路平均分配 2KB 空间
0x0_5000	4KB	WDTx1	(主/打印/扫描系统可共享)
0x0_6000	4KB	PMIOx1	(主/打印/扫描系统可共享)
0x0_7000	4KB	PWMx8	(主/打印/扫描系统可共享)，有效空间 128B，每路 16B 空间
0x0_8000	4KB	APB-DMA	(主/打印/扫描系统可共享)
0x0_9000-0x0_ffff	28KB	Reserved	
0x1_0000	32KB	SDIO0	(主/打印/扫描系统可共享)
0x1_8000	32KB	SDIO1	(主/打印/扫描系统可共享)

表 3-3 地址空间分配之打印系统视角

地址空间 (unmapped)	大小	功能	拓扑
		Reserved	
0x1400_0000 - 0x1401_ffff	128KB	CONFBUS (AXI-CONF)	MAIN-CONFBUS (主/打印/扫描系统可共享)

			享)
0x1402_0000 - 0x141f_ffff	2MB-128KB	AXI-DEVs (Main-sys)	Main-DEVs (除 IMG 外可共享)
0x1420_0000 - 0x142f_ffff	1MB	APB-DEVs (Main-sys)	MAIN-APB-DEVs (主/打印/扫描系统可共享)
		Reserved	
0x1500_0000 - 0x150f_ffff	1MB	SRAM (128K)	PR-BOOT/DATA
0x1510_0000 - 0x151f_ffff	1MB	DEVs (APB)	PR-DEVs (主/打印/扫描系统可共享)
0x1520_0000 - 0x152f_ffff	1MB	SC-DEVs	SC-DEVs (主/打印/扫描系统可共享)
		Reserved	
0x1c00_0000 - 0x1c0f_ffff	1MB	BOOT	SYS-BOOT (可共享)
		Reserved	
0x8000_0000 - 0xffff_ffff	2GB	DDR	DDR/CCIO
地址空间 (AXI) [19:0] Base 0x1500_0000	大小 1MB	功能	拓扑
0x0_0000 - 0x1_ffff	128KB	SRAM (128KB)	PR-BOOT/DATA (共享 PR-SRAM 存储)
地址空间 (APB) [19:0] Base 0x1510_0000	大小 1MB	功能	拓扑
0x0_0000	4KB	UARTx2	(主/打印/扫描系统可共享)，每路平均分配 2KB 空间
0x0_1000	4KB	GPIOx32	(主/打印/扫描系统可共享)
0x0_2000	4KB	I ² Cx1	(主/打印/扫描系统可共享)
0x0_3000	4KB	CONF	MAILBOX/CLK/INT/CHIP/MUX... (主/打印/扫描系统可共享)
0x0_4000	4KB	Reserved	
0x0_5000	4KB	ADC	(主/打印/扫描系统可共享)
0x0_6000	4KB	Reserved	-
0x0_7000	4KB	PWMx8	(主/打印/扫描系统可共享) 有效空间 8*16B，每路 16B 空间
0x0_8000	4KB	PMIOx1	(主/打印/扫描系统可共享)
0x0_9000	4KB	Reserved	
0x0_a000	4KB	APB-DMA	(主/打印/扫描系统可共享)
0x0_b000	4KB	HPETx2	(主/打印/扫描系统可共享)
0x0_c000	16KB	PRT-VIDEO	(主/打印/扫描系统可共享)

表 3-4 地址空间分配之扫描系统视角

地址空间 (unmapped)	大小	功能	拓扑
		Reserved	
0x1400_0000 - 0x1401_ffff	128KB	CONFBUS (AXI-CONF)	MAIN-CONFBUS (主/打印/扫描系统可共享)
		Reserved	
0x1420_0000 - 0x142f_ffff	1MB	DEVs (APB)	MAIN-APB-DEVs (主/打印/扫描系统可共享)
		Reserved	
0x1500_0000 - 0x1501_ffff	128KB	SRAM (128K)	PR-BOOT/DATA (主/打印/扫描系统可共享)
0x1510_0000 - 0x151f_ffff	1MB	DEVs (APB)	PR-DEVs (主/打印/扫描系统可共享)
0x1520_0000 - 0x152f_ffff	1MB	SC-DEVs	SC-DEVs (主/打印/扫描系统可共享)
		Reserved	
0x1c00_0000 - 0x1c0f_ffff	1MB	BOOT	SYS-BOOT (主/扫描系统可共享)
		Reserved	
0x8000_0000 - 0xffff_ffff	2GB	DDR	DDR/CCIO
地址空间 (APB) [19:0]	大小	功能	拓扑

Base 0x1520_0000	1MB		
0x0_0000	4KB	UART×1	(主/打印/扫描系统可共享) 每路平均分配 2KB 空间
0x0_1000	4KB	GPIOx29	(主/打印/扫描系统可共享)
0x0_2000	4KB	CONF	MAILBOX/INT... (设备可共享)
0x0_3000	4KB	SPIx1	(主/打印/扫描系统可共享)
0x0_4000	4KB	PMIO×1	(主/打印/扫描系统可共享)
0x0_5000	4KB	HPETx2	(主/打印/扫描系统可共享)
0x0_6000	4KB	SCA-IFC	(主/打印/扫描系统可共享)

注：共享资源说明：

(1) 主系统-设备可共享：

	Confbus (AXI)：主系统、打印系统、扫描系统；
	BOOT (AXI)：主系统、打印系统、扫描系统；
	DEVs (APB)：主系统、打印系统、扫描系统；
	OTG/USB/GMAC (AXI-DEVs)：主系统、打印系统；

(2) 打印系统设备共享：

	PRT 设备 (SRAM/DEVs) 可共享：主系统、打印系统、扫描系统；
---	---------------------------------------

(3) 扫描系统设备共享：

	SCAN 设备 (SC-DEVs) 可共享：主系统、打印系统、扫描系统。
---	--------------------------------------

龙芯 2P0300 内部可发起 DMA 的主设备包括 GMAC、USB、IMAGE、SDIO/eMMC 及其他支持 DMA 的 APB 设备等，主设备的请求可以访问路由到表 3-5 中的所有空间。

表 3-5 地址空间分配之 DMA 视角

地址空间 (mapped)	大小	功能	说明
0x0000_0000 - 0x7fff_ffff	2G	DDR	同时设有 CACHE 访问配置位，可配置选择访问 DDR 或 CCIO，默认按照表中地址进行分配。
0x8000_0000 - 0xffff_ffff	2G	CCIO	
0x0000_0000 - 0x0001_ffff	128KB	SRAM	片内 RAM 共享存储，其中 GMAC/OTG/USB 可备选数据存储。

3.3 时钟与复位控制

3.3.1 时钟配置概要

龙芯 2P0300 的片上时钟有硬件、软件两种配置模式。硬件配置模式下时钟生成完全不需要软件参与，但频率选择非常有限。软件配置模式下所有时钟相关的参数都可以改变，非常灵活，但在操作时需谨慎。本小节主要描述软件配置的流程。

在系统复位结束后，所有 PLL 输出均被旁路为参考时钟，分频器设置初始化为默认分频数，整个系统以最低速度运行。这时引导程序应当对片上的每个 PLL 进行以下操作：

1. 将对应的 PLL 的 PD 信号设置为 1；
2. 设置寄存器除了 sel_pll_* 及 soft_set_pll 之外的其它寄存器，即这两个寄存器

在设置的过程中写为 0；

3. 将对应的 PLL 的 PD 信号设置为 0；
4. 其他寄存器值不变，将 soft_set_pll 设置为 1；
5. 等待寄存器中的锁定信号 locked_* 为 1；
6. 设置 sel_pll_* 为 1，此时对应的时钟频率将切换为软件设置的频率。

如果后期有需要修改 PLL 参数，则要先切换时钟为参考时钟，然后按上述步骤再配一遍。

3.3.2 展频 PLL 配置

展频 PLL 概念性结构如图3-1 所示，输入时钟 Fin 经过输入分频器得到 Fref 送到倍频器，倍频器送出 Fvco，然后在输出前除以一个分频系数。展频 PLL 还带有一个展频控制器，能够用三角波对输出时钟的频率进行调制。

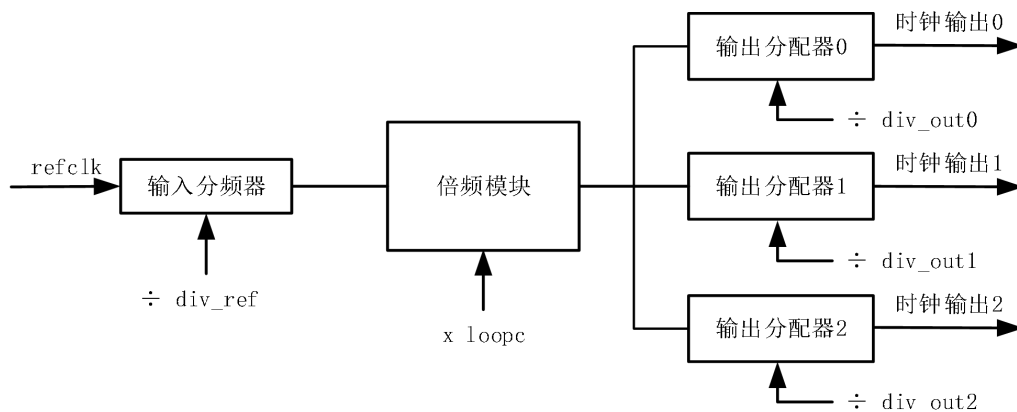


图3-1 展频 PLL 概念性结构

输出时钟频率的计算方式如下： $\text{clock_out} = \text{refclk} / \text{div_ref} * \text{loopc} / \text{divoutN}$ ；

其中，2P0300 的 refclk 为 100MHz（晶振模式），此外需要保证输入分频器的输出（ $\text{refclk} / \text{div_ref}$ ）在 20 ~ 40MHz 的范围内，倍频模块倍频后的频率（ $\text{refclk} / \text{div_ref} * \text{loopc}$ ）在 1.0GHz ~ 3.2GHz 的范围内。

PLL 相关的配置信号及说明见表 3-6。

表 3-6 PLL 相关配置信号说明表

信号	位数	方向	说明
pll_div_out0	7	R/W	PLL 输出时钟 0 分频数
pll_div_out1	7	R/W	PLL 输出时钟 1 分频数
pll_div_out2	7	R/W	PLL 输出时钟 2 分频数
pll_loopc	9	R/W	PLL 倍频乘数
pll_div_ref	7	R/W	PLL 输入分频数
pll_locked	1	RO	PLL 锁定
sel_pll_out0	1	R/W	选择 PLL 输出时钟 0
sel_pll_out1	1	R/W	选择 PLL 输出时钟 1

信号	位数	方向	说明
sel_pll_out2	1	R/W	选择 PLL 输出时钟 2
set_pll_param	1	R/W	设置 PLL 配置参数
pll_bypass	1	R/W	PLL 内部 bypass
pll_pd	1	R/W	PLL powerdown

3.3.3 复位控制

龙芯 2P0300 内部有多种复位机制，用户可根据需要进行选择：

- 模块级复位：只针对一个模块发起，如 USB、GMAC 等。一般只在配置时钟完成，还未开始工作前进行。在工作后复位有使系统死锁的危险。
- 系统热复位：除芯片配置寄存器和时钟配置寄存器保持不变外，全系统复位。
- 系统冷复位：由看门狗或者复位模块发起，详见相关章节。

3.4 芯片配置寄存器

龙芯 2P0300 有大量的配置寄存器，根据各个系统配置应用不同，分为主系统、打印系统、扫描系统配置空间,其余多数分布于各个功能模块中，本节介绍各个系统芯片级的配置寄存器。

表 3-7 主系统芯片配置寄存器列表

地址	名称	描述
0x14000100	MAIN_CHIP_CTRL0	主系统通用配置寄存器 0
0x14000104	MAIN_CHIP_CTRL1	主系统通用配置寄存器 1
0x14000108	MAIN_CHIP_CTRL2	主系统通用配置寄存器 2
0x1400010c	MAIN_CHIP_CTRL3	主系统通用配置寄存器 3
0x14000110	MAIN_CHIP_CTRL4	主系统通用配置寄存器 4
0x14000114	MAIN_CHIP_CTRL5	主系统通用配置寄存器 5
0x14000118	MAIN_CHIP_CTRL6	主系统通用配置寄存器 6
0x1400011c	MAIN_CHIP_CTRL7	主系统通用配置寄存器 7
0x14000120	MAIN_CHIP_CTRL8	主系统通用配置寄存器 8
0x14000124	MAIN_CHIP_CTRL9	主系统通用配置寄存器 9
0x14000140	MAIN_CHIP_SAMP0	主系统采样参数寄存器 0
0x14000144	MAIN_CHIP_SAMP1	主系统采样参数寄存器 1
0x14000150	MAIN_CHIP_HPT0	主系统高精度计数器低 32 位
0x14000154	MAIN_CHIP_HPT1	主系统高精度计数器高 32 位
0x14000490	MAIN_GPIO_CFG0	主 GPIO00~15 复用配置寄存器
0x14000494	MAIN_GPIO_CFG1	主 GPIO16~31 复用配置寄存器
0x14000498	MAIN_GPIO_CFG2	主 GPIO32~47 复用配置寄存器
0x14000500	USB_PHY0	USB 的 PHY 配置寄存器 0
0x14000504	USB_PHY1	USB 的 PHY 配置寄存器 1

地址	名称	描述
0x14001040	CORE_INTISR0	路由给 CORE的低 32位中断状态
0x14001044	INTISR0	低 32位中断状态寄存器
0x14001048	CORE_INTISR1	路由给 CORE的高 32位中断状态
0x1400104c	INTISR1	高 32位中断状态寄存器
0x14001148	EXTIOI_ACK	扩展中断设备反馈寄存器
0x14001400	ENTRY0_0	8 位中断路由寄存器[0--7]
0x14001408	ENTRY8_0	8 位中断路由寄存器[8--15]
0x14001410	ENTRY16_0	8 位中断路由寄存器[16--23]
0x14001418	ENTRY24_0	8 位中断路由寄存器[24--31]
0x14001420	INTISR_0	低 32位中断状态寄存器
0x14001424	INTIEN_0	低 32位中断使能状态寄存器
0x14001428	INTSET_0	低 32位中断设置使能寄存器
0x1400142c	INTCLR_0	低 32 位中断清除寄存器，清除使能寄存器和脉冲触发的中断
0x14001430	INTPOL_0	低 32 位极性设置寄存器(电平中断)
0x14001434	INTEDGE_0	低 32 位触发方式寄存器（1：脉冲触发；0：电平触发）
0x14001440	ENTRY0_1	8 位中断路由寄存器[32--39]
0x14001448	ENTRY8_1	8 位中断路由寄存器[40--47]
0x14001450	ENTRY16_1	8 位中断路由寄存器[48--55]
0x14001458	ENTRY24_1	8 位中断路由寄存器[56--63]
0x14001460	INTISR_1	高 32 位中断状态寄存器
0x14001464	INTIEN_1	高 32 位中断使能状态寄存器
0x14001468	INTSET_1	高 32 位设置使能寄存器
0x1400146c	INTCLR_1	高 32 位中断清除寄存器，清除使能寄存器和脉冲触发的中断
0x14001470	INTPOL_1	高 32 位极性设置寄存器(电平中断)
0x14001474	INTEDGE_1	高 32 位触发方式寄存器（1：脉冲触发；0：电平触发）
0x140014c0	EXTIOI_MAP	扩展中断设备路由寄存器
0x14001600	EXTIOI_IEN0	扩展中断设备使能寄存器 0
0x14001604	EXTIOI_IEN1	扩展中断设备使能寄存器 1
0x14001608	EXTIOI_IEN2	扩展中断设备使能寄存器 2
0x1400160c	EXTIOI_IEN3	扩展中断设备使能寄存器 3
0x14001640	EXTIOI_POLO	扩展中断电平配置寄存器 0
0x14001644	EXTIOI_POL1	扩展中断电平配置寄存器 1
0x14001648	EXTIOI_POL2	扩展中断电平配置寄存器 2
0x1400164c	EXTIOI_POL3	扩展中断电平配置寄存器 3
0x14001700	EXTIOI_ISR0	扩展中断状态寄存器 0
0x14001704	EXTIOI_ISR1	扩展中断状态寄存器 1

地址	名称	描述
0x14001708	EXTIOI_ISR2	扩展中断状态寄存器 2
0x1400170c	EXTIOI_ISR3	扩展中断状态寄存器 3
0x14001800	EXTIOI_CORE_ISR0	路由至 CORE 扩展中断状态寄存器 0
0x14001804	EXTIOI_CORE_ISR1	路由至 CORE 扩展中断状态寄存器 1
0x14001808	EXTIOI_CORE_ISR2	路由至 CORE 扩展中断状态寄存器 2
0x1400180c	EXTIOI_CORE_ISR3	路由至 CORE 扩展中断状态寄存器 3
0x14003fc0	CHIP_CHIPID0	芯片标识号 0
0x14003fc8	CHIP_CHIPID1	芯片标识号 1
0x14003fd0	CHIP_CHIPID2	芯片标识号 2
0x14003fd8	CHIP_CHIPID3	芯片标识号 3
0x14003fe0	CHIP_CHIPID4	芯片标识号 4
0x14003fe8	CHIP_CHIPID5	芯片标识号 5
0x14003ff0	CHIP_CHIPID6	芯片标识号 6
0x14003ff8	CHIP_CHIPID7	芯片标识号 7

表 3-8 打印系统芯片配置寄存器列表

地址	名称	描述
0x15103000	PLL_SYS_0	SYS-PLL 低 32 位配置
0x15103004	PLL_SYS_1	SYS-PLL 高 32 位配置
0x15103008	PLL_DDR_0	内存控制器 PLL 低 32 位配置
0x1510300c	PLL_DDR_1	内存控制器 PLL 高 32 位配置
0x15103010	PLL_VID_0	VID-PLL 低 32 位配置
0x15103014	PLL_VID_1	VID-PLL 高 32 位配置
0x15103020	FREQSCALE0	设备时钟分频配置 0
0x15103024	FREQSCALE1	设备时钟分频配置 1
0x15103028	FREQSCALE2	设备时钟分频配置 2
0x1510302c	PLLCLK_EN	设备 PLL 时钟使能配置
0x15103030	PRDEV_CLK_CTRL	打印设备时钟门控配置
0x15103034	PRDEV_RST_CTRL	打印设备复位配置
0x15103038	NODE_DVFS_CTRL	NODE DFS 寄存器描述
0x1510303c	LA132_DVFS_CTRL	LA132 DFS 寄存器描述
0x15103040	PRT_GPIO_CFG0	打印 GPIO00~15 复用配置
0x15103044	PRT_GPIO_CFG1	打印 GPIO16~31 复用配置
0x15103050	PLLCLK_MODE	设备时钟输出模式配置寄存器
0x15103060	PRT_CHIPID0	打印系统识别号 0
0x15103064	PRT_CHIPID1	打印系统识别号 1
0x15103068	PRT_CHIPID2	打印系统识别号 2
0x1510306c	PRT_CHIPID3	打印系统识别号 3
0x15103070	PRT_CHIP_CTRL0	打印系统通用配置 0

地址	名称	描述
0x15103074	PRT_CHIP_CTRL1	打印系统通用配置 1
0x15103078	PRT_CHIP_SAMP	打印系统采样状态
0x1510307c	PRT_CHIP_HPT	打印系统计数寄存器
0x151030e0	PRT_DEVS_BASE	打印设备基址路由配置寄存器
0x15103100	PRT_BOOT_ENTRY	打印启动入口配置
0x15103160	PRT_CHIPID4	打印系统识别号 4
0x15103164	PRT_CHIPID5	打印系统识别号 5
0x15103168	PRT_CHIPID6	打印系统识别号 6
0x1510316c	PRT_CHIPID7	打印系统识别号 7
0x15103400	PRT_INT_ISR0	打印设备中断状态 0
0x15103404	PRT_INT_IEN0	打印设备中断使能配置 0
0x1510340c	PRT_INT_CLR0	打印设备中断清除配置 0
0x15103410	PRT_INT_POLO	打印设备中断电平配置 0
0x15103414	PRT_INT_EDGE0	打印设备中断边沿配置 0
0x15103418	PRT_INT_DUAL0	打印设备中断双沿配置 0
0x15103420	PRT_INT_ISR1	打印设备中断状态 1
0x15103424	PRT_INT_IEN1	打印设备中断使能配置 1
0x1510342c	PRT_INT_CLR1	打印设备中断清除配置 1
0x15103430	PRT_INT_POL1	打印设备中断电平配置 1
0x15103434	PRT_INT_EDGE1	打印设备中断边沿配置 1
0x15103438	PRT_INT_DUAL1	打印设备中断双沿配置 1
0x15103440	PRT_INT_ISR2	打印设备中断状态 2
0x15103444	PRT_INT_IEN2	打印设备中断使能配置 2
0x1510344c	PRT_INT_CLR2	打印设备中断清除配置 2
0x15103450	PRT_INT_POL2	打印设备中断电平配置 2
0x15103454	PRT_INT_EDGE2	打印设备中断边沿配置 2
0x15103458	PRT_INT_DUAL2	打印设备中断双沿配置 2
0x15103460	PRT_INT_MAP0	打印设备中断路由配置 0
0x15103464	PRT_INT_MAP1	打印设备中断路由配置 1
0x15103468	PRT_INT_MAP2	打印设备中断路由配置 2
0x1510346c	PRT_INT_MAP3	打印设备中断路由配置 3
0x15103470	PRT_INT_MAP4	打印设备中断路由配置 4
0x15103474	PRT_INT_MAP5	打印设备中断路由配置 5
0x15103800	MLB0_IPI_Status	0 号 Mailbox 的 IPI_Status 寄存器
0x15103804	MLB0_IPI_Enable	0 号 Mailbox 的 IPI_Enalbe 寄存器
0x15103808	MLB0_IPI_Set	0 号 Mailbox 的 IPI_Set 寄存器
0x1510380c	MLB0_IPI_Clear	0 号 Mailbox 的 IPI_Clear 寄存器
0x15103810	MLB0_message0	0 号 Mailbox 的 Message0 寄存器
0x15103814	MLB0_message1	0 号 Mailbox 的 Message1 寄存器

地址	名称	描述
0x15103818	MLB0_message2	0 号 Mailbox 的 Message2 寄存器
0x1510381c	MLB0_message3	0 号 Mailbox 的 Message3 寄存器
0x15103820	MLB0_message4	0 号 Mailbox 的 Message4 寄存器
0x15103824	MLB0_message5	0 号 Mailbox 的 Message5 寄存器
0x15103828	MLB0_message6	0 号 Mailbox 的 Message6 寄存器
0x1510382c	MLB0_message7	0 号 Mailbox 的 Message7 寄存器
0x15103830	MLB1_IPI_Status	1 号 Mailbox 的 IPI_Status 寄存器
0x15103834	MLB1_IPI_Enable	1 号 Mailbox 的 IPI_Enalbe 寄存器
0x15103838	MLB1_IPI_Set	1 号 Mailbox 的 IPI_Set 寄存器
0x1510383c	MLB1_IPI_Clear	1 号 Mailbox 的 IPI_Clear 寄存器
0x15103840	MLB1_message0	1 号 Mailbox 的 Message0 寄存器
0x15103844	MLB1_message1	1 号 Mailbox 的 Message1 寄存器
0x15103848	MLB1_message2	1 号 Mailbox 的 Message2 寄存器
0x1510384c	MLB1_message3	1 号 Mailbox 的 Message3 寄存器
0x15103850	MLB1_message4	1 号 Mailbox 的 Message4 寄存器
0x15103854	MLB1_message5	1 号 Mailbox 的 Message5 寄存器
0x15103858	MLB1_message6	1 号 Mailbox 的 Message6 寄存器
0x1510385c	MLB1_message7	1 号 Mailbox 的 Message7 寄存器
0x15103860	MLB2_IPI_Status	2 号 Mailbox 的 IPI_Status 寄存器
0x15103864	MLB2_IPI_Enable	2 号 Mailbox 的 IPI_Enalbe 寄存器
0x15103868	MLB2_IPI_Set	2 号 Mailbox 的 IPI_Set 寄存器
0x1510386c	MLB2_IPI_Clear	2 号 Mailbox 的 IPI_Clear 寄存器
0x15103870	MLB2_message0	2 号 Mailbox 的 Message0 寄存器
0x15103874	MLB2_message1	2 号 Mailbox 的 Message1 寄存器
0x15103878	MLB2_message2	2 号 Mailbox 的 Message2 寄存器
0x1510387c	MLB2_message3	2 号 Mailbox 的 Message3 寄存器
0x15103880	MLB2_message4	2 号 Mailbox 的 Message4 寄存器
0x15103884	MLB2_message5	2 号 Mailbox 的 Message5 寄存器
0x15103888	MLB2_message6	2 号 Mailbox 的 Message6 寄存器
0x1510388c	MLB2_message7	2 号 Mailbox 的 Message7 寄存器
0x15103890	PRT_MailBox_MAP	打印系统 MailBox 中断路由配置

表 3-9 扫描系统芯片配置寄存器列表

地址	名称	描述
0x15202030	SCDEV_CLK_CTRL	扫描设备时钟门控配置
0x15202034	SCDEV_RST_CTRL	扫描设备复位配置
0x15202040	SCA_GPIO_CFG0	扫描 GPIO00~15 复用配置寄存器
0x15202044	SCA_GPIO_CFG1	扫描 GPIO16~31 复用配置寄存器
0x15202060	SCA_CHIPID	扫描系统识别号

地址	名称	描述
0x15202070	SCA_CHIP_CTRL	扫描系统通用配置
0x15202078	SCA_CHIP_SAMP	扫描系统采样状态
0x1520207c	SCA_CHIP_HPT	扫描系统计数寄存器
0x15202100	SCA_BOOT_ENTRY	扫描启动入口配置
0x15202400	SCA_INT_ISR0	扫描设备中断状态 0
0x15202404	SCA_INT_IEN0	扫描设备中断使能配置 0
0x1520240c	SCA_INT_CLR0	扫描设备中断清除配置 0
0x15202410	SCA_INT_POL0	扫描设备中断电平配置 0
0x15202414	SCA_INT_EDGE0	扫描设备中断边沿配置 0
0x15202418	SCA_INT_DUAL0	扫描设备中断双沿配置 0
0x15202420	SCA_INT_ISR1	扫描设备中断状态 1
0x15202424	SCA_INT_IEN1	扫描设备中断使能配置 1
0x1520242c	SCA_INT_CLR1	扫描设备中断清除配置 1
0x15202430	SCA_INT_POL1	扫描设备中断电平配置 1
0x15202434	SCA_INT_EDGE1	扫描设备中断边沿配置 1
0x15202438	SCA_INT_DUAL1	扫描设备中断双沿配置 1
0x15202440	SCA_INT_ISR2	扫描设备中断状态 2
0x15202444	SCA_INT_IEN2	扫描设备中断使能配置 2
0x1520244c	SCA_INT_CLR2	扫描设备中断清除配置 2
0x15202450	SCA_INT_POL2	扫描设备中断电平配置 2
0x15202454	SCA_INT_EDGE2	扫描设备中断边沿配置 2
0x15202458	SCA_INT_DUAL2	扫描设备中断双沿配置 2
0x15202460	SCA_INT_MAP0	扫描设备中断路由配置 0
0x15202464	SCA_INT_MAP1	扫描设备中断路由配置 1
0x15202468	SCA_INT_MAP2	扫描设备中断路由配置 2
0x1520246c	SCA_INT_MAP3	扫描设备中断路由配置 3
0x15202470	SCA_INT_MAP4	扫描设备中断路由配置 4
0x15202474	SCA_INT_MAP5	扫描设备中断路由配置 5
0x15202800	MLB0_IPI_Status	0 号 Mailbox 的 IPI_Status 寄存器
0x15202804	MLB0_IPI_Enable	0 号 Mailbox 的 IPI_Enable 寄存器
0x15202808	MLB0_IPI_Set	0 号 Mailbox 的 IPI_Set 寄存器
0x1520280c	MLB0_IPI_Clear	0 号 Mailbox 的 IPI_Clear 寄存器
0x15202810	MLB0_message0	0 号 Mailbox 的 Message0 寄存器
0x15202814	MLB0_message1	0 号 Mailbox 的 Message1 寄存器
0x15202818	MLB0_message2	0 号 Mailbox 的 Message2 寄存器
0x1520281c	MLB0_message3	0 号 Mailbox 的 Message3 寄存器
0x15202820	MLB0_message4	0 号 Mailbox 的 Message4 寄存器
0x15202824	MLB0_message5	0 号 Mailbox 的 Message5 寄存器
0x15202828	MLB0_message6	0 号 Mailbox 的 Message6 寄存器

地址	名称	描述
0x1520282c	MLB0_message7	0 号 Mailbox 的 Message7 寄存器
0x15202830	MLB1_IPI_Status	1 号 Mailbox 的 IPI_Status 寄存器
0x15202834	MLB1_IPI_Enable	1 号 Mailbox 的 IPI_Enable 寄存器
0x15202838	MLB1_IPI_Set	1 号 Mailbox 的 IPI_Set 寄存器
0x1520283c	MLB1_IPI_Clear	1 号 Mailbox 的 IPI_Clear 寄存器
0x15202840	MLB1_message0	1 号 Mailbox 的 Message0 寄存器
0x15202844	MLB1_message1	1 号 Mailbox 的 Message1 寄存器
0x15202848	MLB1_message2	1 号 Mailbox 的 Message2 寄存器
0x1520284c	MLB1_message3	1 号 Mailbox 的 Message3 寄存器
0x15202850	MLB1_message4	1 号 Mailbox 的 Message4 寄存器
0x15202854	MLB1_message5	1 号 Mailbox 的 Message5 寄存器
0x15202858	MLB1_message6	1 号 Mailbox 的 Message6 寄存器
0x1520285c	MLB1_message7	1 号 Mailbox 的 Message7 寄存器
0x15202860	MLB2_IPI_Status	2 号 Mailbox 的 IPI_Status 寄存器
0x15202864	MLB2_IPI_Enable	2 号 Mailbox 的 IPI_Enable 寄存器
0x15202868	MLB2_IPI_Set	2 号 Mailbox 的 IPI_Set 寄存器
0x1520286c	MLB2_IPI_Clear	2 号 Mailbox 的 IPI_Clear 寄存器
0x15202870	MLB2_message0	2 号 Mailbox 的 Message0 寄存器
0x15202874	MLB2_message1	2 号 Mailbox 的 Message1 寄存器
0x15202878	MLB2_message2	2 号 Mailbox 的 Message2 寄存器
0x1520287c	MLB2_message3	2 号 Mailbox 的 Message3 寄存器
0x15202880	MLB2_message4	2 号 Mailbox 的 Message4 寄存器
0x15202884	MLB2_message5	2 号 Mailbox 的 Message5 寄存器
0x15202888	MLB2_message6	2 号 Mailbox 的 Message6 寄存器
0x1520288c	MLB2_message7	2 号 Mailbox 的 Message7 寄存器
0x15202890	SCA_MailBox_MAP	扫描系统 Mailbox 中断路由配置

一、主系统通用配置寄存器，主要包括以下寄存器说明：

3.4.1 主系统通用配置寄存器 0

主系统通用配置寄存器 0，包括 UART 引脚复用模式、GMAC、USB、内存控制器的配置等。

寄存器地址：0x14000100。

表 3-10 通用配置寄存器 0

位域	名称	访问	缺省值	描述
31:24	conf_iodma_spare_rd	RW	0x0	iodma读操作最大数设置
23:20	conf_usb_flush_idle	RW	0xf	设置清空 write buffer 前空闲周期数
19	conf_usb_prefetch	RW	0x0	USB 接口总线使能读预取
18	conf_usb_flush_wr	RW	0x0	USB 接口总线设置写命令发出后是否清空 read buffer
17	conf_usb_stop_waw	RW	0x0	USB 接口总线是否允许在上一个写完成前发出写命令

位域	名称	访问	缺省值	描述
16	conf_usb_stop_raw	RW	0x1	USB 接口总线是否允许在上一个写完成前发出读命令
15	extioi_en	RW	0x0	扩展中断使能控制位： 1：打印扩展中断(传统/扩展中断同时有效)； 0：关闭扩展中断(仅传统中断有效)。
14	Reserved	RO	0x0	-
13:12	hpet_int_ctrl	RW	0x0	主系统 hpet0~1 中断输出模式配置： 0：单中断输出模式(内部 3 个计数器共享 1 个中断)； 1：三中断输出模式(内部 3 个计数器独立 3 个中断)。
11:10	Reserved	RO	0x0	-
9	ddr4_ecc_regroup	RW	0x0	DDR ECC 使能位
8	ddr4_shut	RW	0x0	DDR 模块关闭控制位
7	ddr4_lpconf_en	RW	0x0	DDR 低功耗软件控制使能位
6	ddr4_lpmc_en	RW	0x0	DDR 低功耗软件控制位
5	ddr4_regs_default	RW	0x1	窗口不命中处理 0：关闭访问内存控制器默认路由响应功能 1：当访问内存控制器所有窗口不命中时，由内存配置空间默认给出响应，防止内存访问卡死
4	ddr4_regs_disable	RW	0x0	DDR 配置空间关闭，高有效 DDR 控制器在内存空间中开辟了一小段配置空间(1MB @0xf0,0000)，在关闭后软件就可以使用这段空间。为避免意外访问，建议在配置完成后及时关闭
3:2	jtag_mode_config	RW	0x0	JTAG 功能调试接口复用模式配置： 00：LA264 处理器 JTAG 调试接口； 01：PR132 处理器 JTAG 调试接口； 10：SC132 处理器 JTAG 调试接口； 11：LA264->PR132->SC132 串行 JTAG。
1	gmac_test_lpbk	RW	0x0	GMAC 接口 loopback 环回测试模式使能： 1：使能 loopback 环回测试模式； 0：关闭 loopback 环回测试模式。
0	gmac_mii_sel	RW	0x0	GMAC 接口 MII 模式选择： 1：MII 接口模式；0：RGMII 接口模式。

3.4.2 主系统通用配置寄存器 1

主系统通用配置寄存器 1，包括对 USB、GMAC 等 Cache 一致性配置等。

寄存器地址：0x14000104。

表 3-11 通用配置寄存器 1

位域	名称	访问	缺省值	描述
31:27	Reserved	RO	0x0	-
26	scdev_coherent	RW	0x0	扫描系统设备 DMA 内部互联总线 CACHE 访问配置位(开启 IO 设备 CACHE 使能位后配置有效)： 1：开启 CACHE 加速访问；0：关闭 CACHE 加速访问
25	prdev_coherent	RW	0x0	打印系统设备 DMA 内部互联总线 CACHE 访问配置位(开启 IO 设备 CACHE 使能位后配置有效)： 1：开启 CACHE 加速访问；0：关闭 CACHE 加速访问

位域	名称	访问	缺省 值	描述
24	apb_coherent	RW	0x0	APB 设备 DMA 内部互联总线 CACHE 访问配置位(开启 IO 设备 CACHE 使能位后配置有效): 1: 开启 CACHE 加速访问; 0: 关闭 CACHE 加速访问
23	image_coherent	RW	0x0	IMAGE 设备(JBIG+JPEG)内部互联总线 CACHE 访问配置位(开启 IO 设备 CACHE 使能位后配置有效): 1: 开启 CACHE 加速访问; 0: 关闭 CACHE 加速访问
22	gmac_coherent	RW	0x0	GMAC 内部互联总线 CACHE 访问配置位(开启 IO 设备 CACHE 使能位后配置有效): 1: 开启 CACHE 加速访问; 0: 关闭 CACHE 加速访问
21	usb_coherent	RW	0x0	USB/OTG 内部互联总线 CACHE 访问配置位(开启 IO 设备 CACHE 使能位后配置有效): 1: 开启 CACHE 加速访问; 0: 关闭 CACHE 加速访问
20:19	Reserved	RO	0x0	—
18	scdev_coherent_enable	RW	0x1	扫描系统设备 DMA 内部互联 CACHE 访问使能位: 1: 使能设备 CACHE 访问配置有效, 配置对应设备 coherent 位开启 CACHE 加速访问; 0: 关闭设备 CACHE 访问配置, 对应设备 coherent 位配置无效, 此时设备可通过内部总线地址最高位(第 32 位)选择是否 CACHE 访问(1:开启, 0:关闭)
17	prdev_coherent_enable	RW	0x1	打印系统设备 DMA 内部互联 CACHE 访问使能位: 1: 使能设备 CACHE 访问配置有效, 配置对应设备 coherent 位开启 CACHE 加速访问; 0: 关闭设备 CACHE 访问配置, 对应设备 coherent 位配置无效, 此时设备可通过内部总线地址最高位(第 32 位)选择是否 CACHE 访问(1:开启, 0:关闭)
16	apb_coherent_enable	RW	0x1	APB 设备 DMA 内部互联 CACHE 访问使能位: 1: 使能设备 CACHE 访问配置有效, 配置对应设备 coherent 位开启 CACHE 加速访问; 0: 关闭设备 CACHE 访问配置, 对应设备 coherent 位配置无效, 此时设备可通过内部总线地址最高位(第 32 位)选择是否 CACHE 访问(1:开启, 0:关闭)
15	image_coherent_enable	RW	0x1	IMAGE 设备(JBIG+JPEG)内部互联 CACHE 访问使能位: 1: 使能设备 CACHE 访问配置有效, 配置对应设备 coherent 位开启 CACHE 加速访问; 0: 关闭设备 CACHE 访问配置, 对应设备 coherent 位配置无效, 此时设备可通过内部总线地址最高位(第 32 位)选择是否 CACHE 访问(1:开启, 0:关闭)
14	gmac_coherent_enable	RW	0x1	GMAC 设备内部互联 CACHE 访问使能位: 1: 使能设备 CACHE 访问配置有效, 配置对应设备 coherent 位开启 CACHE 加速访问; 0: 关闭设备 CACHE 访问配置, 对应设备 coherent 位配置无效, 此时设备可通过内部总线地址最高位(第 32 位)选择是否 CACHE 访问(1:开启, 0:关闭)
13	usb_coherent_enable	RW	0x1	USB/OTG 内部互联总线 CACHE 访问配置位(开启 IO 设备 CACHE 使能位后配置有效): 1: 开启 CACHE 加速访问; 0: 关闭 CACHE 加速访问

位域	名称	访问	缺省值	描述
12:11	Reserved	RO	0x0	–
10	scdev_order_en	RW	0x0	扫描系统设备 DMA 内部互联读写请求按序执行使能位，高电平有效
9	prdev_order_en	RW	0x0	打印系统设备 DMA 内部互联读写请求按序执行使能位，高电平有效
8	apb_order_en	RW	0x0	APB 设备 DMA 内部互联读写请求按序执行使能位，高电平有效
7	image_order_en	RW	0x0	IMAGE(JBIG+JPEG) 内部互联读写请求按序执行使能位，高电平有效
6	gmac_order_en	RW	0x0	GMAC 内部互联读写请求按序执行使能位，高电平有效
5	usb_order_en	RW	0x0	USB/OTG 内部互联读写请求按序执行使能位，高电平有效
4	cpu_order_en	RW	0x0	CPU 内部互联读写请求按序执行使能位，高电平有效
3:2	uart1_enable	RW	0x0	UART1 对应的 UART 控制器模式及引脚复用关系（具体引脚复用分配参看芯片数据手册 uart 接口定义）： Bit0：对应 uart1，保留； Bit1：为 1 对应 uart0 复用为 2 线模式，为 0 对应 uart1 的 4 线模式。
1:0	uart0_enable	RW	0x0	UART0 对应的 UART 控制器模式及引脚复用关系（具体引脚复用分配参看芯片数据手册 uart 接口定义）： Bit0：对应 uart0，保留； Bit1：为 1 对应 uart1 复用为 2 线模式，为 0 对应 uart0 的 4 线模式。

3.4.3 主系统通用配置寄存器 2

主系统通用配置寄存器 2，包括对各功能模块 DMA 内部互联读写请求优先级的配置等。
寄存器地址：0x14000108。

表 3-12 通用配置寄存器 2

位域	名称	访问	缺省值	描述
31:24	hpet_clkdiv	RW	0x1	主系统 HPET0/1 时钟分频系数：1~255
23:15	Reserved	RO	0x0	–
14	scdev_read_upgrade	RW	0x0	扫描系统设备 DMA 内部互联总线读请求优先级使能配置位（高电平有效，默认按序访问）
13	prdev_read_upgrade	RW	0x0	打印系统设备 DMA 内部互联总线读请求优先级使能配置位（高电平有效，默认按序访问）
12	apb_read_upgrade	RW	0x0	APB 设备 DMA 内部互联总线读请求优先级使能配置位（高电平有效，默认按序访问）
11	image_read_upgrade	RW	0x0	IMAGE(JBIG+JPEG) 内部互联总线读请求优先级使能配置位（高电平有效，默认按序访问）
10	gmac_read_upgrade	RW	0x0	GMAC 内部互联总线读请求优先级使能配置位（高电平有效，默认按序访问）
9	usb_read_upgrade	RW	0x0	USB/OTG 内部互联总线读请求优先级使能配置位（高电平有效，默认按序访问）
8:7	Reserved	RO	0x0	–
6	scdev_write_upgrade	RW	0x0	扫描系统设备 DMA 内部互联总线写请求优先级使能配置位（高电平有效，默认按序访问）
5	prdev_write_upgrade	RW	0x0	打印系统设备 DMA 内部互联总线写请求优先级使能配置位（高电平有效，默认按序访问）

位域	名称	访问	缺省值	描述
4	apb_write_upgrade	RW	0x0	APB 设备 DMA 内部互联总线写请求优先级使能配置位（高电平有效，默认按序访问）
3	image_write_upgrade	RW	0x0	IMAGE(JBIG+JPEG)内部互联总线写请求优先级使能配置位（高电平有效，默认按序访问）
2	gmac_write_upgrade	RW	0x0	GMAC 内部互联总线写请求优先级使能配置位（高电平有效，默认按序访问）
1	usb_write_upgrade	RW	0x0	USB/OTG 内部互联总线写请求优先级使能配置位（高电平有效，默认按序访问）
0	Reserved	RO	0x0	-

3.4.4 主系统通用配置寄存器 3

主系统 2 通用配置寄存器 3，包括各功能模块时钟门控的控制。

寄存器地址：0x1400010c。

表 3-13 通用配置寄存器 3

位域	名称	访问	缺省值	描述
31:30	Reserved	RO	0x1	-
29	netw_clk_ctrl	RW	0x1	Network 时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
28	node_clk_ctrl	RW	0x1	Node 时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
27	iodma_clk_ctrl	RW	0x1	IODMA 时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
26	la264_clk_ctrl	RW	0x1	LA264 时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
25	ddr_clk_ctrl	RW	0x1	DDR 模块时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
24	boot_clk_ctrl	RW	0x1	SPI0(boot flash)模块时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
23	img_clk_ctrl	RW	0x1	图像处理单元内部网络时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
22	jpeg_clk_ctrl	RW	0x1	JPEG 模块时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
21	jbig1_clk_ctrl	RW	0x1	JBIG1 模块时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
20	jbig0_clk_ctrl	RW	0x1	JBIG0 模块时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
19	otg_clk_ctrl	RW	0x1	OTG 模块时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
18	usb_clk_ctrl	RW	0x1	USB 模块时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
17	usbm_clk_ctrl	RW	0x1	USB 模块总时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
16	gmac_clk_ctrl	RW	0x1	GMAC 模块时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
15	Reserved	RO	0x1	-
14	apbm_clk_ctrl	RW	0x1	APB 设备总时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
13	sdiom_clk_ctrl	RW	0x1	SDIO 模块总时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
12	sdio1_clk_ctrl	RW	0x1	SDIO1 模块时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
11	sdio0_clk_ctrl	RW	0x1	SDIO0 模块时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。

位域	名称	访问	缺省值	描述
10	dma_clk_ctrl	RW	0x1	DMA 模块时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
9	pwm_clk_ctrl	RW	0x1	PWM 模块时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
8	pmio_m_clk_ctrl	RW	0x1	主系统 PMIO 模块时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
7	wdt_clk_ctrl	RW	0x1	WDT 模块时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
6	hpet1_clk_ctrl	RW	0x1	HPET1 模块时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
5	hpet0_clk_ctrl	RW	0x1	HPET0 模块时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
4	spi_clk_ctrl	RW	0x1	SPI 模块时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
3	i2c_clk_ctrl	RW	0x1	I ² C 模块时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
2	gpio_clk_ctrl	RW	0x1	GPIO 模块时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
1	uart1_clk_ctrl	RW	0x1	UART1 模块时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
0	uart0_clk_ctrl	RW	0x1	UART0 模块时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。

3.4.5 主系统通用配置寄存器 4

主系统通用配置寄存器 4，包括各功能模块软件复位的控制。

寄存器地址：0x14000110。

表 3-14 通用配置寄存器 4

位域	名称	访问	缺省值	描述
31:30	Reserved	RO	0x1	—
29	netw_rst_ctrl	RW	0x1	Network 复位软件配置： 1：模块复位无效；0：模块复位有效。
28	node_rst_ctrl	RW	0x1	Node 复位软件配置： 1：模块复位无效；0：模块复位有效。
27	iodma_rst_ctrl	RW	0x1	IODMA 复位软件配置： 1：模块复位无效；0：模块复位有效。
26	la264_rst_ctrl	RW	0x1	LA264 复位软件配置： 1：模块复位无效；0：模块复位有效。
25	ddr_rst_ctrl	RW	0x1	DDR 模块复位软件配置： 1：模块复位无效；0：模块复位有效。
24	boot_rst_ctrl	RW	0x1	SPI0 (boot flash) 模块复位软件配置： 1：模块复位无效；0：模块复位有效。
23	img_rst_ctrl	RW	0x1	图像处理单元内部网络复位软件配置： 1：模块复位无效；0：模块复位有效。
22	jpeg_rst_ctrl	RW	0x1	JPEG 模块复位软件配置： 1：模块复位无效；0：模块复位有效。
21	jbig1_rst_ctrl	RW	0x1	JBIG1 模块复位软件配置： 1：模块复位无效；0：模块复位有效。
20	jbig0_rst_ctrl	RW	0x1	JBIG0 模块复位软件配置： 1：模块复位无效；0：模块复位有效。
19	otg_rst_ctrl	RW	0x1	OTG 模块复位软件配置： 1：模块复位无效；0：模块复位有效。
18	usb_rst_ctrl	RW	0x1	USB 模块复位软件配置： 1：模块复位无效；0：模块复位有效。

位域	名称	访问	缺省值	描述
17	usbm_rst_ctrl	RW	0x1	USB 模块总复位软件配置： 1：模块复位无效；0：模块复位有效。
16	gmac_rst_ctrl	RW	0x1	GMAC 模块复位软件配置： 1：模块复位无效；0：模块复位有效。
15	Reserved	RO	0x1	-
14	apbm_rst_ctrl	RW	0x1	APB 设备总复位软件配置： 1：模块复位无效；0：模块复位有效。
13	sdiom_rst_ctrl	RW	0x1	SDIO 模块总复位软件配置： 1：模块复位无效；0：模块复位有效。
12	sdiol_rst_ctrl	RW	0x1	SDIO1 模块复位软件配置： 1：模块复位无效；0：模块复位有效。
11	sdio0_rst_ctrl	RW	0x1	SDIO0 模块复位软件配置： 1：模块复位无效；0：模块复位有效。
10	dma_rst_ctrl	RW	0x1	DMA 模块复位软件配置： 1：模块复位无效；0：模块复位有效。
9	pwm_rst_ctrl	RW	0x1	PWM 模块复位软件配置： 1：模块复位无效；0：模块复位有效。
8	pmio_m_rst_ctrl	RW	0x1	主系统 PMIO 模块复位软件配置： 1：模块复位无效；0：模块复位有效。
7	wdt_clk_ctrl	RW	0x1	WDT 模块复位软件配置： 1：模块复位无效；0：模块复位有效。
6	hpet1_rst_ctrl	RW	0x1	HPET1 模块复位软件配置： 1：模块复位无效；0：模块复位有效。
5	hpet0_rst_ctrl	RW	0x1	HPET0 模块复位软件配置： 1：模块复位无效；0：模块复位有效。
4	spi_rst_ctrl	RW	0x1	SPI 模块复位软件配置： 1：模块复位无效；0：模块复位有效。
3	i2c_rst_ctrl	RW	0x1	I ² C 模块复位软件配置： 1：模块复位无效；0：模块复位有效。
2	gpio_rst_ctrl	RW	0x1	GPIO 模块复位软件配置： 1：模块复位无效；0：模块复位有效。
1	uart1_rst_ctrl	RW	0x1	UART1 模块复位软件配置： 1：模块复位无效；0：模块复位有效。
0	uart0_rst_ctrl	RW	0x1	UART0 模块复位软件配置： 1：模块复位无效；0：模块复位有效。

3.4.6 主系统通用配置寄存器 5

主系统通用配置寄存器 5。

寄存器地址：0x14000114。

表 3-15 通用配置寄存器 5

位域	名称	访问	缺省值	描述
31:30	Reserved	RO	0x0	-
29:28	pad_ctrl_spi	RW	0x1	打印系统 SPI 引脚驱动类型参数配置： 00：2mA；01：4mA；10：8mA；11：12mA。
27:25	pad_ctrl_emmc	RW	0x2	eMMC0 引脚驱动类型参数配置（1.8V/3.3V）： 000：6mA；001：9mA； 010：12mA；011：15mA。
24:22	pad_ctrl_qspi	RW	0x1	QSPI 引脚驱动类型参数配置（1.8V/3.3V）： 000：6mA；001：9mA； 010：12mA；011：15mA。
21:20	pad_ctrl_usb	RW	0x1	USB OC 数字引脚驱动类型参数配置： 00：2mA；01：4mA；10：8mA；11：12mA。
19:18	pad_ctrl_scavid	RW	0x2	扫描数据接口引脚驱动类型参数配置： 00：2mA；01：4mA；10：8mA；11：12mA。

17:16	pad_ctrl_prtvid	RW	0x2	打印数据接口引脚驱动类型参数配置： 00: 2mA; 01: 4mA; 10: 8mA; 11: 12mA。
15:14	pad_ctrl_pmio_s	RW	0x2	扫描系统 PMIO 引 0~7 脚驱动类型参数配置： 00: 2mA; 01: 4mA; 10: 8mA; 11: 12mA。
13:12	pad_ctrl_pmio_p	RW	0x2	打印系统 PMIO0~23 引脚驱动类型参数配置： 00: 2mA; 01: 4mA; 10: 8mA; 11: 12mA。
11:10	pad_ctrl_pmio_m	RW	0x2	主系统 PMIO0~9 引脚驱动类型参数配置： 00: 2mA; 01: 4mA; 10: 8mA; 11: 12mA。
9:8	pad_ctrl_gmac	RW	0x2	GMAC 引脚驱动类型参数配置： 00: 2mA; 01: 4mA; 10: 8mA; 11: 12mA。
7:6	pad_ctrl_sdio	RW	0x2	SDIO1 引脚驱动类型参数配置： 00: 2mA; 01: 4mA; 10: 8mA; 11: 12mA。
5:4	pad_ctrl_i2c	RW	0x0	主系统 I ² C 引脚驱动类型参数配置： 00: 2mA; 01: 4mA; 10: 8mA; 11: 12mA。
3:2	pad_ctrl_uart	RW	0x0	主/扫描系统 UART 引脚驱动类型参数配置： 00: 2mA; 01: 4mA; 10: 8mA; 11: 12mA。
1:0	pad_ctrl_jtag	RW	0x0	JTAG 引脚驱动类型参数配置： 00: 2mA; 01: 4mA; 10: 8mA; 11: 12mA。

3.4.7 主系统通用配置寄存器 6

主系统通用配置寄存器 6。

寄存器地址：0x14000118。

表 3-16 通用配置寄存器 6

位域	名称	访问	缺省值	描述
31:0	Reserved	RO	0x0	—

3.4.8 主系统通用配置寄存器 7

主系统通用配置寄存器 7。

寄存器地址：0x1400011c。

表 3-17 通用配置寄存器 7

位域	名称	访问	缺省值	描述
31:0	Reserved	RO	0x0	—

3.4.9 主系统通用配置寄存器 8

主系统通用配置寄存器 8。

寄存器地址：0x14000120。

表 3-18 通用配置寄存器 8

位域	名称	访问	缺省值	描述
31:0	Reserved	RO	0x0	—

3.4.10 主系统通用配置寄存器 9

主系统通用配置寄存器 9。

寄存器地址：0x14000124。

表 3-19 通用配置寄存器 9

位域	名称	访问	缺省值	描述
31:22	Reserved	RO	0x0	—
21	usb_clkdiv_mode	RW	0x0	USB 参考时钟分频模式使能位，高电平有效。

				当采用 USB 内部参考时钟输入模式，且系统参考时钟为单端晶振模式（时钟频率需为 100MHz），该分频模式使能后 USB 参考时钟为系统参考时钟 5 分频后输入，时钟频率为 20MHz。
20:0	Reserved	RO	0x0	-

3.4.11 主系统采样参数寄存器

主系统采样参数寄存器，包括芯片启动配置相关信息。

寄存器地址：0x14000140。

表 3-20 主系统采样参数寄存器

位域	名称	访问	缺省值	描述
31:9	Reserved	RO	0x0	-
8	qspi_pad_type	RO	0x0	QSPI 引脚电平类型配置 0: 3.3V-I/O 类型 1: 1.8V-I/O 类型
7	emmc_pad1_type	RO	0x0	eMMC (8 线模式) 引脚电平类型配置 0: 3.3V-I/O 类型 1: 1.8V-I/O 类型
6	emmc_pad0_type	RO	0x0	eMMC (4 线模式) 引脚电平类型配置 0: 3.3V-I/O 类型 1: 1.8V-I/O 类型
5	usbrefclk_mode	RO	0x0	USB 参考时钟模式输入 0: 内部参考时钟输入 1: 外部引脚晶振输入
4	sdio1_mode	RO	0x0	SDIO1 模式配置输入 0: SDIO 模式 1: eMMC 模式
3	sdio0_mode	RO	0x0	SDIO0 模式配置输入 0: SDIO 模式 1: eMMC 模式
2:1	clk_sel	RO	0x0	芯片内部 PLL 输出时钟上电配置选择： 00: 硬件低频时钟配置模式，PLL 按照低频配置参数输出时钟； 01: 硬件高频时钟配置模式，PLL 按照高频配置参数输出时钟； 10: 软件配置模式，PLL 按照软件配置选择输出时钟； 11: 硬件 bypass 模式，PLL 输出时钟全部 bypass 使用外部输入系统时钟。
0	boot_sel	RO	0x0	芯片启动选择方式： 0: SPI0 启动；1: SDIO0/eMMC0 启动。

3.4.12 主系统高精度计数器 0

主系统计数寄存器 0，64 位高精度时钟计数器 0~31 位，工作频率为内部总线时钟频率。

寄存器地址：0x14000150。

表 3-21 主系统高精度计数器 0

位域	名称	访问	缺省值	描述
31:0	CHIP_HPT[31:0]	RW	0x0	64 位高精度时钟计数器低 32 位

3.4.13 主系统高精度计数器 1

主系统计数寄存器 1，64 位高精度时钟计数器 32~63 位，工作频率为内部总线时钟频率。

率。

寄存器地址：0x14000154。

表 3-22 主系统高精度计数器 1

位域	名称	访问	缺省值	描述
31:0	CHIP_HPT[63:32]	RW	0x0	64 位高精度时钟计数器高 32 位

3.4.14 主系统 GPIO0~15 复用配置寄存器

主系统 GPIO0~15 复用配置寄存器。

寄存器地址：0x14000490。

表 3-23 主系统 GPIO0~15 复用配置寄存器

位域	名称	访问	缺省值	描述
31:30	GPIO15_MUX	RW	0x0	GPIO15 引脚复用配置： 00: 复用为 GPIO15; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
29:28	GPIO14_MUX	RW	0x0	GPIO14 引脚复用配置： 00: 复用为 GPIO14; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
27:26	GPIO13_MUX	RW	0x0	GPIO13 引脚复用配置： 00: 复用为 GPIO13; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
25:24	GPIO12_MUX	RW	0x0	GPIO12 引脚复用配置： 00: 复用为 GPIO12; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
23:22	GPIO11_MUX	RW	0x0	GPIO11 引脚复用配置： 00: 复用为 GPIO11; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
21:20	GPIO10_MUX	RW	0x0	GPIO10 引脚复用配置： 00: 复用为 GPIO10; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
19:18	GPIO09_MUX	RW	0x0	GPIO09 引脚复用配置： 00: 复用为 GPIO09; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
17:16	GPIO08_MUX	RW	0x0	GPIO08 引脚复用配置： 00: 复用为 GPIO08; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
15:14	GPIO07_MUX	RW	0x0	GPIO07 引脚复用配置： 00: 复用为 GPIO07; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
13:12	GPIO06_MUX	RW	0x0	GPIO06 引脚复用配置： 00: 复用为 GPIO06; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
11:10	GPIO05_MUX	RW	0x0	GPIO05 引脚复用配置： 00: 复用为 GPIO05; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
9:8	GPIO04_MUX	RW	0x0	GPIO04 引脚复用配置： 00: 复用为 GPIO04; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
7:6	GPIO03_MUX	RW	0x0	GPIO03 引脚复用配置： 00: 复用为 GPIO03; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
5:4	GPIO02_MUX	RW	0x0	GPIO02 引脚复用配置： 00: 复用为 GPIO02; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。

位域	名称	访问	缺省值	描述
3:2	GPI001_MUX	RW	0x0	GPI001 引脚复用配置： 00: 复用为 GPI001; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
1:0	GPI000_MUX	RW	0x0	GPI000 引脚复用配置： 00: 复用为 GPI000; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。

3.4.15 主系统 GPI016~31 复用配置寄存器

主系统 GPI016~31 复用配置寄存器。

寄存器地址: 0x14000494。

表 3-24 主系统 GPI016~31 复用配置寄存器

位域	名称	访问	缺省值	描述
31:30	GPI031_MUX	RW	0x0	GPI031 引脚复用配置： 00: 复用为 GPI031; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
29:28	GPI030_MUX	RW	0x0	GPI030 引脚复用配置： 00: 复用为 GPI030; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
27:26	GPI029_MUX	RW	0x0	GPI029 引脚复用配置： 00: 复用为 GPI029; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
25:24	GPI028_MUX	RW	0x0	GPI028 引脚复用配置： 00: 复用为 GPI028; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
23:22	GPI027_MUX	RW	0x0	GPI027 引脚复用配置： 00: 复用为 GPI027; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
21:20	GPI026_MUX	RW	0x0	GPI026 引脚复用配置： 00: 复用为 GPI026; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
19:18	GPI025_MUX	RW	0x0	GPI025 引脚复用配置： 00: 复用为 GPI025; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
17:16	GPI024_MUX	RW	0x0	GPI024 引脚复用配置： 00: 复用为 GPI024; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
15:14	GPI023_MUX	RW	0x0	GPI023 引脚复用配置： 00: 复用为 GPI023; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
13:12	GPI022_MUX	RW	0x0	GPI022 引脚复用配置： 00: 复用为 GPI022; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
11:10	GPI021_MUX	RW	0x0	GPI021 引脚复用配置： 00: 复用为 GPI021; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
9:8	GPI020_MUX	RW	0x0	GPI020 引脚复用配置： 00: 复用为 GPI020; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
7:6	GPI019_MUX	RW	0x0	GPI019 引脚复用配置： 00: 复用为 GPI019; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
5:4	GPI018_MUX	RW	0x0	GPI018 引脚复用配置： 00: 复用为 GPI018; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。

位域	名称	访问	缺省值	描述
3:2	GPI017_MUX	RW	0x0	GPI017 引脚复用配置： 00: 复用为 GPI017; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
1:0	GPI016_MUX	RW	0x0	GPI016 引脚复用配置： 00: 复用为 GPI016; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。

3.4.16 主系统 GPI032~47 复用配置寄存器

主系统 GPI032~47 复用配置寄存器。

寄存器地址：0x14000498。

表 3-25 主系统 GPI032~47 复用配置寄存器

位域	名称	访问	缺省值	描述
31:30	GPI047_MUX	RW	0x0	GPI047 引脚复用配置： 00: 复用为 GPI047; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
29:28	GPI046_MUX	RW	0x0	GPI046 引脚复用配置： 00: 复用为 GPI046; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
27:26	GPI045_MUX	RW	0x0	GPI045 引脚复用配置： 00: 复用为 GPI045; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
25:24	GPI044_MUX	RW	0x0	GPI044 引脚复用配置： 00: 复用为 GPI044; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
23:22	GPI043_MUX	RW	0x0	GPI043 引脚复用配置： 00: 复用为 GPI043; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
21:20	GPI042_MUX	RW	0x0	GPI042 引脚复用配置： 00: 复用为 GPI042; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
19:18	GPI041_MUX	RW	0x0	GPI041 引脚复用配置： 00: 复用为 GPI041; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
17:16	GPI040_MUX	RW	0x0	GPI040 引脚复用配置： 00: 复用为 GPI040; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
15:14	GPI039_MUX	RW	0x0	GPI039 引脚复用配置： 00: 复用为 GPI039; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
13:12	GPI038_MUX	RW	0x0	GPI038 引脚复用配置： 00: 复用为 GPI038; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
11:10	GPI037_MUX	RW	0x0	GPI037 引脚复用配置： 00: 复用为 GPI037; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
9:8	GPI036_MUX	RW	0x0	GPI036 引脚复用配置： 00: 复用为 GPI036; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
7:6	GPI035_MUX	RW	0x0	GPI035 引脚复用配置： 00: 复用为 GPI035; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
5:4	GPI034_MUX	RW	0x0	GPI034 引脚复用配置： 00: 复用为 GPI034; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。

位域	名称	访问	缺省值	描述
3:2	GPI033_MUX	RW	0x0	GPI033 引脚复用配置: 00: 复用为 GPI033; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
1:0	GPI032_MUX	RW	0x0	GPI032 引脚复用配置: 00: 复用为 GPI032; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。

3.4.17 USB PHY 配置寄存器 0

USB PHY 配置寄存器 0, 配置 USB 接口 0 的电气特性, 可参考 USB-PHY 手册寄存器描述。
寄存器地址: 0x14000500。

3.4.18 USB PHY 配置寄存器 1

USB PHY 配置寄存器 1, 配置 USB 接口 1 的电气特性, 可参考 USB-PHY 手册寄存器描述。
寄存器地址: 0x14000504。

3.4.19 USB PHY 配置寄存器 2

USB PHY 配置寄存器 2, 配置 USB 接口 2 的电气特性。
寄存器地址: 0x14000508。

表 3-26 USB PHY 配置寄存器 2

位域	名称	访问	缺省值	描述
31	otg_suspend_config	RW	0x0	OTG 端口挂起软件配置位, 高有效
30	usb_suspend_config	RW	0x0	USB 端口挂起软件配置位, 高有效
29	otg_suspend_soften	RW	0x1	OTG 端口挂起软件配置使能位, 高有效
28	usb_suspend_soften	RW	0x1	USB 端口挂起软件配置使能位, 高有效
27	usb_phy_por	RW	0x0	USB 端口复位控制位, 高有效
26:19	Reserved	RO	0x0	—
18	fs_data_mod	RW	0x0	FS 数据模式
17	otg_pulldown_soften	RW	0x0	OTG 端口下拉模式软件配置使能 0-关闭; 1-使能。
16	otg_opmode_soften	RW	0x0	OTG 端口操作模式软件配置使能 0-关闭; 1-使能。
15: 14	otg_opmode0	RW	0x0	OTG 端口操作模式配置 00-normal 正常模式; 01-无驱动模式; 10-关闭位填充/NRZI 编码模式; 11-关闭 SYNC/EOP 生成模式。
13: 12	Reserved	RO	0x0	—
11	dppulldown0	RW	0x0	USB 端口 dp 端口下拉电阻使能 1: D+使能 0: D+关闭
10	dmpulldown0	RW	0x0	USB 端口 dm 端口下拉电阻使能 1: D-使能 0: D-关闭
9	dppulldown0	RW	0x0	OTG 端口 dp 端口下拉电阻使能 1: D+使能 0: D+关闭

位域	名称	访问	缺省值	描述
8	dmpulldown0	RW	0x0	OTG 端口 dm 端口下拉电阻使能 1: D-使能 0: D-关闭
7:4	Reserved	RO	0x0	-
3	usb_phy_clkssel	RW	0x1	USB 端口参考时钟模式选择位: 0-24MHz 参考时钟; 1-20MHz 参考时钟。
2	Reserved	RO	0x0	-
1	usb_utmi_resetrn	RW	0x0	USB 端口复位控制位: 0- 复位有效; 1- 复位撤销。
0	otg_utmi_resetrn	RW	0x0	OTG 端口复位控制位: 0- 复位有效; 1- 复位撤销。

二、打印系统通用配置寄存器，主要包括以下寄存器说明：

3.4.20 SYS PLL 时钟配置寄存器 0

SYS PLL 时钟配置寄存器 0，用于 SYS PLL 时钟参数配置。

寄存器地址：0x15103000。

表 3-27 SYS-PLL 时钟配置寄存器 0

位域	名称	访问	缺省值	描述
31	Reserved	RO	0x0	-
30:24	odiv_node	RW	0x0	NODE PLL 分频系数配置：0~127
23:15	div_loopc	RW	0x0	PLL 倍频系数：0~511
14:8	div_refc	RW	0x0	PLL 参考时钟分频系数：0~127
7	pll_locked	RO	0x0	PLL 锁定标志，1 代表锁定
6	Reserved	RO	0x0	-
5	pd_pll	RW	0x0	PLL 关电控制，1 代表关电
4	bypass	RW	0x0	PLL 时钟 bypass 控制，1 代表 bypass
3	pll_soft_set	RW	0x0	允许软件设置 PLL，1 代表允许软件配置
2	pll_sel_sys0	RW	0x0	SYS0 选择 PLL 时钟输出配置，1 代表选择 PLL 时钟输出
1	pll_sel_la132	RW	0x0	LA132 选择 PLL 时钟输出配置，1 代表选择 PLL 时钟输出
0	pll_sel_node	RW	0x0	NODE 选择 PLL 时钟输出配置，1 代表选择 PLL 时钟输出

3.4.21 SYSPLL 时钟配置寄存器 1

SYS PLL 时钟配置寄存器 1。

寄存器地址：0x15103004。

表 3-28 SYS-PLL 时钟配置寄存器 1

位域	名称	访问	缺省值	描述
31:25	Reserved	RO	0x0	-
24:22	ldo_ctrl	RW	0x3	PLL LDO 控制配置，保持缺省值
21	ldo_bypass	RW	0x0	PLL LDO BYPASS 配置，保持缺省值
20	ldo_en	RW	0x1	PLL LDO 使能配置，保持缺省值
19:15	Reserved	RO	0x0	-
14:8	odiv_sys0	RW	0x0	SYS0 PLL 分频系数配置：0~127

7:6	Reserved	RO	0x0	-
5:0	odiv_la132	RW	0x0	LA132 PLL 分频系数配置: 0~63

3.4.22 DDR PLL 时钟配置寄存器 0

DDR PLL 时钟配置寄存器 0, 用于 DDR PLL 时钟参数配置。

寄存器地址: 0x15103008。

表 3-29 DDR-PLL 时钟配置寄存器 0

位域	名称	访问	缺省值	描述
31	Reserved	RO	0x0	-
30:24	odiv_ddr	RW	0x0	DDR PLL 分频系数配置: 0~127
23:15	div_loopc	RW	0x0	PLL 倍频系数: 0~511
14:8	div_refc	RW	0x0	PLL 参考时钟分频系数: 0~127
7	pll_locked	RO	0x0	PLL 锁定标志, 1 代表锁定
6	Reserved	RO	0x0	-
5	pd_pll	RW	0x0	PLL 关电控制, 1 代表关电
4	bypass	RW	0x0	PLL 时钟 bypass 控制, 1 代表 bypass
3	pll_soft_set	RW	0x0	允许软件设置 PLL, 1 代表允许软件配置
2	pll_sel_sys1	RW	0x0	SYS1 选择 PLL 时钟输出配置, 1 代表选择 PLL 时钟输出
1	pll_sel_network	RW	0x0	NETWORK 选择 PLL 时钟输出配置, 1 代表选择 PLL 时钟输出
0	pll_sel_ddr	RW	0x0	DDR 选择 PLL 时钟输出配置, 1 代表选择 PLL 时钟输出

3.4.23 DDR PLL 时钟配置寄存器 1

DDR PLL 时钟配置寄存器 1, 用于 DDR PLL 时钟参数配置。

寄存器地址: 0x1510300c。

表 3-30 DDR-PLL 时钟配置寄存器 1

位域	名称	访问	缺省值	描述
31:25	Reserved	RO	0x0	-
24:22	ldo_ctrl	RW	0x3	PLL LDO控制配置, 保持缺省值
21	ldo_bypass	RW	0x0	PLL LDO BYPASS配置, 保持缺省值
20	ldo_en	RW	0x1	PLL LDO 使能配置, 保持缺省值
19:15	Reserved	RO	0x0	-
14:8	odiv_sys1	RW	0x0	SYS1 PLL 分频系数配置: 0~127
7	Reserved	RO	0x0	-
6:0	odiv_network	RW	0x0	NETWORK PLL 分频系数配置: 0~127

3.4.24 VID PLL 时钟配置寄存器 0

VID PLL 时钟配置寄存器 0, 用于 VID PLL 时钟参数配置。

寄存器地址: 0x15103010。

表 3-31 VID-PLL 时钟配置寄存器 0

位域	名称	访问	缺省值	描述
31	Reserved	RO	0x0	-
30:24	odiv_prvid	RW	0x0	PRVID PLL 分频系数配置: 0~127
23:15	div_loopc	RW	0x0	PLL 倍频系数: 0~511
14:8	div_refc	RW	0x0	PLL 参考时钟分频系数: 0~127
7	pll_locked	RO	0x0	PLL 锁定标志, 1 代表锁定

6	Reserved	RO	0x0	-
5	pd_pll	RW	0x0	PLL 关电控制，1 代表关电
4	bypass	RW	0x0	PLL 时钟 bypass 控制，1 代表 bypass
3	pll_soft_set	RW	0x0	允许软件设置 PLL，1 代表允许软件配置
2	pll_sel_prsys	RW	0x0	PRSYS 时钟选择 PLL 时钟输出配置，1 代表选择 PLL 时钟输出
1	pll_sel_scvid	RW	0x0	扫描数据 VIDEO 选择 PLL 时钟输出配置，1 代表选择 PLL 时钟输出
0	pll_sel_prvid	RW	0x0	打印数据 VIDEO 选择 PLL 时钟输出配置，1 代表选择 PLL 时钟输出

3.4.25 VID PLL 时钟配置寄存器 1

SYS PLL 时钟配置寄存器 1，用于 VID PLL 时钟参数配置。

寄存器地址：0x15103014。

表 3-32 VID-PLL 时钟配置寄存器 1

位域	名称	访问	缺省值	描述
31:25	Reserved	RO	0x0	-
24:22	ldo_ctrl	RW	0x3	PLL LDO控制配置，保持缺省值
21	ldo_bypass	RW	0x0	PLL LDO BYPASS配置，保持缺省值
20	ldo_en	RW	0x1	PLL LDO 使能配置，保持缺省值
19:15	Reserved	RO	0x0	-
14:8	odiv_prsys	RW	0x0	PRSYS PLL 分频系数配置：0~127
7	Reserved	RO	0x0	-
6:0	odiv_scvid	RW	0x0	SCVID PLL 分频系数配置：0~127

3.4.26 设备时钟分频配置寄存器 0

设备时钟分频配置寄存器 0，其中，NODE/PR132/PRXBAR/SC132/SCXBAR 模块分频，按照分频模式分为两种分频配置：

- (1) freq_mode=0，设备时钟分频计算公式为： $f_{out}=f_{in}*(freqscale[2:0]+1)/8$ ；
- (2) freq_mode=1，设备时钟分频计算公式为： $f_{out}=f_{in}/(freqscale[2:0]+1)$ 。

其他设备模块分频均为： $f_{out}=f_{in}/(*_clkdiv[3:0])$ ，分频范围：1~15，其中 $clkdiv=0/1$ 不分频。

寄存器地址：0x15103020。

表 3-33 设备时钟分频配置寄存器

位域	名称	访问	缺省值	描述
31:28	jpeg_clkdiv	RW	0x1	JPEG 时钟分配系数($CLK/*_clkdiv$): 对应 JPEG 时钟分频系数:0~15(0/1 均不分频)。 注：对应时钟源为 SYS/DDR-PLL 中 SYS0/SYS1 分频输出时钟，分频输出时钟用于 JPEG 编码模块时钟
27:24	jbig_clkdiv	RW	0x1	JBIG 时钟分配系数($CLK/*_clkdiv$): 对应 JBIG 时钟分频系数:0~15(0/1 均不分频)。 注：对应时钟源为 SYS/DDR-PLL 中 SYS0/SYS1 分频输出时钟，分频输出时钟用于 JBIG 解码模块时钟
23:20	sdio_clkdiv	RW	0x1	SDIO 时钟分配系数($CLK/*_clkdiv$): 对应 SDIO 时钟分频系数:0~15(0/1 均不分频)。 注：对应时钟源为 SYS/DDR-PLL 中 SYS0/SYS1 分频输出时钟，分频输出时钟用于 SDIO/eMMC 模块时钟

位域	名称	访问	缺省值	描述
19:16	apb_clkdiv	RW	0x1	APB 时钟分配系数(CLK/*_clkdiv): 对应 APB 时钟分频系数:0~15(0/1 均不分频)。 注: 对应时钟源为 SYS/DDR-PLL 中 SYS0/SYS1 分频输出时钟, 分频输出时钟用于主系统 APB 各功能模块时钟
15:12	boot_clkdiv	RW	0x1	BOOT 时钟分配系数(CLK/*_clkdiv): 对应 BOOT 时钟分频系数:0~15(0/1 均不分频)。 注: 对应时钟源为 SYS/DDR-PLL 中 SYS0/SYS1 分频输出时钟, 分频输出时钟用于主系统 BOOT 功能模块时钟(INT/CONF/SPI0)
11:8	gmac_clkdiv	RW	0x1	GMAC 时钟分配系数(CLK/*_clkdiv): 对应 GMAC 时钟分频系数:0~15(0/1 均不分频)。 注: 对应时钟源为 SYS/DDR/VID-PLL 中 SYS0/SYS1/PRSYS 分频输出时钟, 分频输出时钟用于 GMAC 模块时钟
7:4	usb_clkdiv	RW	0x1	USB 时钟分配系数(CLK/*_clkdiv): 对应 USB 时钟分频系数:0~15(0/1 均不分频)。 注: 对应时钟源为 SYS/DDR-PLL 中 SYS0/SYS1 分频输出时钟, 分频输出时钟用于 USB/OTG 模块时钟
3:0	node_freqscale	RW	0x7	node_freqscale[3]:NODE 时钟分频模式 freq_mode; node_freqscale[2:0]:NODE 时钟输出分频系数: 0~7。 注: 对应时钟源为 SYS PLL 中 NODE 分频输出时钟, 分频输出时钟用于 NODE 模块 (LA264/SCache/IODMA)

3.4.27 设备时钟分频配置寄存器 1

设备时钟分频配置寄存器 1, 其中, NODE/PR132/PRXBAR/SC132/SCXBAR 模块分频, 按照分频模式分为两种分频配置:

- (1) freq_mode=0, 设备时钟分频计算公式为: $f_{out} = f_{in} * (freqscale[2:0] + 1) / 8$;
- (2) freq_mode=1, 设备时钟分频计算公式为: $f_{out} = f_{in} / (freqscale[2:0] + 1)$ 。

其他设备模块分频均为: $f_{out} = f_{in} / (*_clkdiv[3:0])$, 分频范围: 1~15, 其中 clkdiv=0/1 不分频。

寄存器地址: 0x15103024。

表 3-34 设备时钟分频配置寄存器

位域	名称	访问	缺省值	描述
31:20	Reserved	RO	0x0	-
19:16	prvid_clkdiv	RW	0x1	PRVID 时钟分配系数(CLK/*_clkdiv): 对应 PRVID 时钟分频系数:0~15(0/1 均不分频)。 注: 对应时钟源为 VID PLL 中 PRVID 分频输出时钟, 分频输出时钟用于 PRINTER 打印功能模块时钟
15:12	prifc_clkdiv	RW	0x1	PRIFC 时钟分配系数(CLK/*_clkdiv): 对应 PRIFC 时钟分频系数:0~15(0/1 均不分频)。 注: 对应时钟源为 VID PLL 中 PRVID 分频输出时钟, 分频输出时钟用于 PRINTER 打印模块内部总线时钟
11:8	prdev_clkdiv	RW	0x1	PRDEV 时钟分配系数(CLK/*_clkdiv): 对应 PRDEV 时钟分频系数:0~15(0/1 均不分频)。 注: 对应时钟源为 VID/SYS/DDR-PLL 中 PRSYS/SYS0/SYS1 分频输出时钟, 分频输出时钟用于打印系统内各设备模块时钟

位域	名称	访问	缺省值	描述
7:4	prxbar_freqscale	RW	0x7	prxbar_freqscale[3]:PRXBAR 时钟分频模式 freq_mode; prxbar_freqscale[2:0]:PRXBAR 时钟输出分频系数: 0~7 注: 对应时钟源为 SYS PLL 中 LA132 分频输出时钟, 分频输出时钟用于打印系统内部互连模块时钟
3:0	prl32_freqscale	RW	0x7	prl32_freqscale[3]:PR132 时钟分频模式 freq_mode; prl32_freqscale[2:0]:PR132 时钟输出分频系数: 0~7 注: 对应时钟源为 SYS PLL 中 LA132 分频输出时钟, 分频输出时钟用于打印系统 LA132 核时钟

3.4.28 设备时钟分频配置寄存器 2

设备时钟分频配置寄存器 2, 其中, NODE/PR132/PRXBAR/SC132/SCXBAR 模块分频, 按照分频模式分为两种分频配置:

- (1) freq_mode=0, 设备时钟分频计算公式为: $f_{out}=f_{in}*(freqscale[2:0]+1)/8$;
- (2) freq_mode=1, 设备时钟分频计算公式为: $f_{out}=f_{in}/(freqscale[2:0]+1)$ 。

其他设备模块分频均为: $f_{out}=f_{in}/(*_clkdiv[3:0])$, 分频范围: 1~15, 其中 clkdiv=0/1 不分频。

寄存器地址: 0x15103028。

表 3-35 设备时钟分频配置寄存器

位域	名称	访问	缺省值	描述
31:12	Reserved	RO	0x0	-
11:8	scdev_clkdiv	RW	0x1	SCDEV 时钟分配系数(CLK/*_clkdiv): 对应 SCDEV 时钟分频系数:0~15(0/1 均不分频)。 注: 对应时钟源为 VID/SYS/DDR-PLL 中 PRSYS/SYS0/SYS1 分频输出时钟, 分频输出时钟用于扫描系统内各设备模块时钟
7:4	scxbar_freqscale	RW	0x7	scxbar_freqscale[3]:SCDEV 时钟分频模式 freq_mode; scxbar_freqscale[2:0]:SCDEV 时钟输出分频系数: 0~7 注: 对应时钟源为 SYS PLL 中 LA132 分频输出时钟, 分频输出时钟用于扫描系统内部互连模块时钟
3:0	sc132_freqscale	RW	0x7	sc132_freqscale[3]:SC132 时钟分频模式 freq_mode; sc132_freqscale[2:0]:SC132 时钟输出分频系数: 0~7 注: 对应时钟源为 SYS PLL 中 LA132 分频输出时钟, 分频输出时钟用于扫描系统 LA132 核时钟

3.4.29 设备时钟输出配置寄存器

设备时钟输出配置寄存器,

寄存器地址: 0x1510302c。

表 3-36 设备时钟输出使能配置寄存器

位域	名称	访问	缺省值	描述
31:24	Reserved	RO	0x7f	-
23	scifc_clken	RW	0x1	scifc 模块时钟输出使能配置: 1: 时钟输出使能; 0: 时钟输出关闭

位域	名称	访问	缺省值	描述
22	scdev_clken	RW	0x1	scdev 模块时钟输出使能配置： 1: 时钟输出使能; 0: 时钟输出关闭
21	scxbar_clken	RW	0x1	scxbar 模块时钟输出使能配置： 1: 时钟输出使能; 0: 时钟输出关闭
20	sc132_clken	RW	0x1	sc132 模块时钟输出使能配置： 1: 时钟输出使能; 0: 时钟输出关闭
19:17	Reserved	RO	0x7	-
16	prvid_clken	RW	0x1	prvid 模块时钟输出使能配置： 1: 时钟输出使能; 0: 时钟输出关闭
15	prifc_clken	RW	0x1	prifc 模块时钟输出使能配置： 1: 时钟输出使能; 0: 时钟输出关闭
14	prdev_clken	RW	0x1	prdev 模块时钟输出使能配置： 1: 时钟输出使能; 0: 时钟输出关闭
13	prxbar_clken	RW	0x1	prxbar 模块时钟输出使能配置： 1: 时钟输出使能; 0: 时钟输出关闭
12	pr132_clken	RW	0x1	pr132 模块时钟输出使能配置： 1: 时钟输出使能; 0: 时钟输出关闭
11:10	Reserved	RO	0x0	-
9	jpeg_clken	RW	0x1	jpeg 模块时钟输出使能配置： 1: 时钟输出使能; 0: 时钟输出关闭
8	jbig_clken	RW	0x1	jbig 模块时钟输出使能配置： 1: 时钟输出使能; 0: 时钟输出关闭
7	Reserved	RO	0x0	-
6	apb_clken	RW	0x1	apb 设备时钟输出使能配置： 1: 时钟输出使能; 0: 时钟输出关闭
5	boot_clken	RW	0x1	boot 模块时钟输出使能配置： 1: 时钟输出使能; 0: 时钟输出关闭
4	Reserved	RO	0x0	-
3	usb_clken	RW	0x1	usb 模块时钟输出使能配置： 1: 时钟输出使能; 0: 时钟输出关闭
2:1	Reserved	RO	0x0	-
0	node_clken	RW	0x1	node 模块时钟输出使能配置： 1: 时钟输出使能; 0: 时钟输出关闭

3.4.30 打印系统设备时钟门控寄存器

打印系统各设备时钟门控寄存器，包括打印系统各功能模块时钟门控的控制。

寄存器地址：0x15103030。

表 3-37 打印系统设备时钟门控寄存器

位域	名称	访问	缺省值	描述
31:17	Reserved	RO	0x0	-
16	scsys_clk_ctrl	RW	0x1	SC132 Network 时钟门控配置： 1: 时钟门控打开; 0: 时钟门控关闭。
15	pr132_clk_ctrl	RW	0x1	PR132 CORE 及 JTAG 时钟门控配置： 1: 时钟门控打开; 0: 时钟门控关闭。
14	prsrsm_clk_ctrl	RW	0x1	PR132 SRAM 时钟门控配置： 1: 时钟门控打开; 0: 时钟门控关闭。
13	Reserved	RO	0x0	-
12	prdev_clk_ctrl	RW	0x1	打印系统 DMA 及打印系统模块总时钟门控配置： 1: 时钟门控打开; 0: 时钟门控关闭。
11	otp_clk_ctrl	RW	0x1	打印系统 OTP 模块时钟门控配置： 1: 时钟门控打开; 0: 时钟门控关闭。
10	pri_clk_ctrl	RW	0x1	打印系统 Printer 模块时钟门控配置： 1: 时钟门控打开; 0: 时钟门控关闭。

位域	名称	访问	缺省值	描述
9	hpet1_clk_ctrl	RW	0x1	打印系统 HPET1 模块时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
8	hpet0_clk_ctrl	RW	0x1	打印系统 HPET0 模块时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
7	dma_clk_ctrl	RW	0x1	打印系统 DMA 模块时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
6	pmio_p_clk_ctrl	RW	0x1	打印系统 PMOIO 模块时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
5	pwm_clk_ctrl	RW	0x1	打印系统 PWM 模块时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
4	adc_clk_ctrl	RW	0x1	打印系统 ADC 模块时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
3	i2c_clk_ctrl	RW	0x1	打印系统 I ² C 模块时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
2	gpio_clk_ctrl	RW	0x1	打印系统 GPIO 模块时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
1	uart1_clk_ctrl	RW	0x1	打印系统 UART1 模块时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
0	uart0_clk_ctrl	RW	0x1	打印系统 UART0 模块时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。

3.4.31 打印系统设备复位控制寄存器

打印系统各设备复位配置寄存器，包括打印系统各功能模块软件复位的控制。

寄存器地址：0x15103034。

表 3-38 打印系统设备复位控制寄存器

位域	名称	访问	缺省值	描述
31:17	Reserved	RO	0x0	—
16	scsys_rst_ctrl	RW	0x1	SC132 Network 复位软件配置： 1：模块复位无效；0：模块复位有效。
15	prl32_rst_ctrl	RW	0x1	PR132 CORE 及 JTAG 复位软件配置： 1：模块复位无效；0：模块复位有效。
14	prsrsm_rst_ctrl	RW	0x1	PR132 SRAM 复位软件配置： 1：模块复位无效；0：模块复位有效。
13	Reserved	RO	0x0	—
12	prdev_rst_ctrl	RW	0x1	打印系统 DMA 及打印系统模块总复位软件配置： 1：模块复位无效；0：模块复位有效。
11	otp_rst_ctrl	RW	0x1	打印系统 OTP 模块复位软件配置： 1：模块复位无效；0：模块复位有效。
10	pri_rst_ctrl	RW	0x1	打印系统 Printer 模块复位软件配置： 1：模块复位无效；0：模块复位有效。
9	hpet1_rst_ctrl	RW	0x1	打印系统 HPET1 模块复位软件配置： 1：模块复位无效；0：模块复位有效。
8	hpet0_rst_ctrl	RW	0x1	打印系统 HPET0 模块复位软件配置： 1：模块复位无效；0：模块复位有效。
7	dma_rst_ctrl	RW	0x1	打印系统 DMA 模块复位软件配置： 1：模块复位无效；0：模块复位有效。
6	pm0io_p_rst_ctrl	RW	0x1	打印系统 PMOIO 模块复位软件配置： 1：模块复位无效；0：模块复位有效。
5	pwm_rst_ctrl	RW	0x1	打印系统 PWM 模块复位软件配置： 1：模块复位无效；0：模块复位有效。
4	adc_rst_ctrl	RW	0x1	打印系统 ADC 模块复位软件配置： 1：模块复位无效；0：模块复位有效。
3	i2c_rst_ctrl	RW	0x1	打印系统 I ² C 模块复位软件配置： 1：模块复位无效；0：模块复位有效。

位域	名称	访问	缺省值	描述
2	gpio_rst_ctrl	RW	0x1	打印系统 GPIO 模块复位软件配置： 1：模块复位无效；0：模块复位有效。
1	uart1_rst_ctrl	RW	0x1	打印系统 UART1 模块复位软件配置： 1：模块复位无效；0：模块复位有效。
0	uart0_rst_ctrl	RW	0x1	打印系统 UART0 模块复位软件配置： 1：模块复位无效；0：模块复位有效。

3.4.32 打印系统 GPIO0~15 复用配置寄存器

打印系统 GPIO0~15 复用配置寄存器。

寄存器地址：0x15103040。

表 3-39 打印系统 GPIO0~15 复用配置寄存器

位域	名称	访问	缺省值	描述
31:30	GPIO15_MUX	RW	0x0	GPIO15 引脚复用配置： 00：复用为 GPIO15；01：第一复用； 10：第二复用； 11：引脚主功能。
29:28	GPIO14_MUX	RW	0x0	GPIO14 引脚复用配置： 00：复用为 GPIO14；01：第一复用； 10：第二复用； 11：引脚主功能。
27:26	GPIO13_MUX	RW	0x0	GPIO13 引脚复用配置： 00：复用为 GPIO13；01：第一复用； 10：第二复用； 11：引脚主功能。
25:24	GPIO12_MUX	RW	0x0	GPIO12 引脚复用配置： 00：复用为 GPIO12；01：第一复用； 10：第二复用； 11：引脚主功能。
23:22	GPIO11_MUX	RW	0x0	GPIO11 引脚复用配置： 00：复用为 GPIO11；01：第一复用； 10：第二复用； 11：引脚主功能。
21:20	GPIO10_MUX	RW	0x0	GPIO10 引脚复用配置： 00：复用为 GPIO10；01：第一复用； 10：第二复用； 11：引脚主功能。
19:18	GPIO09_MUX	RW	0x0	GPIO09 引脚复用配置： 00：复用为 GPIO09；01：第一复用； 10：第二复用； 11：引脚主功能。
17:16	GPIO08_MUX	RW	0x0	GPIO08 引脚复用配置： 00：复用为 GPIO08；01：第一复用； 10：第二复用； 11：引脚主功能。
15:14	GPIO07_MUX	RW	0x0	GPIO07 引脚复用配置： 00：复用为 GPIO07；01：第一复用； 10：第二复用； 11：引脚主功能。
13:12	GPIO06_MUX	RW	0x0	GPIO06 引脚复用配置： 00：复用为 GPIO06；01：第一复用； 10：第二复用； 11：引脚主功能。
11:10	GPIO05_MUX	RW	0x0	GPIO05 引脚复用配置： 00：复用为 GPIO05；01：第一复用； 10：第二复用； 11：引脚主功能。
9:8	GPIO04_MUX	RW	0x0	GPIO04 引脚复用配置： 00：复用为 GPIO04；01：第一复用； 10：第二复用； 11：引脚主功能。
7:6	GPIO03_MUX	RW	0x0	GPIO03 引脚复用配置： 00：复用为 GPIO03；01：第一复用； 10：第二复用； 11：引脚主功能。
5:4	GPIO02_MUX	RW	0x0	GPIO02 引脚复用配置： 00：复用为 GPIO02；01：第一复用； 10：第二复用； 11：引脚主功能。

位域	名称	访问	缺省值	描述
3:2	GPI001_MUX	RW	0x0	GPI001 引脚复用配置： 00: 复用为 GPI001；01: 第一复用； 10: 第二复用； 11: 引脚主功能。
1:0	GPI000_MUX	RW	0x0	GPI000 引脚复用配置： 00: 复用为 GPI000；01: 第一复用； 10: 第二复用； 11: 引脚主功能。

3.4.33 打印系统 GPIO16~31 复用配置寄存器

打印系统 GPIO16~31 复用配置寄存器。

寄存器地址：0x15103044。

表 3-40 打印系统 GPIO16~31 复用配置寄存器

位域	名称	访问	缺省值	描述
31:30	GPI031_MUX	RW	0x0	GPI031 引脚复用配置： 00: 复用为 GPI031；01: 第一复用； 10: 第二复用； 11: 引脚主功能。
29:28	GPI030_MUX	RW	0x0	GPI030 引脚复用配置： 00: 复用为 GPI030；01: 第一复用； 10: 第二复用； 11: 引脚主功能。
27:26	GPI029_MUX	RW	0x0	GPI029 引脚复用配置： 00: 复用为 GPI029；01: 第一复用； 10: 第二复用； 11: 引脚主功能。
25:24	GPI028_MUX	RW	0x0	GPI028 引脚复用配置： 00: 复用为 GPI028；01: 第一复用； 10: 第二复用； 11: 引脚主功能。
23:22	GPI027_MUX	RW	0x0	GPI027 引脚复用配置： 00: 复用为 GPI027；01: 第一复用； 10: 第二复用； 11: 引脚主功能。
21:20	GPI026_MUX	RW	0x0	GPI026 引脚复用配置： 00: 复用为 GPI026；01: 第一复用； 10: 第二复用； 11: 引脚主功能。
19:18	GPI025_MUX	RW	0x0	GPI025 引脚复用配置： 00: 复用为 GPI025；01: 第一复用； 10: 第二复用； 11: 引脚主功能。
17:16	GPI024_MUX	RW	0x0	GPI024 引脚复用配置： 00: 复用为 GPI024；01: 第一复用； 10: 第二复用； 11: 引脚主功能。
15:14	GPI023_MUX	RW	0x0	GPI023 引脚复用配置： 00: 复用为 GPI023；01: 第一复用； 10: 第二复用； 11: 引脚主功能。
13:12	GPI022_MUX	RW	0x0	GPI022 引脚复用配置： 00: 复用为 GPI022；01: 第一复用； 10: 第二复用； 11: 引脚主功能。
11:10	GPI021_MUX	RW	0x0	GPI021 引脚复用配置： 00: 复用为 GPI021；01: 第一复用； 10: 第二复用； 11: 引脚主功能。
9:8	GPI020_MUX	RW	0x0	GPI020 引脚复用配置： 00: 复用为 GPI020；01: 第一复用； 10: 第二复用； 11: 引脚主功能。
7:6	GPI019_MUX	RW	0x0	GPI019 引脚复用配置： 00: 复用为 GPI019；01: 第一复用； 10: 第二复用； 11: 引脚主功能。
5:4	GPI018_MUX	RW	0x0	GPI018 引脚复用配置： 00: 复用为 GPI018；01: 第一复用； 10: 第二复用； 11: 引脚主功能。

位域	名称	访问	缺省值	描述
3:2	GPI017_MUX	RW	0x0	GPI017 引脚复用配置： 00: 复用为 GPI017; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
1:0	GPI016_MUX	RW	0x0	GPI016 引脚复用配置： 00: 复用为 GPI016; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。

3.4.34 设备时钟输出模式配置寄存器

设备时钟输出模式配置寄存器，

寄存器地址：0x15103050。

表 3-41 设备时钟输出模式配置寄存器

位域	名称	访问	缺省值	描述
31:23	Reserved	RO	0x0	-
22:20	scdev_clkmode	RW	0x0	扫描系统 APB 设备时钟源选择配置： 000: VID-PLL prsys 001: SYS-PLL sys0 01X: DDR-PLL sys1 1XX: 系统参考时钟
19	Reserved	RO	0x0	-
18:16	prdev_clkmode	RW	0x0	打印系统 APB 设备时钟源选择配置： 000: VID-PLL prsys 001: SYS-PLL sys0 01X: DDR-PLL sys1 1XX: 系统参考时钟
15	Reserved	RO	0x0	-
14:12	gmac_clkmode	RW	0x0	GMAC 模块时钟源选择配置： 000: SYS-PLL sys0 001: DDR-PLL sys1 01X: VID-PLL prsys 1XX: 系统参考时钟
11:10	jpeg_clkmode	RW	0x0	JPEG 模块时钟源选择配置： 00: SYS-PLL sys0 01: DDR-PLL sys1 1X: 系统参考时钟
9:8	jbig_clk_mode	RW	0x0	JBIG 模块时钟源选择配置： 00: SYS-PLL sys0 01: DDR-PLL sys1 1X: 系统参考时钟
7:6	sdio_clkmode	RW	0x0	SDIO 模块时钟源选择配置： 00: SYS-PLL sys0 01: DDR-PLL sys1 1X: 系统参考时钟
5:3	apb_clkmode	RW	0x0	主系统 APB 设备时钟源选择配置： 00: SYS-PLL sys0 01: DDR-PLL sys1 1X: 系统参考时钟
3:2	boot_clkmode	RW	0x0	BOOT 模块时钟源选择配置： 00: SYS-PLL sys0 01: DDR-PLL sys1 1X: 系统参考时钟
1:0	usb_clkmode	RW	0x0	USB 模块时钟源选择配置： 00: SYS-PLL sys0 01: DDR-PLL sys1 1X: 系统参考时钟

3.4.35 打印系统通用配置寄存器 0

打印系统通用配置寄存器 0，包括打印模块、PM0IO、RTC 等参数配置。

寄存器地址：0x15103070。

表 3-42 通用配置寄存器 0

位域	名称	访问	缺省值	描述
31:28	Reserved	RO	0x0	—
27:26	adc_jtrgr_sel	RW	0x0	ADC 控制器注入组触发信号选择： 0: PMIO_P[6:0] 1: PMIO_P[22:16] 2: PMIO_M[6:0] 3: PMIO_S[6:0]
25:24	adc_rtrgr_sel	RW	0x0	ADC 控制器规则组触发信号选择： 0: 打印系统 PMIO_P[6:0] 1: 打印系统 PMIO_P[22:16] 2: 主系统 PMIO_M[6:0] 3: 扫描系统 PMIO_S[6:0]
23:16	hpet_clkdiv	RW	0x1	打印系统 HPET0/1 时钟分频系数：1~255
15:14	Reserved	RO	0x0	—
13	pmio_p_extin_sel	RW	0x0	打印系统 PMIO 模块 EXTIN[31:0]信号输入选择： 0: 由 PM1IO_INTCON[31:0]端口输入； 1: 由 PM1IO_INTCON[31:6]、 VIDCLK_OUT[1:0]、PRT_OVLIN[3:0]依次输入。
12	printer_exttimin_sel	RW	0x0	打印模块 16 路外部 TIMING 信号输入选择： 0: 由 PM0IO[15:00]内部输出端口信号输入； 1: 由 PM0IO[31:16]内部输出端口信号输入。
11	otp_ctrl_en	RW	0x0	OTP 模块使能位
10:8	adc_dma_map	RW	0x0	ADC模块DMA通道映射： RX对应的DMA通道为 000-111：通用 DMA 通道 0-7。
7	hpet1_int_ctrl	RW	0x0	打印系统 hpet1 中断输出模式配置： 0: 单中断输出模式(内部 3 个计数器共享 1 个中断)； 1: 三中断输出模式(内部 3 个计数器独立 3 个中断)
6	hpet0_int_ctrl	RW	0x0	打印系统 hpet0 中断输出模式配置： 0: 单中断输出模式(内部 3 个计数器共享 1 个中断)； 1: 三中断输出模式(内部 3 个计数器独立 3 个中断)。
5	prl32_nmi	RW	0x0	打印系统处理器核 NMI 输入信号，高电平有效
4	Reserved	RO	0x0	—
3:2	prvidout_lvds_tm	RW	0x0	2 路打印数据 LVDS 差分端口输出模式配置： 0: 数据 normal 输出模式(时钟采样后输出)； 1: 数据 bypass 输出模式。
1:0	prvidout_lvds_pd	RW	0x3	2 路打印数据 LVDS 差分端口 PD 配置： 0: poweron, 差分端口正常工作，LVDS 模式输出； 1: powerdown, 差分端口不工作，TTL 模式输出。

3.4.36 打印系统通用配置寄存器 1

打印系统通用配置寄存器 1，配置 OTG、GMAC 等设备 DMA 共享存储、基址路由。
寄存器地址：0x15103074。

表 3-43 通用配置寄存器 1

位域	名称	访问	缺省值	描述
31:16	otgdma_base_addr	RW	0x0	OTG设备DMA访问共享存储基址配置： otgdma_base_addr[0]：共享使能位： 0-不可共享打印系统SRAM(128KB)，仅可访问主系统内存； 1-可共享打印系统SRAM(128KB)，通过地址路由访问主系统内存、打印系统SRAM。 otgdma_base_addr[15:1]：共享打印系统 SRAM 存储基址，对应设备 DMA 共享访问地址[31:17]路由。
15:0	gmacdma_base_addr	RW	0x0	GMAC设备DMA访问共享存储基址配置： gmacdma_base_addr[0]：共享使能位： 0-不可共享打印系统SRAM(128KB)，仅可访问主系统内存； 1-可共享打印系统SRAM(128KB)，通过地址路由访问主系统内存、打印系统SRAM。 gmacdma_base_addr[15:1]：共享打印系统 SRAM 存储基址，对应设备 DMA 共享访问地址[31:17]路由。

3.4.37 打印系统采样参数寄存器

打印系统采样参数寄存器，包括打印系统启动配置及其他系统休眠相关信息。
寄存器地址：0x15103078。

表 3-44 芯片采样参数寄存器

位域	名称	访问	缺省值	描述
31:11	Reserved	RO	0x0	-
10	sc132_sleeping_sta	RO	0x0	扫描系统处理器核休眠状态监测： 0=WORKON 状态 1=SLEEPING 状态
9	pr132_sleeping_sta	RO	0x0	打印系统处理器核休眠状态监测： 0=WORKON 状态 1=SLEEPING 状态
8	qspi_pad_type	RO	0x0	QSPI 引脚电平类型配置 0：3.3V-I/O 类型 1：1.8V-I/O 类型
7	emmc_pad1_type	RO	0x0	eMMC（8 线模式）引脚电平类型配置 0：3.3V-I/O 类型 1：1.8V-I/O 类型
6	emmc_pad0_type	RO	0x0	eMMC（4 线模式）引脚电平类型配置 0：3.3V-I/O 类型 1：1.8V-I/O 类型
5	usbrefclk_mode	RO	0x0	USB 参考时钟模式输入 0：内部参考时钟输入 1：外部 PAD 晶振输入
4	sdio1_mode	RO	0x0	SDIO1 模式配置输入 0：SDIO 模式 1：eMMC 模式

3	sdio0_mode	RO	0x0	SDIO0 模式配置输入 0: SDIO 模式 1: eMMC 模式
2:1	clk_sel	RO	0x0	芯片内部 PLL 输出时钟上电配置选择: 00: 硬件低频时钟配置模式, PLL 按照低频配置参数输出时钟; 01: 硬件高频时钟配置模式, PLL 按照高频配置参数输出时钟; 10: 软件配置模式, PLL 按照软件配置选择输出时钟; 11: 硬件 bypass 模式, PLL 输出时钟全部使用外部输入系统时钟。
0	boot_sel	RO	0x0	芯片启动选择方式: 0: SPI0 启动; 1: SDIO0/eMMC0 启动。

3.4.38 打印系统高精度计数器

打印系统计数寄存器, 32 位高精度时钟计数器, 工作频率为内部总线时钟频率。

寄存器地址: 0x1510307c。

表 3-45 打印系统高精度计数器

位域	名称	访问	缺省值	描述
31:0	PRCHIP_HPT	RW	0x0	32 位高精度时钟计数器

3.4.39 打印系统设备 DMA 路由基址配置寄存器

打印系统设备 DMA 路由基址配置寄存器, 配置共享打印系统 SRAM、打印设备 DMA 基址路由。

寄存器地址: 0x151030e0。

表 3-46 打印系统设备 DMA 路由基址配置寄存器

位域	名称	访问	缺省值	描述
31:16	prdma_base_addr	RW	0x0	打印系统设备 DMA 访问共享存储基址配置: prdma_base_addr[0]: 共享使能位: 0-不可共享打印系统 SRAM(128KB), 仅可访问主系统内存; 1-可共享打印系统 SRAM(128KB), 通过地址路由访问主系统内存、打印系统 SRAM。 prdma_base_addr[15:1]: 共享打印系统 SRAM 存储基址, 对应设备 DMA 共享访问地址[31:17]路由。
15:0	prsrsm_base_addr	RW	0x1500	其他系统共享访问打印系统 SRAM 访问基址配置, 对应其他系统共享访问地址[31:16]路由

3.4.40 打印系统启动入口配置寄存器

打印系统启动入口配置寄存器, 配置打印系统处理器核启动地址。

寄存器地址: 0x15103100。

表 3-47 打印系统启动入口配置寄存器

位域	名称	访问	缺省值	描述
31:0	prl32_boot_entry	RW	0x1c000000	配置打印系统处理器核启动入口地址, bit[0]为地址配置有效使能位, 高电平有效: 1-入口地址配置有效; 0-入口地址配置无效, 采用默认启动入口地址(0x1c000000)

3.4.41 打印系统 0 号 MAILBOX 中断状态寄存器

打印系统 0 号 MAILBOX 中断状态寄存器，记录中断触发状态。

寄存器地址：0x15103800。

表 3-48 打印系统 0 号 MAILBOX 中断状态寄存器

位域	名称	访问	缺省值	描述
31:0	mlb0_ipi_status	RO	0x0	0 号 Mailbox 中断状态寄存器，记录被触发中断状态位，任何一位被置 1 且对应中断使能情况下，路由至对应处理器核中断位将被置位

3.4.42 打印系统 0 号 MAILBOX 中断使能寄存器

打印系统 0 号 MAILBOX 中断使能寄存器，使能相应中断位。

寄存器地址：0x15103804。

表 3-49 打印系统 0 号 MAILBOX 中断使能寄存器

位域	名称	访问	缺省值	描述
31:0	mlb0_ipi_enable	RW	0x0	0 号 Mailbox 中断使能寄存器，使能对应中断位，高电平有效

3.4.43 打印系统 0 号 MAILBOX 中断触发寄存器

打印系统 0 号 MAILBOX 中断触发寄存器，触发相应中断位。

寄存器地址：0x15103808。

表 3-50 打印系统 0 号 MAILBOX 中断触发寄存器

位域	名称	访问	缺省值	描述
31:0	mlb0_ipi_set	W	-	0 号 Mailbox 中断触发寄存器，触发对应中断位，高电平有效。当相应中断使能位有效时，该位置 1 将触发相应中断，并将路由至对应处理器核中断位置位

3.4.44 打印系统 0 号 MAILBOX 中断清除寄存器

打印系统 0 号 MAILBOX 中断清除寄存器，清除相应中断状态位。

寄存器地址：0x1510380c。

表 3-51 打印系统 0 号 MAILBOX 中断清除寄存器

位域	名称	访问	缺省值	描述
31:0	mlb0_ipi_clear	W	-	0 号 Mailbox 中断清除寄存器，清除对应中断位，写 1 清除。对应位写 1 将清除相应中断状态位，取消相应中断触发，中断使能位不做清除

3.4.45 打印系统 0 号 MAILBOX 消息寄存器 0~7

打印系统 0 号 MAILBOX 消息寄存器 0~7，传递记录相应消息，可由用户自定义。

寄存器地址：0x15103810~0x1510382c。

表 3-52 打印系统 0 号 MAILBOX 消息寄存器 0~7

位域	名称	访问	缺省值	描述
31:0	mlb0_message7	RW	0x0	0 号 Mailbox 消息寄存器 7，记录相应消息定义，可进行不同系统间消息传递
31:0	mlb0_message6	RW	0x0	0 号 Mailbox 消息寄存器 6，记录相应消息定义，可进行不同系统间消息传递

位域	名称	访问	缺省值	描述
31:0	mlb0_message5	RW	0x0	0 号 Mailbox 消息寄存器 5，记录相应消息定义，可进行不同系统间消息传递
31:0	mlb0_message4	RW	0x0	0 号 Mailbox 消息寄存器 4，记录相应消息定义，可进行不同系统间消息传递
31:0	mlb0_message3	RW	0x0	0 号 Mailbox 消息寄存器 3，记录相应消息定义，可进行不同系统间消息传递
31:0	mlb0_message2	RW	0x0	0 号 Mailbox 消息寄存器 2，记录相应消息定义，可进行不同系统间消息传递
31:0	mlb0_message1	RW	0x0	0 号 Mailbox 消息寄存器 1，记录相应消息定义，可进行不同系统间消息传递
31:0	mlb0_message0	RW	0x0	0 号 Mailbox 消息寄存器 0，记录相应消息定义，可进行不同系统间消息传递

3.4.46 打印系统 1 号 MAILBOX 中断状态寄存器

打印系统 1 号 MAILBOX 中断状态寄存器，记录中断触发状态。

寄存器地址：0x15103830。

表 3-53 打印系统 1 号 MAILBOX 中断状态寄存器

位域	名称	访问	缺省值	描述
31:0	mlb1_ipi_status	RO	0x0	1 号 Mailbox 中断状态寄存器，记录被触发中断状态位，任何一位被置 1 且对应中断使能情况下，路由至对应处理器核中断位将被置位

3.4.47 打印系统 1 号 MAILBOX 中断使能寄存器

打印系统 1 号 MAILBOX 中断使能寄存器，使能相应中断位。

寄存器地址：0x15103834。

表 3-54 打印系统 1 号 MAILBOX 中断使能寄存器

位域	名称	访问	缺省值	描述
31:0	mlb1_ipi_enable	RW	0x0	1 号 Mailbox 中断使能寄存器，使能对应中断位，高电平有效

3.4.48 打印系统 1 号 MAILBOX 中断触发寄存器

打印系统 1 号 MAILBOX 中断触发寄存器，触发相应中断位。

寄存器地址：0x15103838。

表 3-55 打印系统 1 号 MAILBOX 中断触发寄存器

位域	名称	访问	缺省值	描述
31:0	mlb1_ipi_set	W	-	1 号 Mailbox 中断触发寄存器，触发对应中断位，高电平有效。当相应中断使能位有效时，该位置 1 将触发相应中断，并将路由至对应处理器核中断位置位

3.4.49 打印系统 1 号 MAILBOX 中断清除寄存器

打印系统 1 号 MAILBOX 中断清除寄存器，清除相应中断状态位。

寄存器地址：0x1510383c。

表 3-56 打印系统 1 号 MAILBOX 中断清除寄存器

位域	名称	访问	缺省值	描述
31:0	mlb1_ipi_clear	W	-	1 号 Mailbox 中断清除寄存器，清除对应中断位，写 1 清除。对应位写 1 将清除相应中断状态位，取消相应中断触发，中断使能位不做清除

3.4.50 打印系统 1 号 MAILBOX 消息寄存器 0~7

打印系统 1 号 MAILBOX 消息寄存器 0~7，传递记录相应消息，可由用户自由定义。

寄存器地址：0x15103840~0x1510385c。

表 3-57 打印系统 1 号 MAILBOX 消息寄存器 0~7

位域	名称	访问	缺省值	描述
31:0	mlb1_message7	RW	0x0	1 号 Mailbox 消息寄存器 7，记录相应消息定义，可进行不同系统间消息传递
31:0	mlb1_message6	RW	0x0	1 号 Mailbox 消息寄存器 6，记录相应消息定义，可进行不同系统间消息传递
31:0	mlb1_message5	RW	0x0	1 号 Mailbox 消息寄存器 5，记录相应消息定义，可进行不同系统间消息传递
31:0	mlb1_message4	RW	0x0	1 号 Mailbox 消息寄存器 4，记录相应消息定义，可进行不同系统间消息传递
31:0	mlb1_message3	RW	0x0	1 号 Mailbox 消息寄存器 3，记录相应消息定义，可进行不同系统间消息传递
31:0	mlb1_message2	RW	0x0	1 号 Mailbox 消息寄存器 2，记录相应消息定义，可进行不同系统间消息传递
31:0	mlb1_message1	RW	0x0	1 号 Mailbox 消息寄存器 1，记录相应消息定义，可进行不同系统间消息传递
31:0	mlb1_message0	RW	0x0	1 号 Mailbox 消息寄存器 0，记录相应消息定义，可进行不同系统间消息传递

3.4.51 打印系统 2 号 MAILBOX 中断状态寄存器

打印系统 2 号 MAILBOX 中断状态寄存器，记录中断触发状态。

寄存器地址：0x15103860。

表 3-58 打印系统 2 号 MAILBOX 中断状态寄存器

位域	名称	访问	缺省值	描述
31:0	mlb2_ipi_status	RO	0x0	2 号 Mailbox 中断状态寄存器，记录被触发中断状态位，任何一位被置 1 且对应中断使能情况下，路由至对应处理器核中断位将被置位

3.4.52 打印系统 2 号 MAILBOX 中断使能寄存器

打印系统 2 号 MAILBOX 中断使能寄存器，使能相应中断位。

寄存器地址：0x15103864。

表 3-59 打印系统 2 号 MAILBOX 中断使能寄存器

位域	名称	访问	缺省值	描述
31:0	mlb2_ipi_enable	RW	0x0	2 号 Mailbox 中断使能寄存器，使能对应中断位，高电平有效

3.4.53 打印系统 2 号 MAILBOX 中断触发寄存器

打印系统 2 号 MAILBOX 中断触发寄存器，触发相应中断位。

寄存器地址：0x15103868。

表 3-60 打印系统 2 号 MAILBOX 中断触发寄存器

位域	名称	访问	缺省值	描述
31:0	mlb2_ipi_set	W	-	2 号 Mailbox 中断触发寄存器，触发对应中断位，高电平有效。当相应中断使能位有效时，该位置 1 将触发相应中断，并将路由至对应处理器核中断位置位

3.4.54 打印系统 2 号 MAILBOX 中断清除寄存器

打印系统 2 号 MAILBOX 中断清除寄存器，清除相应中断状态位。

寄存器地址：0x1510386c。

表 3-61 打印系统 2 号 MAILBOX 中断清除寄存器

位域	名称	访问	缺省值	描述
31:0	mlb2_ipi_clear	W	-	2 号 Mailbox 中断清除寄存器，清除对应中断位，写 1 清除。对应位写 1 将清除相应中断状态位，取消相应中断触发，中断使能位不做清除

3.4.55 打印系统 2 号 MAILBOX 消息寄存器 0~7

打印系统 2 号 MAILBOX 消息寄存器 0~7，传递记录相应消息，可由用户自定义。

寄存器地址：0x15103870~0x1510388c。

表 3-62 打印系统 0 号 MAILBOX 消息寄存器 0~7

位域	名称	访问	缺省值	描述
31:0	mlb2_message7	RW	0x0	2 号 Mailbox 消息寄存器 7，记录相应消息定义，可进行不同系统间消息传递
31:0	mlb2_message6	RW	0x0	2 号 Mailbox 消息寄存器 6，记录相应消息定义，可进行不同系统间消息传递
31:0	mlb2_message5	RW	0x0	2 号 Mailbox 消息寄存器 5，记录相应消息定义，可进行不同系统间消息传递
31:0	mlb2_message4	RW	0x0	2 号 Mailbox 消息寄存器 4，记录相应消息定义，可进行不同系统间消息传递
31:0	mlb2_message3	RW	0x0	2 号 Mailbox 消息寄存器 3，记录相应消息定义，可进行不同系统间消息传递
31:0	mlb2_message2	RW	0x0	2 号 Mailbox 消息寄存器 2，记录相应消息定义，可进行不同系统间消息传递
31:0	mlb2_message1	RW	0x0	2 号 Mailbox 消息寄存器 1，记录相应消息定义，可进行不同系统间消息传递
31:0	mlb2_message0	RW	0x0	2 号 Mailbox 消息寄存器 0，记录相应消息定义，可进行不同系统间消息传递

3.4.56 打印系统 MAILBOX 中断路由寄存器

打印系统 MAILBOX 中断路由寄存器，配置 0~2 号 MAILBOX 中断路由。

寄存器地址：0x15103890。

表 3-63 打印系统 MAILBOX 中断路由寄存器

位域	名称	访问	缺省值	描述
31:11	Reserved	RO	0x0	-
10:8	mlb2_ipi_map	RW	0x0	2 号 Mailbox 中断路由配置： 001:2 号 Mailbox 触发中断路由至主系统中断； 010:2 号 Mailbox 触发中断路由至打印系统中断； 100:2 号 Mailbox 触发中断路由至扫描系统中断

位域	名称	访问	缺省值	描述
7	Reserved	RO	0x0	–
6:4	mlb1_ipi_map	RW	0x0	1 号 Mailbox 中断路由配置： 001:1 号 Mailbox 触发中断路由至主系统中断； 010:1 号 Mailbox 触发中断路由至打印系统中断； 100:1 号 Mailbox 触发中断路由至扫描系统中断
3	Reserved	RO	0x0	–
2:0	mlb0_ipi_map	RW	0x0	0 号 Mailbox 中断路由配置： 001:0 号 Mailbox 触发中断路由至主系统中断； 010:0 号 Mailbox 触发中断路由至打印系统中断； 100:0 号 Mailbox 触发中断路由至扫描系统中断

三、扫描系统通用配置寄存器，主要包括以下寄存器说明：

3.4.57 扫描系统设备时钟门控寄存器

扫描系统各设备时钟门控寄存器，包括扫描系统各功能模块时钟门控的控制。

寄存器地址：0x15202030。

表 3-64 扫描系统设备时钟门控寄存器

位域	名称	访问	缺省值	描述
31:9	Reserved	RO	0x7ffffff	–
8	sc132_clk_ctrl	RW	0x1	扫描系统处理器核时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
7	Reserved	RO	–	–
6	scvid_clk_ctrl	RW	0x1	扫描系统 Scanner 模块时钟门控配置：
5	hpet1_clk_ctrl	RW	0x1	扫描系统 HPET1 模块时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
4	hpet0_clk_ctrl	RW	0x1	扫描系统 HPET0 模块时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
3	pmio_clk_ctrl	RW	0x1	扫描系统 PMIO 模块时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
2	spi_clk_ctrl	RW	0x1	扫描系统 SPI 模块时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
1	gpio_clk_ctrl	RW	0x1	扫描系统 GPIO 模块时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
0	uart_clk_ctrl	RW	0x1	扫描系统 UART 模块时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。

3.4.58 扫描系统设备复位控制寄存器

扫描系统各设备复位配置寄存器，包括扫描系统各功能模块软件复位的控制。

寄存器地址：0x15202034。

表 3-65 扫描系统设备复位控制寄存器

位域	名称	访问	缺省值	描述
31:9	Reserved	RO	0x7ffffff	–
8	sc132_rst_ctrl	RW	0x1	扫描系统处理器核复位软件配置： 1：模块复位无效；0：模块复位有效。
7	Reserved	RO	–	–
6	scvid_rst_ctrl	RW	0x1	扫描系统 Scanner 模块复位软件配置：
5	hpet1_rst_ctrl	RW	0x1	扫描系统 HPET1 模块复位软件配置： 1：模块复位无效；0：模块复位有效。
4	hpet0_rst_ctrl	RW	0x1	扫描系统 HPET0 模块复位软件配置： 1：模块复位无效；0：模块复位有效。

位域	名称	访问	缺省值	描述
3	pmio_rst_ctrl	RW	0x1	扫描系统 PMIO 模块复位软件配置： 1：模块复位无效；0：模块复位有效。
2	spi_rst_ctrl	RW	0x1	扫描系统 SPI 模块复位软件配置： 1：模块复位无效；0：模块复位有效。
1	gpio_rst_ctrl	RW	0x1	扫描系统 GPIO 模块复位软件配置： 1：模块复位无效；0：模块复位有效。
0	uart_rst_ctrl	RW	0x1	扫描系统 UART 模块复位软件配置： 1：模块复位无效；0：模块复位有效。

3.4.59 扫描系统 GPIO0~15 复用配置寄存器

扫描系统 GPIO0~15 复用配置寄存器。

寄存器地址：0x15202040。

表 3-66 扫描系统 GPIO0~15 复用配置寄存器

位域	名称	访问	缺省值	描述
31:30	GPIO15_MUX	RW	0x0	GPIO15 引脚复用配置： 00：复用为 GPIO15；01：第一复用； 10：第二复用； 11：引脚主功能。
29:28	GPIO14_MUX	RW	0x0	GPIO14 引脚复用配置： 00：复用为 GPIO14；01：第一复用； 10：第二复用； 11：引脚主功能。
27:26	GPIO13_MUX	RW	0x0	GPIO13 引脚复用配置： 00：复用为 GPIO13；01：第一复用； 10：第二复用； 11：引脚主功能。
25:24	GPIO12_MUX	RW	0x0	GPIO12 引脚复用配置： 00：复用为 GPIO12；01：第一复用； 10：第二复用； 11：引脚主功能。
23:22	GPIO11_MUX	RW	0x0	GPIO11 引脚复用配置： 00：复用为 GPIO12；01：第一复用； 10：第二复用； 11：引脚主功能。
21:20	GPIO10_MUX	RW	0x0	GPIO10 引脚复用配置： 00：复用为 GPIO10；01：第一复用； 10：第二复用； 11：引脚主功能。
19:18	GPIO09_MUX	RW	0x0	GPIO09 引脚复用配置： 00：复用为 GPIO09；01：第一复用； 10：第二复用； 11：引脚主功能。
17:16	GPIO08_MUX	RW	0x0	GPIO08 引脚复用配置： 00：复用为 GPIO08；01：第一复用； 10：第二复用； 11：引脚主功能。
15:14	GPIO07_MUX	RW	0x0	GPIO07 引脚复用配置： 00：复用为 GPIO07；01：第一复用； 10：第二复用； 11：引脚主功能。
13:12	GPIO06_MUX	RW	0x0	GPIO06 引脚复用配置： 00：复用为 GPIO06；01：第一复用； 10：第二复用； 11：引脚主功能。
11:10	GPIO05_MUX	RW	0x0	GPIO05 引脚复用配置： 00：复用为 GPIO05；01：第一复用； 10：第二复用； 11：引脚主功能。
9:8	GPIO04_MUX	RW	0x0	GPIO04 引脚复用配置： 00：复用为 GPIO04；01：第一复用； 10：第二复用； 11：引脚主功能。
7:6	GPIO03_MUX	RW	0x0	GPIO03 引脚复用配置： 00：复用为 GPIO03；01：第一复用； 10：第二复用； 11：引脚主功能。

位域	名称	访问	缺省值	描述
5:4	GPI002_MUX	RW	0x0	GPI002 引脚复用配置： 00: 复用为 GPI002; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
3:2	GPI001_MUX	RW	0x0	GPI001 引脚复用配置： 00: 复用为 GPI001; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
1:0	GPI000_MUX	RW	0x0	GPI000 引脚复用配置： 00: 复用为 GPI000; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。

3.4.60 扫描系统 GPI016~28 复用配置寄存器

扫描系统 GPI016~28 复用配置寄存器。

寄存器地址：0x15202044。

表 3-67 扫描系统 GPI016~28 复用配置寄存器

位域	名称	访问	缺省值	描述
31:26	Reserved	RO	-	-
25:24	GPI028_MUX	RW	0x0	GPI028 引脚复用配置： 00: 复用为 GPI028; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
23:22	GPI027_MUX	RW	0x0	GPI027 引脚复用配置： 00: 复用为 GPI027; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
21:20	GPI026_MUX	RW	0x0	GPI026 引脚复用配置： 00: 复用为 GPI026; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
19:18	GPI025_MUX	RW	0x0	GPI025 引脚复用配置： 00: 复用为 GPI025; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
17:16	GPI024_MUX	RW	0x0	GPI024 引脚复用配置： 00: 复用为 GPI024; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
15:14	GPI023_MUX	RW	0x0	GPI023 引脚复用配置： 00: 复用为 GPI023; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
13:12	GPI022_MUX	RW	0x0	GPI022 引脚复用配置： 00: 复用为 GPI022; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
11:10	GPI021_MUX	RW	0x0	GPI021 引脚复用配置： 00: 复用为 GPI021; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
9:8	GPI020_MUX	RW	0x0	GPI020 引脚复用配置： 00: 复用为 GPI020; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
7:6	GPI019_MUX	RW	0x0	GPI019 引脚复用配置： 00: 复用为 GPI019; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
5:4	GPI018_MUX	RW	0x0	GPI018 引脚复用配置： 00: 复用为 GPI018; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
3:2	GPI017_MUX	RW	0x0	GPI017 引脚复用配置： 00: 复用为 GPI017; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
1:0	GPI016_MUX	RW	0x0	GPI016 引脚复用配置： 00: 复用为 GPI016; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。

3.4.61 扫描系统通用配置寄存器

扫描系统通用配置寄存器 0，包括 UART 引脚复用模式、GMAC、USB、内存控制器的配置等。

寄存器地址：0x15202070。

表 3-68 扫描系统通用配置寄存器 0

位域	名称	访问	缺省值	描述
31:24	Reserved	RO	0x0	–
23:16	hpet_clkdiv	RW	0x0	扫描系统 hpet0/1 时钟分频配置系统：1~255
8	pmio_extin_sel	RW	0x0	扫描系统 PMIO 模块 EXTIN[15:0] 信号输入选择： 0：由 PRT_PMIO_INTCON[31:16] 端口输入； 1：由 PRT_PMIO_INTCON[31:24]、 SCA_CISCLK[6:0]、SCA_AFECLK 依次输入。
7	hpet1_int_ctrl	RW	0x0	扫描系统 hpet1 中断输出模式配置： 0：单中断输出模式(内部 3 个计数器共享 1 个中断)； 1：三中断输出模式(内部 3 个计数器独立 3 个中断)
6	hpet0_int_ctrl	RW	0x0	扫描系统 hpet0 中断输出模式配置： 0：单中断输出模式(内部 3 个计数器共享 1 个中断)； 1：三中断输出模式(内部 3 个计数器独立 3 个中断)
5	scl32_nmi	RW	0x0	扫描系统处理器核 NMI 输入信号，高电平有效
4:0	Reserved	RO	0x0	–

3.4.62 扫描系统采样参数寄存器

扫描系统采样参数寄存器，包括扫描系统启动配置及其他系统休眠相关信息。

寄存器地址：0x15202078。

表 3-69 扫描系统采样参数寄存器

位域	名称	访问	缺省值	描述
31:9	Reserved	RO	0x0	–
9	scl32_sleeping_sta	RO	0x0	扫描系统处理器核休眠状态监测： 0=WORKON 状态 1=SLEEPING 状态
8	qspi_pad_type	RO	0x0	QSPI 引脚电平类型 0 = 3.3V IO 1 = 1.8V IO
7	emmc_pad1_type	RO	0x0	eMMC (8 线模式) 引脚电平类型 0 = 3.3V IO 1 = 1.8V IO
6	emmc_pad0_type	RO	0x0	eMMC (4 线模式) 引脚电平类型 0 = 3.3V IO 1 = 1.8V IO
5	usbrefclk_mode	RO	0x0	OTG 参考时钟模式输入 0=内部参考时钟输入 1=外部参考时钟输入
4	sdio1_mode	RO	0x0	SDIO1 模式配置输入 0=SDIO 模式 1=eMMC 模式

位域	名称	访问	缺省值	描述
3	sdio0_mode	RO	0x0	SDIO0 模式配置输入 0=SDIO 模式 1=eMMC 模式
2:1	clk_sel	RO	0x0	芯片内部 PLL 输出时钟上电配置选择： 00：硬件低频时钟配置模式，PLL 按照低频配置参数输出时钟； 01：硬件高频时钟配置模式，PLL 按照高频配置参数输出时钟； 10：软件配置模式，PLL 按照软件配置选择输出时钟； 11：硬件 bypass 模式，PLL 输出时钟全部使用外部输入系统时钟。
0	boot_sel	RO	0x0	芯片启动选择方式： 0：SPI0 启动；1：SDIO0/eMMC0 启动。

3.4.63 扫描系统高精度计数器

扫描系统计数寄存器，32 位高精度时钟计数器，工作频率为内部总线时钟频率。

寄存器地址：0x1520207c。

表 3-70 扫描系统高精度计数器

位域	名称	访问	缺省值	描述
31:0	SCACHIP_HPT	RW	0x0	32 位高精度时钟计数器

3.4.64 扫描系统启动入口配置寄存器

扫描系统启动入口配置寄存器，配置扫描系统处理器核启动地址。

寄存器地址：0x15202100。

表 3-71 扫描系统启动入口配置寄存器

位域	名称	访问	缺省值	描述
31:0	scl32_boot_entry	RW	0x1c000000	配置扫描系统处理器核启动入口地址， bit[0]为地址配置有效使能位，高电平有效： 1-入口地址配置有效；0-入口地址配置无效，采用默认启动入口地址(0x1c000000)

3.4.65 扫描系统 0 号 MAILBOX 中断状态寄存器

扫描系统 0 号 MAILBOX 中断状态寄存器，记录中断触发状态。

寄存器地址：0x15202800。

表 3-72 扫描系统 0 号 MAILBOX 中断状态寄存器

位域	名称	访问	缺省值	描述
31:0	mlb0_ipi_status	RO	0x0	0 号 Mailbox 中断状态寄存器，记录被触发中断状态位，任何一位被置 1 且对应中断使能情况下，路由至对应处理器核中断位将被置位

3.4.66 扫描系统 0 号 MAILBOX 中断使能寄存器

扫描系统 0 号 MAILBOX 中断使能寄存器，使能相应中断位。

寄存器地址：0x15202804。

表 3-73 扫描系统 0 号 MAILBOX 中断使能寄存器

位域	名称	访问	缺省值	描述
31:0	mlb0_ipi_enable	RW	0x0	0 号 Mailbox 中断使能寄存器，使能对应中断位，高电平有效

3.4.67 扫描系统 0 号 MAILBOX 中断触发寄存器

扫描系统 0 号 MAILBOX 中断触发寄存器，触发相应中断位。

寄存器地址：0x15202808。

表 3-74 扫描系统 0 号 MAILBOX 中断触发寄存器

位域	名称	访问	缺省值	描述
31:0	mlb0_ipi_set	W	-	0 号 Mailbox 中断触发寄存器，触发对应中断位，高电平有效。当相应中断使能位有效时，该位置 1 将触发相应中断，并将路由至对应处理器核中断位置

3.4.68 扫描系统 0 号 MAILBOX 中断清除寄存器

扫描系统 0 号 MAILBOX 中断清除寄存器，清除相应中断状态位。

寄存器地址：0x1520280c。

表 3-75 扫描系统 0 号 MAILBOX 中断清除寄存器

位域	名称	访问	缺省值	描述
31:0	mlb0_ipi_clear	W	-	0 号 Mailbox 中断清除寄存器，清除对应中断位，写 1 清除。对应位写 1 将清除相应中断状态位，取消相应中断触发，中断使能位不做清除

3.4.69 扫描系统 0 号 MAILBOX 消息寄存器 0~7

扫描系统 0 号 MAILBOX 消息寄存器 0~7，传递记录相应消息，可由用户自由定义。

寄存器地址：0x15202810~0x1520282c。

表 3-76 扫描系统 0 号 MAILBOX 消息寄存器 0~7

位域	名称	访问	缺省值	描述
31:0	mlb0_message7	RW	0x0	0 号 Mailbox 消息寄存器 7，记录相应消息定义，可进行不同系统间消息传递
31:0	mlb0_message6	RW	0x0	0 号 Mailbox 消息寄存器 6，记录相应消息定义，可进行不同系统间消息传递
31:0	mlb0_message5	RW	0x0	0 号 Mailbox 消息寄存器 5，记录相应消息定义，可进行不同系统间消息传递
31:0	mlb0_message4	RW	0x0	0 号 Mailbox 消息寄存器 4，记录相应消息定义，可进行不同系统间消息传递
31:0	mlb0_message3	RW	0x0	0 号 Mailbox 消息寄存器 3，记录相应消息定义，可进行不同系统间消息传递
31:0	mlb0_message2	RW	0x0	0 号 Mailbox 消息寄存器 2，记录相应消息定义，可进行不同系统间消息传递
31:0	mlb0_message1	RW	0x0	0 号 Mailbox 消息寄存器 1，记录相应消息定义，可进行不同系统间消息传递
31:0	mlb0_message0	RW	0x0	0 号 Mailbox 消息寄存器 0，记录相应消息定义，可进行不同系统间消息传递

3.4.70 扫描系统 1 号 MAILBOX 中断状态寄存器

扫描系统 1 号 MAILBOX 中断状态寄存器，记录中断触发状态。

寄存器地址：0x15202830。

表 3-77 扫描系统 1 号 MAILBOX 中断状态寄存器

位域	名称	访问	缺省值	描述
31:0	mlbl_ipi_status	RO	0x0	1 号 Mailbox 中断状态寄存器，记录被触发中断状态位，任何一位被置 1 且对应中断使能情况下，路由至对应处理器核中断位将被置位

3.4.71 扫描系统 1 号 MAILBOX 中断使能寄存器

扫描系统 1 号 MAILBOX 中断使能寄存器，使能相应中断位。

寄存器地址：0x15202834。

表 3-78 扫描系统 1 号 MAILBOX 中断使能寄存器

位域	名称	访问	缺省值	描述
31:0	mlbl_ipi_enable	RW	0x0	1 号 Mailbox 中断使能寄存器，使能对应中断位，高电平有效

3.4.72 扫描系统 1 号 MAILBOX 中断触发寄存器

扫描系统 1 号 MAILBOX 中断触发寄存器，触发相应中断位。

寄存器地址：0x15202838。

表 3-79 扫描系统 1 号 MAILBOX 中断触发寄存器

位域	名称	访问	缺省值	描述
31:0	mlbl_ipi_set	W	-	1 号 Mailbox 中断触发寄存器，触发对应中断位，高电平有效。当相应中断使能位有效时，该位置 1 将触发相应中断，并将路由至对应处理器核中断位置位

3.4.73 扫描系统 1 号 MAILBOX 中断清除寄存器

扫描系统 1 号 MAILBOX 中断清除寄存器，清除相应中断状态位。

寄存器地址：0x1520283c。

表 3-80 扫描系统 1 号 MAILBOX 中断清除寄存器

位域	名称	访问	缺省值	描述
31:0	mlbl_ipi_clear	W	-	1 号 Mailbox 中断清除寄存器，清除对应中断位，写 1 清除。对应位写 1 将清除相应中断状态位，取消相应中断触发，中断使能位不做清除

3.4.74 扫描系统 1 号 MAILBOX 消息寄存器 0~7

扫描系统 1 号 MAILBOX 消息寄存器 0~7，传递记录相应消息，可由用户自由定义。

寄存器地址：0x15202840~0x1520285c。

表 3-81 扫描系统 1 号 MAILBOX 消息寄存器 0~7

位域	名称	访问	缺省值	描述
31:0	mlbl_message7	RW	0x0	1 号 Mailbox 消息寄存器 7，记录相应消息定义，可进行不同系统间消息传递
31:0	mlbl_message6	RW	0x0	1 号 Mailbox 消息寄存器 6，记录相应消息定义，可进行不同系统间消息传递
31:0	mlbl_message5	RW	0x0	1 号 Mailbox 消息寄存器 5，记录相应消息定义，可进行不同系统间消息传递
31:0	mlbl_message4	RW	0x0	1 号 Mailbox 消息寄存器 4，记录相应消息定义，可进行不同系统间消息传递

位域	名称	访问	缺省值	描述
31:0	mlb1_message3	RW	0x0	1 号 Mailbox 消息寄存器 3，记录相应消息定义，可进行不同系统间消息传递
31:0	mlb1_message2	RW	0x0	1 号 Mailbox 消息寄存器 2，记录相应消息定义，可进行不同系统间消息传递
31:0	mlb1_message1	RW	0x0	1 号 Mailbox 消息寄存器 1，记录相应消息定义，可进行不同系统间消息传递
31:0	mlb1_message0	RW	0x0	1 号 Mailbox 消息寄存器 0，记录相应消息定义，可进行不同系统间消息传递

3.4.75 扫描系统 2 号 MAILBOX 中断状态寄存器

扫描系统 2 号 MAILBOX 中断状态寄存器，记录中断触发状态。

寄存器地址：0x15202860。

表 3-82 扫描系统 2 号 MAILBOX 中断状态寄存器

位域	名称	访问	缺省值	描述
31:0	mlb2_ipi_status	RO	0x0	2 号 Mailbox 中断状态寄存器，记录被触发中断状态位，任何一位被置 1 且对应中断使能情况下，路由至对应处理器核中断位将被置位

3.4.76 扫描系统 2 号 MAILBOX 中断使能寄存器

扫描系统 2 号 MAILBOX 中断使能寄存器，使能相应中断位。

寄存器地址：0x15202864。

表 3-83 扫描系统 2 号 MAILBOX 中断使能寄存器

位域	名称	访问	缺省值	描述
31:0	mlb2_ipi_enable	RW	0x0	2 号 Mailbox 中断使能寄存器，使能对应中断位，高电平有效

3.4.77 扫描系统 2 号 MAILBOX 中断触发寄存器

扫描系统 2 号 MAILBOX 中断触发寄存器，触发相应中断位。

寄存器地址：0x15202868。

表 3-84 扫描系统 2 号 MAILBOX 中断触发寄存器

位域	名称	访问	缺省值	描述
31:0	mlb2_ipi_set	W	-	2 号 Mailbox 中断触发寄存器，触发对应中断位，高电平有效。当相应中断使能位有效时，该位置 1 将触发相应中断，并将路由至对应处理器核中断位置位

3.4.78 扫描系统 2 号 MAILBOX 中断清除寄存器

扫描系统 2 号 MAILBOX 中断清除寄存器，清除相应中断状态位。

寄存器地址：0x1520286c。

表 3-85 扫描系统 2 号 MAILBOX 中断清除寄存器

位域	名称	访问	缺省值	描述
31:0	mlb2_ipi_clear	W	-	2 号 Mailbox 中断清除寄存器，清除对应中断位，写 1 清除。对应位写 1 将清除相应中断状态位，取消相应中断触发，中断使能位不做清除

3.4.79 扫描系统 2 号 MAILBOX 消息寄存器 0~7

扫描系统 2 号 MAILBOX 消息寄存器 0~7，传递记录相应消息，可由用户自由定义。

寄存器地址：0x15202870~0x1520288c。

表 3-86 扫描系统 0 号 MAILBOX 消息寄存器 0~7

位域	名称	访问	缺省值	描述
31:0	mlb2_message7	RW	0x0	2 号 Mailbox 消息寄存器 7，记录相应消息定义，可进行不同系统间消息传递
31:0	mlb2_message6	RW	0x0	2 号 Mailbox 消息寄存器 6，记录相应消息定义，可进行不同系统间消息传递
31:0	mlb2_message5	RW	0x0	2 号 Mailbox 消息寄存器 5，记录相应消息定义，可进行不同系统间消息传递
31:0	mlb2_message4	RW	0x0	2 号 Mailbox 消息寄存器 4，记录相应消息定义，可进行不同系统间消息传递
31:0	mlb2_message3	RW	0x0	2 号 Mailbox 消息寄存器 3，记录相应消息定义，可进行不同系统间消息传递
31:0	mlb2_message2	RW	0x0	2 号 Mailbox 消息寄存器 2，记录相应消息定义，可进行不同系统间消息传递
31:0	mlb2_message1	RW	0x0	2 号 Mailbox 消息寄存器 1，记录相应消息定义，可进行不同系统间消息传递
31:0	mlb2_message0	RW	0x0	2 号 Mailbox 消息寄存器 0，记录相应消息定义，可进行不同系统间消息传递

3.4.80 扫描系统 MAILBOX 中断路由寄存器

扫描系统 MAILBOX 中断路由寄存器，配置 0~2 号 MAILBOX 中断路由。

寄存器地址：0x15202890。

表 3-87 打印系统 MAILBOX 中断路由寄存器

位域	名称	访问	缺省值	描述
31:11	Reserved	RO	0x0	—
10:8	mlb2_ipi_map	RW	0x0	2 号 Mailbox 中断路由配置： 001:2 号 Mailbox 触发中断路由至主系统中断； 010:2 号 Mailbox 触发中断路由至打印系统中断； 100:2 号 Mailbox 触发中断路由至扫描系统中断
7	Reserved	RO	0x0	—
6:4	mlb1_ipi_map	RW	0x0	1 号 Mailbox 中断路由配置： 001:1 号 Mailbox 触发中断路由至主系统中断； 010:1 号 Mailbox 触发中断路由至打印系统中断； 100:1 号 Mailbox 触发中断路由至扫描系统中断
3	Reserved	RO	0x0	—
2:0	mlb0_ipi_map	RW	0x0	0 号 Mailbox 中断路由配置： 001:0 号 Mailbox 触发中断路由至主系统中断； 010:0 号 Mailbox 触发中断路由至打印系统中断； 100:0 号 Mailbox 触发中断路由至扫描系统中断

3.5 中断配置及路由

龙芯 2P0300 芯片中断根据不同系统应用需求，分为主系统、打印系统、扫描系统中断。

3.5.1 各系统中断概述

其中，主系统分为传统、扩展中断两种方式：

一、主系统传统中断方式

传统中断方式最多支持 64 个中断源，以统一方式进行管理，如下图所示，任意一个 IO 中断源可以被配置为是否使能、触发的方式、以及被路由的目标处理器核中断脚。

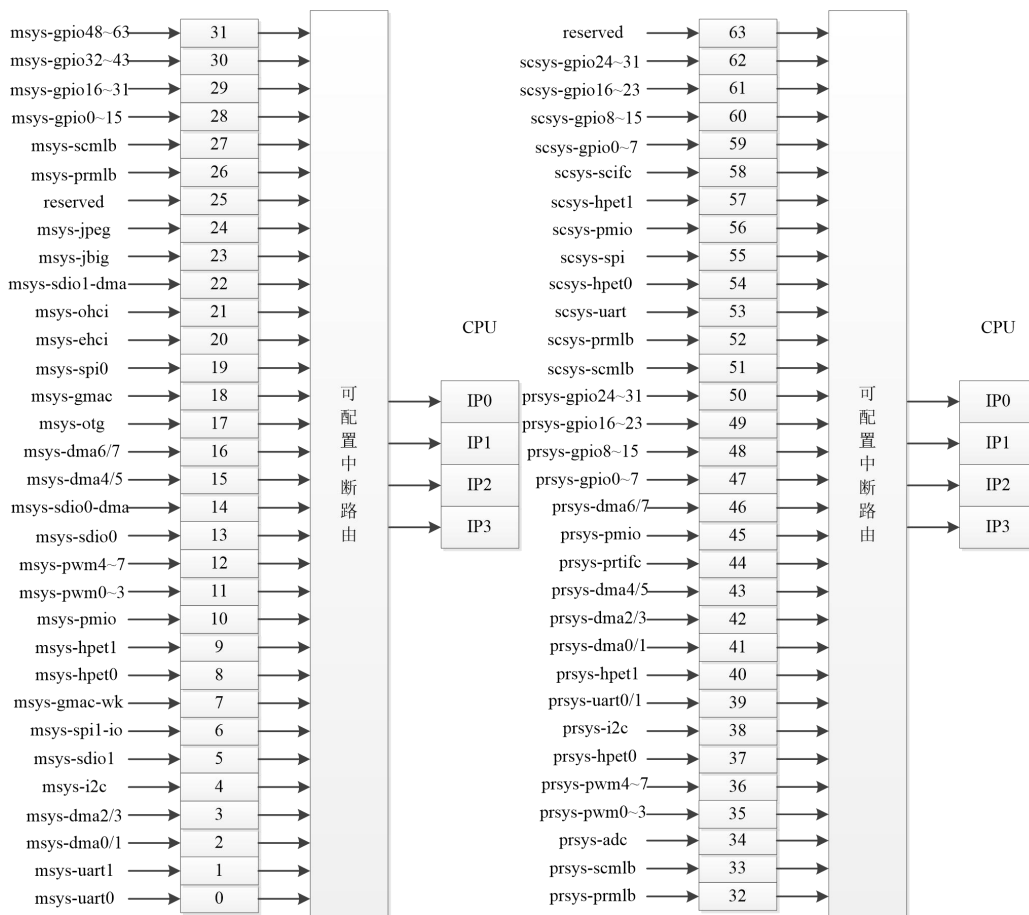


图3-2 主系统传统 IO 中断路由示意图

传统中断相关配置寄存器都是以位的形式对相应的中断线进行控制，中断控制位连接及属性配置见表 3-88。中断使能（Enable）的配置有三个寄存器：Intenset、Intenclr 和 Inten。Intenset 设置中断使能，Intenset 寄存器写 1 的位对应的中断位被使能。Intenclr 清除中断使能，Intenclr 寄存器写 1 的位对应的中断状态和中断使能同时被清除。Inten 中断使能寄存器(只读)，读取当前各中断使能的情况。脉冲形式的中断信号由 Intedge 配置寄存器来选择，写 1 表示脉冲触发，写 0 表示电平触发。Intpol 中断极性寄存器，电平触发模式下，写 0 对应中断源为高电平触发中断，写 1 对应中断源为低电平触发中断；边沿触发模式下，写 0 对应中断源为上升沿触发中断，写 1 对应中断源为下降沿触发中断。中断处理程序可以通过 Intenclr 的相应位来清除脉冲记录,在中断被清除后，由于中断使能位同时被清除，需要重新配置相应的 Intenset，使能相应中断位才能采集到该中断的下一中断触发。

表 3-88 主系统传统 IO 中断寄存器列表

地址	名称	描述
0x14001040	CORE_INTISRO	路由给 CORE 的低 32 位中断状态
0x14001044	INTISRO	低 32 位中断状态寄存器
0x14001048	CORE_INTISR1	路由给 CORE 的高 32 位中断状态
0x1400104c	INTISR1	高 32 位中断状态寄存器
0x14001400	ENTRY0_0	8 位中断路由寄存器[0--7]
0x14001408	ENTRY8_0	8 位中断路由寄存器[8--15]
0x14001410	ENTRY16_0	8 位中断路由寄存器[16--23]
0x14001418	ENTRY24_0	8 位中断路由寄存器[24--31]
0x14001420	INTISR_0	低 32 位中断状态寄存器
0x14001424	INTIEN_0	低 32 位中断使能状态寄存器
0x14001428	INTSET_0	低 32 位设置使能寄存器
0x1400142c	INTCLR_0	低 32 位中断清除寄存器，清除使能寄存器和脉冲触发的中断
0x14001430	INTPOL_0	低 32 位极性设置寄存器(电平中断)
0x14001434	INTEDGE_0	低 32 位触发方式寄存器（1：脉冲触发；0：电平触发）
0x14001440	ENTRY0_1	8 位中断路由寄存器[32--39]
0x14001448	ENTRY8_1	8 位中断路由寄存器[40--47]
0x14001450	ENTRY16_1	8 位中断路由寄存器[48--55]
0x14001458	ENTRY24_1	8 位中断路由寄存器[56--63]
0x14001460	INTISR_1	高 32 位中断状态寄存器
0x14001464	INTIEN_1	高 32 位中断使能状态寄存器
0x14001468	INTSET_1	高 32 位设置使能寄存器
0x1400146c	INTCLR_1	高 32 位中断清除寄存器，清除使能寄存器和脉冲触发的中断
0x14001470	INTPOL_1	高 32 位极性设置寄存器(电平中断)
0x14001474	INTEDGE_1	高 32 位触发方式寄存器（1：脉冲触发；0：电平触发）

二、主系统扩展中断方式

龙芯 2P0300 主系统除了支持上述传统 IO 中断方式外，还增加了扩展 IO 中断方式，即将 2P0300 芯片中所有 IO 设备中断，全部映射至全新的扩展 IO 中断向量，并增加相应扩展中断使能、中断状态、中断清除及路由等功能。该扩展 IO 中断向量最多支持 128 个 IO 设备，IO 设备与扩展中断号对应关系如下图所示，该扩展中断可独立于传统 IO 中断处理方式之外，填补原传统 IO 中断方式仅 64 个有限中断源的处理限制，提升 IO 中断使用的灵活性。

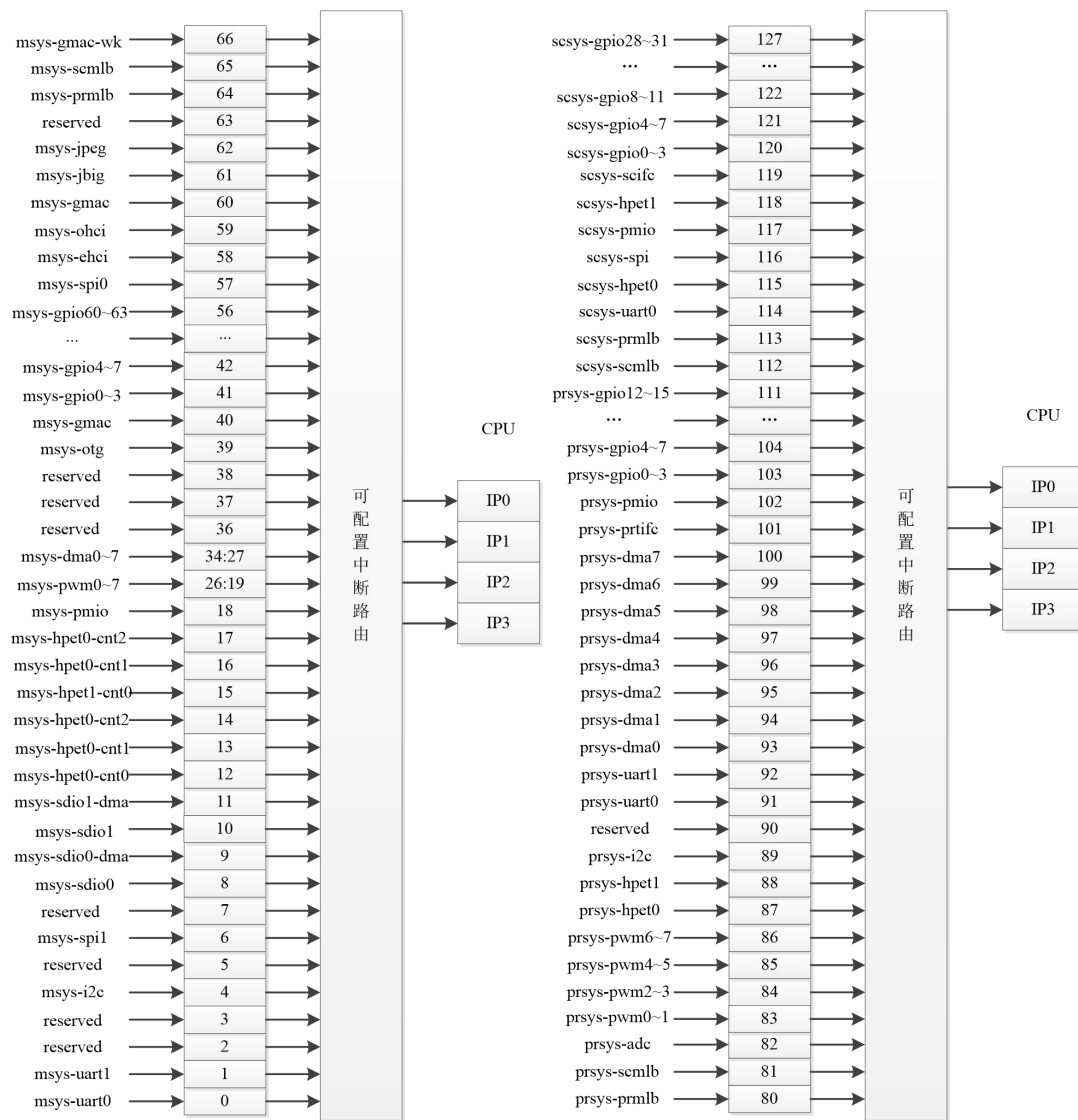


图3-3 主系统扩展 IO 中断路由示意图

2P0300 芯片主系统在使用扩展 IO 中断前，需要使能“主系统通用配置寄存器 0”中的扩展 IO 中断使能位。该寄存器配置地址为 0x14000100，对应配置位如下表：

位域	名称	访问	缺省值	描述
15	extioi_enable	RW	0x0	扩展中断使能控制位： 1：开启扩展中断(传统/扩展中断同时有效)； 0：关闭扩展中断(仅传统中断有效)

扩展 IO 中断相关配置寄存器与传统中断寄存器类似，都是以位的形式对相应的中断线进行控制，中断控制位连接及属性配置见表 3-89。扩展中断配置寄存器主要有三种寄存器：扩展中断使能 EXTINT_IEN、扩展中断状态 EXTINT_ISR 和扩展中断清除 EXTINT_ICLR。EXTINT_IEN 设置扩展中断使能，相应寄存器写 1 的位对应的中断被使能。EXTINT_ISR 扩展中断状态，寄存器相应位为 1 表示对应 IO 中断有效(注：该状态不依赖于中断使能位是否被置起)。EXTINT_ICLR 扩展中断清除寄存器，与扩展中断状态寄存器共用，对应位写 1 清除对应中断状态。处理器核增加扩展中断状态 CORE_EXTISR 和中断清除寄存器 CORE_EXTICLR，CORE_EXTISR 处理器核中断状态寄存器，当对应扩展 IO 中断使能位有效且 IO 中断有效时，该寄存器对应中断状态位被置起，表示对应 IO 中断可被处理器核接收。CORE_EXTICLR 中断清除寄存器与状态寄存器共用，对应位写 1 清除对应中断状态。

此外，扩展 IO 中断增加路由配置寄存器 EXTINT_MAP，该寄存器与传统 IO 中断路由类似，可将所有扩展 IO 中断按组分类路由至处理器核中断引脚向量，对应 CP0_Status 寄存器的 IP0 到 IP3。

表 3-89 主系统 IO 扩展中断寄存器列表

地址	名称	描述
0x14001148	EXTIOI_ACK	扩展中断设备反馈寄存器
0x140014c0	EXTIOI_MAP	扩展中断设备路由寄存器
0x14001600	EXTIOI_IEN0	扩展中断设备使能寄存器 0
0x14001604	EXTIOI_IEN1	扩展中断设备使能寄存器 1
0x14001608	EXTIOI_IEN2	扩展中断设备使能寄存器 2
0x1400160c	EXTIOI_IEN3	扩展中断设备使能寄存器 3
0x14001640	EXTIOI_POLO	扩展中断电平配置寄存器 0
0x14001644	EXTIOI_POL1	扩展中断电平配置寄存器 1
0x14001648	EXTIOI_POL2	扩展中断电平配置寄存器 2
0x1400164c	EXTIOI_POL3	扩展中断电平配置寄存器 3
0x14001700	EXTIOI_ISR0/ICLR0	扩展中断状态/清除寄存器 0
0x14001704	EXTIOI_ISR1/ICLR1	扩展中断状态/清除寄存器 1
0x14001708	EXTIOI_ISR2/ICLR2	扩展中断状态/清除寄存器 2
0x1400170c	EXTIOI_ISR3/ICLR3	扩展中断状态/清除寄存器 3
0x14001800	EXTIOI_CORE_ISR0/ICLR0	路由至 CORE 扩展中断状态/清除寄存器 0
0x14001804	EXTIOI_CORE_ISR1/ICLR1	路由至 CORE 扩展中断状态/清除寄存器 1
0x14001808	EXTIOI_CORE_ISR2/ICLR2	路由至 CORE 扩展中断状态/清除寄存器 2
0x1400180c	EXTIOI_CORE_ISR3/ICLR3	路由至 CORE 扩展中断状态/清除寄存器 3

三、打印系统中断方式

其中，打印系统中断方式最多支持 96 个中断源，以统一方式进行管理，如下图所示，任意一个 IO 中断源可以被配置为是否使能、触发的方式、以及被路由的目标处理器核中断脚。

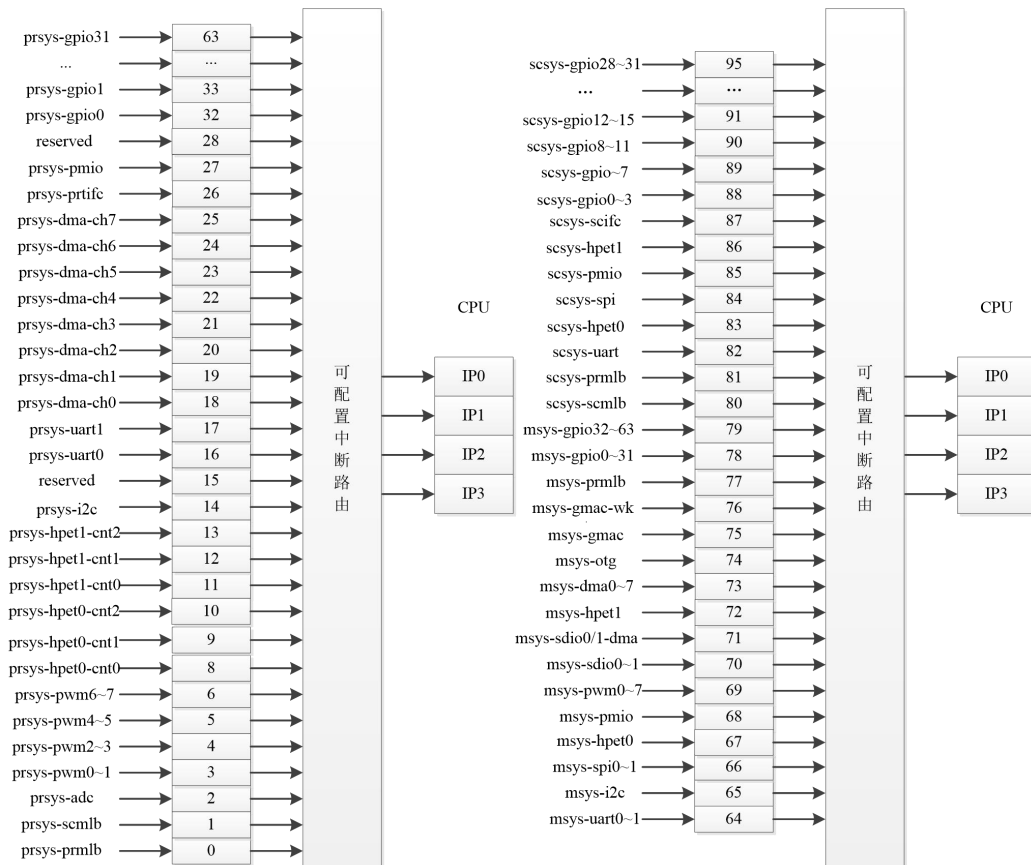


图3-4 打印系统 IO 中断路由示意图

打印系统中断相关配置寄存器都是以位的形式对相应的中断线进行控制，中断控制位连接及属性配置见表 3-90。中断使能（Enable）的配置 INT_IEN 寄存器，INT_IEN 设置中断使能，INT_IEN 寄存器写 1 的位对应的中断位被使能。中断清除 INT_CLR 寄存器，清除中断状态位，INT_CLR 寄存器写 1 的位对应的中断状态被清除（注：仅针对边沿触发形式，且中断使能位不清除）。脉冲形式的中断信号由 INT_EDGE 配置寄存器来选择，写 1 表示脉冲触发，写 0 表示电平触发。当中断触发配置为脉冲形式后，中断电平配置 INT_POL 寄存器可选择相应触发沿，0 为下降沿触发，1 为上升沿触发；中断双沿 INT_DUAL 寄存器可选择是否上升沿、下降沿均触发中断，高电平有效。当中断触发配置为电平触发形式时，中断电平 INT_POL 可选择触发电平形式，0 为低电平触发，1 为高电平触发。中断处理程序可以通过 INT_CLR 的相应位来清除脉冲中断记录，在中断被清除后，不需要重新使能相应中断位，即可采集到该中断位的下一次脉冲中断触发。

表 3-90 打印系统 IO 中断寄存器列表

地址	名称	描述
0x15103400	PRT_INT_ISR0	打印设备中断状态 0
0x15103404	PRT_INT_IEN0	打印设备中断使能配置 0
0x1510340c	PRT_INT_CLR0	打印设备中断清除配置 0
0x15103410	PRT_INT_POL0	打印设备中断电平配置 0
0x15103414	PRT_INT_EDGE0	打印设备中断边沿配置 0
0x15103418	PRT_INT_DUAL0	打印设备中断双沿配置 0
0x15103420	PRT_INT_ISR1	打印设备中断状态 1
0x15103424	PRT_INT_IEN1	打印设备中断使能配置 1
0x1510342c	PRT_INT_CLR1	打印设备中断清除配置 1
0x15103430	PRT_INT_POL1	打印设备中断电平配置 1
0x15103434	PRT_INT_EDGE1	打印设备中断边沿配置 1
0x15103438	PRT_INT_DUAL1	打印设备中断双沿配置 1
0x15103440	PRT_INT_ISR2	打印设备中断状态 2
0x15103444	PRT_INT_IEN2	打印设备中断使能配置 2
0x1510344c	PRT_INT_CLR2	打印设备中断清除配置 2
0x15103450	PRT_INT_POL2	打印设备中断电平配置 2
0x15103454	PRT_INT_EDGE2	打印设备中断边沿配置 2
0x15103458	PRT_INT_DUAL2	打印设备中断双沿配置 2
0x15103460	PRT_INT_MAP0	打印设备中断路由配置 0
0x15103464	PRT_INT_MAP1	打印设备中断路由配置 1
0x15103468	PRT_INT_MAP2	打印设备中断路由配置 2
0x1510346c	PRT_INT_MAP3	打印设备中断路由配置 3
0x15103470	PRT_INT_MAP4	打印设备中断路由配置 4
0x15103474	PRT_INT_MAP5	打印设备中断路由配置 5

四、扫描系统中断方式

其中，扫描系统中断方式最多支持 96 个中断源，以统一方式进行管理，如下图所示，任意一个 IO 中断源可以被配置为是否使能、触发的方式、以及被路由的目标处理器核中断脚。

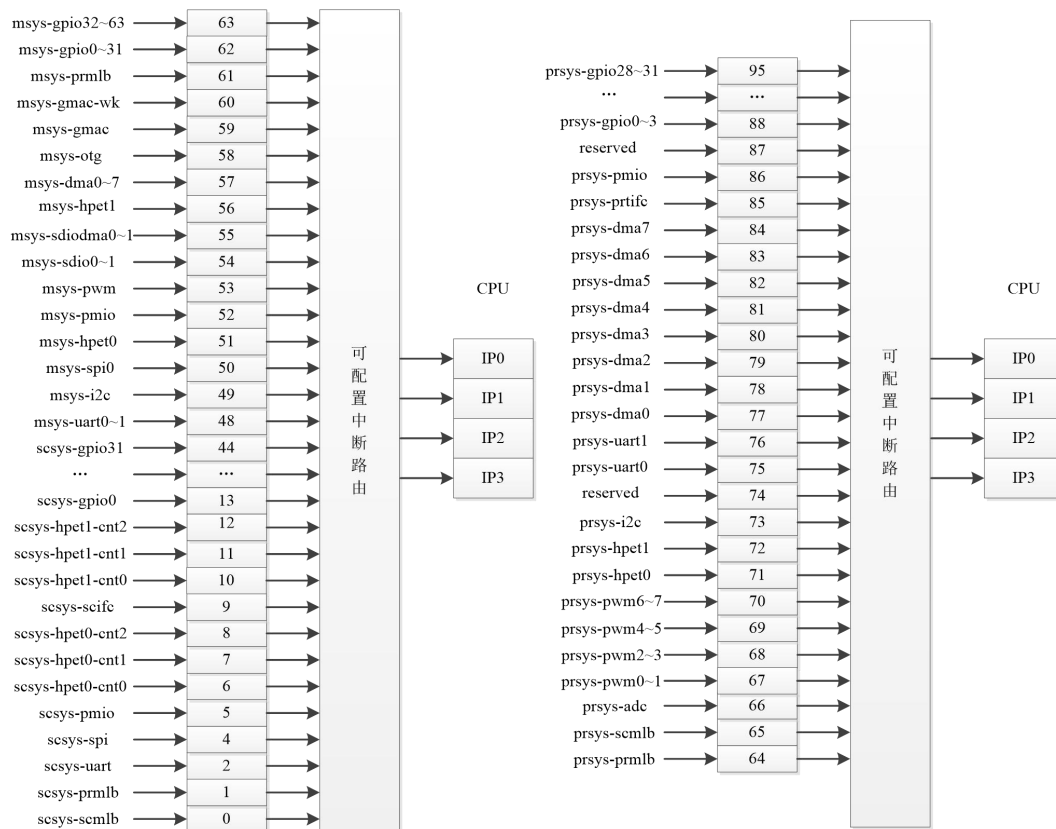


图3-5 扫描系统 IO 中断路由示意图

扫描系统中断相关配置寄存器都是以位的形式对相应的中断线进行控制，中断控制位连接及属性配置见表 3-91。中断使能（Enable）的配置 INT_IEN 寄存器，INT_IEN 设置中断使能，INT_IEN 寄存器写 1 的位对应的中断位被使能。中断清除 INT_CLR 寄存器，清除中断状态位，INT_CLR 寄存器写 1 的位对应的中断状态被清除（注：仅针对边沿触发形式，且中断使能位不清除）。脉冲形式的中断信号由 INT_EDGE 配置寄存器来选择，写 1 表示脉冲触发，写 0 表示电平触发。当中断触发配置为脉冲形式后，中断电平配置 INT_POL 寄存器可选择相应触发沿，0 为下降沿触发，1 为上升沿触发；中断双沿 INT_DUAL 寄存器可选择是否上升沿、下降沿均触发中断，高电平有效。当中断触发配置为电平触发形式时，中断电平 INT_POL 可选择触发电平形式，0 为低电平触发，1 为高电平触发。中断处理程序可以通过 INT_CLR 的相应位来清除脉冲中断记录，在中断被清除后，不需要重新使能相应中断位，即可采集到该中断位的下一次脉冲中断触发。

表 3-91 扫描系统 IO 中断寄存器列表

地址	名称	描述
0x15202400	SCA_INT_ISR0	扫描设备中断状态 0
0x15202404	SCA_INT_IEN0	扫描设备中断使能配置 0
0x1520240c	SCA_INT_CLR0	扫描设备中断清除配置 0
0x15202410	SCA_INT_POLO	扫描设备中断电平配置 0
0x15202414	SCA_INT_EDGE0	扫描设备中断边沿配置 0
0x15202418	SCA_INT_DUAL0	扫描设备中断双沿配置 0
0x15202420	SCA_INT_ISR1	扫描设备中断状态 1
0x15202424	SCA_INT_IEN1	扫描设备中断使能配置 1
0x1520242c	SCA_INT_CLR1	扫描设备中断清除配置 1
0x15202430	SCA_INT_POL1	扫描设备中断电平配置 1
0x15202434	SCA_INT_EDGE1	扫描设备中断边沿配置 1
0x15202438	SCA_INT_DUAL1	扫描设备中断双沿配置 1
0x15202440	SCA_INT_ISR2	扫描设备中断状态 2
0x15202444	SCA_INT_IEN2	扫描设备中断使能配置 2
0x1520244c	SCA_INT_CLR2	扫描设备中断清除配置 2
0x15202450	SCA_INT_POL2	扫描设备中断电平配置 2
0x15202454	SCA_INT_EDGE2	扫描设备中断边沿配置 2
0x15202458	SCA_INT_DUAL2	扫描设备中断双沿配置 2
0x15202460	SCA_INT_MAP0	扫描设备中断路由配置 0
0x15202464	SCA_INT_MAP1	扫描设备中断路由配置 1
0x15202468	SCA_INT_MAP2	扫描设备中断路由配置 2
0x1520246c	SCA_INT_MAP3	扫描设备中断路由配置 3
0x15202470	SCA_INT_MAP4	扫描设备中断路由配置 4
0x15202474	SCA_INT_MAP5	扫描设备中断路由配置 5

3.5.2 中断触发类型

对于 2P0300 来说，内部各控制器中断分为脉冲、电平触发类型，gpio 中断根据需要可以配置成电平触发或者脉冲触发。

3.5.3 中断相关寄存器描述

表 3-92 主系统传统中断控制寄存器属性

位域	访问属性/缺省值						中断源
	Intedge	Inten	Intenset	Intenclr	Intpol	Intentry	
0	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	MSYS_UART0
1	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	MSYS_UART1
2	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	MSYS_APB-DMA0~1
3	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	MSYS_APB-DMA2~3
4	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	MSYS_I2C
5	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	MSYS_SDIO1
6	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	MSYS_SPI1-IO
7	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	MSYS_GMAC_WK
8	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	MSYS_HPETO

位 域	访问属性/缺省值						中断源
	Intedge	Inten	Intenset	Intenclr	Intpol	Intentry	
9	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	MSYS_HPET1
10	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	MSYS_PMIO
11	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	MSYS_PWM0~3
12	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	MSYS_PWM4~7
13	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	MSYS_SDIO0
14	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	MSYS_SDIO0-DMA
15	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	MSYS_APB-DMA4~5
16	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	MSYS_APB-DMA6~7
17	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	MSYS_OTG
18	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	MSYS_GMAC
19	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	MSYS_SPI0
20	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	MSYS_EHCI
21	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	MSYS_OHCI
22	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	MSYS_SDIO1-DMA
23	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	MSYS_JBIG
24	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	MSYS_JPEG
25	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	Reserved
26	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	MSYS_PRMLB
27	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	MSYS_SCMLB
28	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	MSYS_GPIO0~15
29	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	MSYS_GPIO16~31
30	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	MSYS_GPIO32~47
31	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	Reserved
32	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	PRSYS_PRMLB
33	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	PRSYS_SCMLB
34	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	PRSYS_ADC
35	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	PRSYS_PWM0~3
36	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	PRSYS_PWM4~7
37	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	PRSYS_HPETO
38	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	PRSYS_I2C
39	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	PRSYS_UART
40	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	PRSYS_HPET1
41	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	PRSYS_APB-DMA0~1
42	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	PRSYS_APB-DMA2~3
43	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	PRSYS_APB-DMA4~5
44	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	PRSYS_PRTIFC

位域	访问属性/缺省值						中断源
	Intedge	Inten	Intenset	Intenclr	Intpol	Intentry	
45	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	PRSYS_PWM0~7
46	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	PRSYS_APB-DMA6~7
47	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	PRSYS_GPIO0~7
48	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	PRSYS_GPIO8~15
49	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	PRSYS_GPIO16~23
50	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	PRSYS_GPIO24~31
51	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	SCSYS_SCMLB
52	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	SCSYS_PRMLB
53	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	SCSYS_UART
54	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	SCSYS_HPETO
55	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	SCSYS_SPI
56	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	SCSYS_PMIO
57	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	SCSYS_HPET1
58	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	SCSYS_SCIFC
59	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	SCSYS_GPIO0~7
60	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	SCSYS_GPIO8~15
61	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	SCSYS_GPIO16~23
62	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	SCSYS_GPIO24~28
63	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	Reserved

表 3-93 传统中断控制寄存器地址

名称	地址偏移	访问属性	缺省值	描述
Intisr_0	0x14001420	RO	NA	低 32 位中断状态寄存器
Inten_0	0x14001424	RO	NA	低 32 位中断使能状态寄存器
Intenset_0	0x14001428	WO	NA	低 32 位设置使能寄存器
Intenclr_0	0x1400142c	WO	NA	低 32 位清除使能寄存器和脉冲触发的中断
Intpol_0	0x14001430	WO	0x0	低 32 位中断极性选择寄存器（1：极性取反，0：极性相同）
Intedge_0	0x14001434	WO	0x0	低 32 位触发方式寄存器（1：脉冲触发；0：电平触发）
Intisr_1	0x14001460	RO	NA	高 32 位中断状态寄存器
Inten_1	0x14001464	RO	NA	高 32 位中断使能状态寄存器
Intenset_1	0x14001468	WO	NA	高 32 位设置使能寄存器
Intenclr_1	0x1400146c	WO	NA	高 32 位清除使能寄存器和脉冲触发的中断
Intpol_1	0x14001470	WO	0x0	高 32 位中断极性选择寄存器（1：极性取反，0：极性相同）
Intedge_1	0x14001474	WO	0x0	高 32 位触发方式寄存器（1：脉冲触发；0：电平触发）
CORE_IPISR	0x14001000	RO	NA	处理器核的 IPI_Status 寄存器

名称	地址偏移	访问属性	缺省值	描述
CORE_IPIEN	0x14001004	RW	0x0	处理器核的 IPI_Enalbe 寄存器
CORE_IPISET	0x14001008	WO	NA	处理器核的 IPI_Set 寄存器
CORE_IPI_CLR	0x1400100c	WO	NA	处理器核的 IPI_Clear 寄存器
CORE_INTISRO	0x14001040	RO	NA	路由给 CORE 的低 32 位中断状态
CORE_INTISR1	0x14001048	RO	NA	路由给 CORE 的高 32 位中断状态

表 3-94 扩展中断控制寄存器属性

位域	访问属性/缺省值			中断源	位域	访问属性/缺省值			中断源
	IEN	ICLR	POL			IEN	ICLR	POL	
0	R / 0	W / 0	RW / 0	MSYS_UART0	82	R / 0	W / 0	RW / 0	PRSYS_ADC
1	R / 0	W / 0	RW / 0	MSYS_UART1	83	R / 0	W / 0	RW / 0	PRSYS_PWM0~1
2	R / 0	W / 0	RW / 0	Reserved	84	R / 0	W / 0	RW / 0	PRSYS_PWM2~3
3	R / 0	W / 0	RW / 0	Reserved	85	R / 0	W / 0	RW / 0	PRSYS_PWM4~5
4	R / 0	W / 0	RW / 0	MSYS_I2C	86	R / 0	W / 0	RW / 0	PRSYS_PWM6~7
5	R / 0	W / 0	RW / 0	Reserved	87	R / 0	W / 0	RW / 0	PRSYS_HPETO
6	R / 0	W / 0	RW / 0	MSYS_SPI1-I/O	88	R / 0	W / 0	RW / 0	PRSYS_HPET1
7	R / 0	W / 0	RW / 0	Reserved	89	R / 0	W / 0	RW / 0	PRSYS_I2C
8	R / 0	W / 0	RW / 0	MSYS_SDIO0	90	R / 0	W / 0	RW / 0	Reserved
9	R / 0	W / 0	RW / 0	MSYS_SDIO0_DMA	91	R / 0	W / 0	RW / 0	PRSYS_UART0
10	R / 0	W / 0	RW / 0	MSYS_SDIO1	92	R / 0	W / 0	RW / 0	PRSYS_UART1
11	R / 0	W / 0	RW / 0	MSYS_SDIO1_DMA	93	R / 0	W / 0	RW / 0	PRSYS_APB-DMA0
12	R / 0	W / 0	RW / 0	MSYS_HPETO_CNT0	94	R / 0	W / 0	RW / 0	PRSYS_APB-DMA1
13	R / 0	W / 0	RW / 0	MSYS_HPETO_CNT1	95	R / 0	W / 0	RW / 0	PRSYS_APB-DMA2
14	R / 0	W / 0	RW / 0	MSYS_HPETO_CNT2	96	R / 0	W / 0	RW / 0	PRSYS_APB-DMA3
15	R / 0	W / 0	RW / 0	MSYS_HPET1_CNT0	97	R / 0	W / 0	RW / 0	PRSYS_APB-DMA4
16	R / 0	W / 0	RW / 0	MSYS_HPET1_CNT1	98	R / 0	W / 0	RW / 0	PRSYS_APB-DMA5
17	R / 0	W / 0	RW / 0	MSYS_HPET1_CNT2	99	R / 0	W / 0	RW / 0	PRSYS_APB-DMA6
18	R / 0	W / 0	RW / 0	MSYS_PMIO	100	R / 0	W / 0	RW / 0	PRSYS_APB-DMA7
19 20 ~ 26	R / 0	W / 0	RW / 0	MSYS_PWM0 MSYS_PWM1 ~ MSYS_PWM7	101	R / 0	W / 0	RW / 0	PRSYS_PRTIFC
27 28	R / 0	W / 0	RW / 0	MSYS_APB-DMA0 MSYS_APB-DMA1	102	R / 0	W / 0	RW / 0	PRSYS_PMIO
29 30	R / 0	W / 0	RW / 0	MSYS_APB-DMA2 MSYS_APB-DMA3	104 105	R / 0	W / 0	RW / 0	PRSYS_GPIO0~3 PRSYS_GPIO4~7
31 32	R / 0	W / 0	RW / 0	MSYS_APB-DMA4 MSYS_APB-DMA5	106 107	R / 0	W / 0	RW / 0	PRSYS_GPIO8~11 PRSYS_GPIO12~15

位域	访问属性/缺省值			中断源	位域	访问属性/缺省值			中断源
	IEN	ICLR	POL			IEN	ICLR	POL	
33	R / 0	W / 0	RW / 0	MSYS_APB-DMA6	108	R / 0	W / 0	RW / 0	PRSYS_GPIO16~19
34				MSYS_APB-DMA7	109				PRSYS_GPIO20~23
39	R / 0	W / 0	RW / 0	MSYS_OTG	110	R / 0	W / 0	RW / 0	PRSYS_GPIO24~27
					111				PRSYS_GPIO28~31
40	R / 0	W / 0	RW / 0	MSYS_GMAC	112	R / 0	W / 0	RW / 0	SCSYS_SCMLB
41	R / 0	W / 0	RW / 0	MSYS_GPIO00~03	113	R / 0	W / 0	RW / 0	SCSYS_PRMLB
42				MSYS_GPIO04~07					
~				~					
56				MSYS_GPIO44~47					
57	R / 0	W / 0	RW / 0	MSYS_SPI	114	R / 0	W / 0	RW / 0	SCSYS_UART
58	R / 0	W / 0	RW / 0	MSYS_EHCI	115	R / 0	W / 0	RW / 0	SCSYS_HPETO
59	R / 0	W / 0	RW / 0	MSYS_OHCI	116	R / 0	W / 0	RW / 0	SCSYS_SPI
60	R / 0	W / 0	RW / 0	Reserved	117	R / 0	W / 0	RW / 0	SCSYS_PMIO
61	R / 0	W / 0	RW / 0	MSYS_JBIG	118	R / 0	W / 0	RW / 0	SCSYS_HPET1
62	R / 0	W / 0	RW / 0	MSYS_JPEG	119	R / 0	W / 0	RW / 0	SCSYS_SCIFC
63	R / 0	W / 0	RW / 0	Reserved	120	R / 0	W / 0	RW / 0	SCSYS_GPIO0~3
					121				SCSYS_GPIO4~7
64	R / 0	W / 0	RW / 0	MSYS_PRMLB	122	R / 0	W / 0	RW / 0	SCSYS_GPIO8~11
					123				SCSYS_GPIO12~15
65	R / 0	W / 0	RW / 0	MSYS_SCMLB	124	R / 0	W / 0	RW / 0	SCSYS_GPIO16~19
					125				SCSYS_GPIO20~23
66	R / 0	W / 0	RW / 0	MSYS_GMAC_WK	126	R / 0	W / 0	RW / 0	SCSYS_GPIO24~27
80	R / 0	W / 0	RW / 0	PRSYS_PRMLB	127	R / 0	W / 0	RW / 0	SCSYS_GPIO28
81	R / 0	W / 0	RW / 0	PRSYS_SCMLB					

表 3-95 打印系统 IO 中断控制寄存器属性

位域	访问属性/缺省值					中断源
	Intedge	Inten	Intclr	Intpol	Intdual	
0	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	PRSYS_PRMLB
1	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	PRSYS_SCMLB
2	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	PRSYS_ADC
3	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	PRSYS_PWM0~1
4	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	PRSYS_PWM2~3
5	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	PRSYS_PWM4~5
6	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	PRSYS_PWM6~7
8	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	PRSYS_HPETO_CNT0
9	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	PRSYS_HPETO_CNT1

位域	访问属性/缺省值					中断源
	Intedge	Inten	Intclr	Intpol	Intdual	
10	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	PRSYS_HPETO_CNT2
11	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	PRSYS_HPET1_CNT0
12	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	PRSYS_HPET1_CNT1
13	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	PRSYS_HPET1_CNT2
14	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	PRSYS_I2C
15	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	Reserved
16	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	PRSYS_UART0
17	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	PRSYS_UART1
18	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	PRSYS_APB-DMA0
19	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	PRSYS_APB-DMA1
20	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	PRSYS_APB-DMA2
21	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	PRSYS_APB-DMA3
22	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	PRSYS_APB-DMA4
23	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	PRSYS_APB-DMA5
24	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	PRSYS_APB-DMA6
25	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	PRSYS_APB-DMA7
26	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	PRSYS_PRTIFC
27	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	PRSYS_PMIO
28	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	Reserved
32 33 ~ 63	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	PRSYS_GPI000 PRSYS_GPI001 ~ PRSYS_GPI031
64	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	MSYS_UART00~01
65	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	MSYS_I2C
66	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	MSYS_SPI
67	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	MSYS_HPETO
68	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	MSYS_PMIO
69	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	MSYS_PWM
70	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	MSYS_SDIO0~1
71	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	MSYS_SDIO0/1_DMA
72	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	MSYS_HPET1
73	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	MSYS_APB-DMA0~7
74	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	MSYS_OTG
75	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	MSYS_GMAC
76	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	MSYS_GMAC_WK

位域	访问属性/缺省值					中断源
	Intedge	Inten	Intclr	Intpol	Intdual	
77	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	MSYS_PRMLB
78	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	MSYS_GPIO0~31
79	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	MSYS_GPIO32~47
80	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	SCSYS_SCMLB
81	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	SCSYS_PRMLB
82	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	SCSYS_UART
83	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	SCSYS_HPETO
84	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	SCSYS_SPI
85	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	SCSYS_PMIO
86	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	SCSYS_HPET1
87	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	SCSYS_SCIFC
88	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	SCSYS_GPIO0~3
89	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	SCSYS_GPIO4~7
90	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	SCSYS_GPIO8~11
91	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	SCSYS_GPIO12~15
92	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	SCSYS_GPIO16~19
93	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	SCSYS_GPIO20~23
94	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	SCSYS_GPIO24~27
95	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	SCSYS_GPIO28

表 3-96 打印系统中断控制寄存器描述

中断寄存器名称	地址偏移	读写属性	初始值	寄存器描述
int0_src	0x0	R	0	0 号中断状态寄存器，高电平有效
int0_en	0x4	R/W	0	0 号中断使能寄存器，高电平有效
int0_clr	0xc	W	-	0 号中断清除寄存器，写 1 清零，仅清除中断状态，不清除中断使能位
int0_pol	0x10	R/W	0	0 号中断极性配置寄存器，0-低电平触发，1-高电平触发
int0_edge	0x14	R/W	0	0 号中断边沿配置寄存器，0-电平中断，1-边沿中断
int0_dual	0x18	R/W	0	0 号中断双边沿使能配置寄存器，高电平有效，仅在选择边沿触发时有效
Int1_src	0x20	R	0	1 号中断状态寄存器，高电平有效
Int1_en	0x24	R/W	0	1 号中断使能寄存器，高电平有效
Int1_clr	0x2c	W	-	1 号中断清除寄存器，写 1 清零，仅清除中断状态，不清除中断使能位
Int1_pol	0x30	R/W	0	1 号中断极性配置寄存器，0-低电平触发，1-高电平触发

中断寄存器名称	地址偏移	读写属性	初始值	寄存器描述
Int1_edge	0x34	R/W	0	1 号中断边沿配置寄存器, 0-电平中断, 1-边沿中断
Int1_dual	0x38	R/W	0	1 号中断双边沿使能配置寄存器, 高电平有效, 仅在选择边沿触发时有效
Int2_src	0x40	R	0	2 号中断状态寄存器, 高电平有效
Int2_en	0x44	R/W	0	2 号中断使能寄存器, 高电平有效
Int2_clr	0x4c	W	-	2 号中断清除寄存器, 写 1 清零, 仅清除中断状态, 不清除中断使能位
Int2_pol	0x50	R/W	0	2 号中断极性配置寄存器, 0-低电平触发, 1-高电平触发
Int2_edge	0x54	R/W	0	2 号中断边沿配置寄存器, 0-电平中断, 1-边沿中断
Int2_dual	0x58	R/W	0	2 号中断双边沿使能配置寄存器, 高电平有效, 仅在选择边沿触发时有效
Int_map0 ~ Int_map5	0x60 ~ 0x74	R/W	0	96 个中断位路由对应处理器核 4 个中断号路由配置寄存器, 192 位配置, 每 2 位对应 1 个中断资源: 00:路由至处理器核 0 中断号; 01:路由至处理器核 1 中断号; 10:路由至处理器核 2 中断号; 11:路由至处理器核 3 中断号。

表 3-97 扫描系统 IO 中断控制寄存器属性

位域	访问属性/缺省值					中断源
	Intedge	Inten	Intclr	Intpol	Intdual	
0	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	SCSYS_SCMLB
1	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	SCSYS_PRMLB
2	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	SCSYS_UART
3	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	Reserved
4	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	SCSYS_SPI
5	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	SCSYS_PMIO
6	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	SCSYS_HPETO_CNT0
7	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	SCSYS_HPETO_CNT1
8	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	SCSYS_HPETO_CNT2
9	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	SCSYS_SCIFC
10	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	SCSYS_HPET1_CNT0
11	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	SCSYS_HPET1_CNT1
12	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	SCSYS_HPET1_CNT2

位域	访问属性/缺省值					中断源
	Intdge	Inten	Intclr	Intpol	Intdual	
13 14 ~ 44	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	SCSYS_GPIO0 SCSYS_GPIO1 ~ SCSYS_GPIO28
45 ~ 47	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	Reserved
48	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	MSYS_UART00~01
49	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	MSYS_I2C
50	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	MSYS_SPI0
51	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	MSYS_HPETO
52	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	MSYS_PMIO
53	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	MSYS_PWM
54	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	MSYS_SDIO0~1
55	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	MSYS_SDIO0/1_DMA
56	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	MSYS_HPET1
57	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	MSYS_APB-DMA0~7
58	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	MSYS_OTG
59	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	MSYS_GMAC
60	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	MSYS_GMAC_WK
61	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	MSYS_PRMLB
62	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	MSYS_GPIO0~31
63	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	MSYS_GPIO32~47
64	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	PRSYS_PRMLB
65	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	PRSYS_SCMLB
66	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	PRSYS_ADC
67	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	PRSYS_PWM0~1
68	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	PRSYS_PWM2~3
69	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	PRSYS_PWM4~5
70	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	PRSYS_PWM6~7
71	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	PRSYS_HPETO
72	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	PRSYS_HPET1
73	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	PRSYS_I2C
74	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	Reserved
75	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	PRSYS_UART0
76	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	PRSYS_UART1

位域	访问属性/缺省值					中断源
	Intedge	Inten	Intclr	Intpol	Intdual	
77	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	PRSYS_APB-DMA0
78	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	PRSYS_APB-DMA1
79	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	PRSYS_APB-DMA2
80	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	PRSYS_APB-DMA3
81	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	PRSYS_APB-DMA4
82	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	PRSYS_APB-DMA5
83	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	PRSYS_APB-DMA6
84	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	PRSYS_APB-DMA7
85	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	PRSYS_PRTIFC
86	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	PRSYS_PMI0
87	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	Reserved
88	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	PRSYS_GPIO0~03
89 ~ 95	RW / 0	RW / 0	W / 0	RW / 0	RW / 0	PRSYS_GPIO04~07 ~ PRSYS_GPIO28~31

表 3-98 扫描系统中断控制寄存器描述

中断寄存器名称	地址偏移	读写属性	初始值	寄存器描述
int0_src	0x0	R	0	0 号中断状态寄存器，高电平有效
int0_en	0x4	R/W	0	0 号中断使能寄存器，高电平有效
int0_clr	0xc	W	-	0 号中断清除寄存器，写 1 清零，仅清除中断状态，不清除中断使能位
int0_pol	0x10	R/W	0	0 号中断极性配置寄存器，0-低电平触发，1-高电平触发
int0_edge	0x14	R/W	0	0 号中断边沿配置寄存器，0-电平中断，1-边沿中断
int0_dual	0x18	R/W	0	0 号中断双边沿使能配置寄存器，高电平有效，仅在选择边沿触发时有效
Int1_src	0x20	R	0	1 号中断状态寄存器，高电平有效
Int1_en	0x24	R/W	0	1 号中断使能寄存器，高电平有效
Int1_clr	0x2c	W	-	1 号中断清除寄存器，写 1 清零，仅清除中断状态，不清除中断使能位
Int1_pol	0x30	R/W	0	1 号中断极性配置寄存器，0-低电平触发，1-高电平触发
Int1_edge	0x34	R/W	0	1 号中断边沿配置寄存器，0-电平中断，1-边沿中断
Int1_dual	0x38	R/W	0	1 号中断双边沿使能配置寄存器，高电平有效，仅在选择边沿触发时有效
Int2_src	0x40	R	0	2 号中断状态寄存器，高电平有效

中断寄存器名称	地址偏移	读写属性	初始值	寄存器描述
Int2_en	0x44	R/W	0	2 号中断使能寄存器，高电平有效
Int2_clr	0x4c	W	-	2 号中断清除寄存器，写 1 清零，仅清除中断状态，不清除中断使能位
Int2_pol	0x50	R/W	0	2 号中断极性配置寄存器，0-低电平触发，1-高电平触发
Int2_edge	0x54	R/W	0	2 号中断边沿配置寄存器，0-电平中断，1-边沿中断
Int2_dual	0x58	R/W	0	2 号中断双边沿使能配置寄存器，高电平有效，仅在选择边沿触发时有效
Int_map0 ~ Int_map5	0x60 ~ 0x74	R/W	0	96 个中断位路由对应处理器核 4 个中断号路由配置寄存器，192 位配置，每 2 位对应 1 个中断资源： 00:路由至处理器核 0 中断号； 01:路由至处理器核 1 中断号； 10:路由至处理器核 2 中断号； 11:路由至处理器核 3 中断号。

3.5.4 中断路由寄存器描述

龙芯 2P0300 中断源可以选择路由到处理器核中断 INT0 到 INT3 中的任意一个，即对应 CP0_Status 的 IP0 到 IP3。主系统传统中断中 64 个 I/O 中断源中每一个都对应一个 8 位的路由控制器，其格式和地址如下表所示。路由寄存器采用向量的方式进行路由选择，如 0x40 表示路由到处理器的 INT2 上。

表 3-99 传统中断路由寄存器的说明

位域	说 明
3:0	保留
7:4	路由的处理器核中断引脚向量号： 0001: 主系统处理器核 0 中断号； 0010: 主系统处理器核 1 中断号； 0100: 主系统处理器核 2 中断号； 1000: 主系统处理器核 3 中断号。

表 3-100 传统中断路由寄存器地址

名称	地址偏移	描述	名称	地址偏移	描述
Entry0	0x14001400	MSYS_UART0	Entry32	0x14001440	PRSYS_PRMLB
Entry1	0x14001401	MSYS_UART1	Entry33	0x14001441	PRSYS_SCMLB
Entry2	0x14001402	MSYS_APB-DMA0~1	Entry34	0x14001442	PRSYS_ADC
Entry3	0x14001403	MSYS_APB-DMA2~3	Entry35	0x14001443	PRSYS_PWM0~3

名称	地址偏移	描述	名称	地址偏移	描述
Entry4	0x14001404	MSYS_I2C	Entry36	0x14001444	PRSYS_PWM4~7
Entry5	0x14001405	MSYS_SDIO1	Entry37	0x14001445	PRSYS_HPETO
Entry6	0x14001406	MSYS_SPI1-IO	Entry38	0x14001446	PRSYS_I2C
Entry7	0x14001407	MSYS_GMAC_WK	Entry39	0x14001447	PRSYS_UART
Entry8	0x14001408	MSYS_HPETO	Entry40	0x14001448	PRSYS_HPET1
Entry9	0x14001409	MSYS_HPET1	Entry41	0x14001449	PRSYS_APB-DMA0~1
Entry10	0x1400140a	MSYS_PMIO	Entry42	0x1400144a	PRSYS_APB-DMA2~3
Entry11	0x1400140b	MSYS_PWM0~3	Entry43	0x1400144b	PRSYS_APB-DMA4~5
Entry12	0x1400140c	MSYS_PWM4~7	Entry44	0x1400144c	PRSYS_PRTIFC
Entry13	0x1400140d	MSYS_SDIO0	Entry45	0x1400144d	PRSYS_PWM0~7
Entry14	0x1400140e	MSYS_SDIO0-DMA	Entry46	0x1400144e	PRSYS_APB-DMA6~7
Entry15	0x1400140f	MSYS_APB-DMA4~5	Entry47	0x1400144f	PRSYS_GPIO0~7
Entry16	0x14001410	MSYS_APB-DMA6~7	Entry48	0x14001450	PRSYS_GPIO8~15
Entry17	0x14001411	MSYS_OTG	Entry49	0x14001451	PRSYS_GPIO16~23
Entry18	0x14001412	MSYS_GMAC	Entry50	0x14001452	PRSYS_GPIO24~31
Entry19	0x14001413	MSYS_SPI0	Entry51	0x14001453	SCSYS_SCMLB
Entry20	0x14001414	MSYS_EHCI	Entry52	0x14001454	SCSYS_PRMLB
Entry21	0x14001415	MSYS_OHCI	Entry53	0x14001455	SCSYS_UART
Entry22	0x14001416	Reserved	Entry54	0x14001456	SCSYS_HPETO
Entry23	0x14001417	MSYS_JBIG	Entry55	0x14001457	SCSYS_SPI
Entry24	0x14001418	MSYS_JPEG	Entry56	0x14001458	SCSYS_PMIO
Entry25	0x14001419	Reserved	Entry57	0x14001459	SCSYS_HPET1
Entry26	0x1400141a	MSYS_PRMLB	Entry58	0x1400145a	SCSYS_SCIFC
Entry27	0x1400141b	MSYS_SCMLB	Entry59	0x1400145b	SCSYS_GPIO0~7
Entry28	0x1400141c	MSYS_GPIO0~15	Entry60	0x1400145c	SCSYS_GPIO8~15
Entry29	0x1400141d	MSYS_GPIO16~31	Entry61	0x1400145d	SCSYS_GPIO16~23
Entry30	0x1400141e	MSYS_GPIO32~47	Entry62	0x1400145e	SCSYS_GPIO24~28
Entry31	0x1400141f	Reserved	Entry63	0x1400145f	Reserved

表 3-101 扩展中断路由寄存器的说明

位域	EXTIOI_MAP 寄存器说明
31:28	保留
27:24	EXT_IOI_ISR[127:96]统一路由的处理器核中断引脚向量号： 0001：主系统处理器核 0 中断号； 0010：主系统处理器核 1 中断号； 0100：主系统处理器核 2 中断号； 1000：主系统处理器核 3 中断号。
23:20	保留
19:16	EXT_IOI_ISR[95:64]统一路由的处理器核中断引脚向量号： 0001：主系统处理器核 0 中断号； 0010：主系统处理器核 1 中断号； 0100：主系统处理器核 2 中断号； 1000：主系统处理器核 3 中断号。
15:12	保留
11:8	EXT_IOI_ISR[63:32]统一路由的处理器核中断引脚向量号： 0001：主系统处理器核 0 中断号； 0010：主系统处理器核 1 中断号； 0100：主系统处理器核 2 中断号； 1000：主系统处理器核 3 中断号。
7:4	保留
3:0	EXT_IOI_ISR[31:0]统一路由的处理器核中断引脚向量号： 0001：主系统处理器核 0 中断号； 0010：主系统处理器核 1 中断号； 0100：主系统处理器核 2 中断号； 1000：主系统处理器核 3 中断号。

表 3-102 打印系统中断路由寄存器的说明

位域	INT_MAP 寄存器说明
PRSYS_INT_MAP4[31:0]	打印系统共 96 个 IO 中断位，路由对应打印系统处理器核 4 个中断号路由配置寄存器，共计 5 组路由配置寄存器 192 位路由配置，每 2 位对应 1 个中断资源，路由选择： 00:路由至打印系统处理器核 0 中断号； 01:路由至打印系统处理器核 1 中断号； 10:路由至打印系统处理器核 2 中断号； 11:路由至打印系统处理器核 3 中断号。
PRSYS_INT_MAP3[31:0]	
PRSYS_INT_MAP2[31:0]	
PRSYS_INT_MAP1[31:0]	
PRSYS_INT_MAP0[31:0]	

表 3-103 扫描系统中断路由寄存器的说明

位域	INT_MAP 寄存器说明
SCSYS_INT_MAP4[31:0]	扫描系统共 96 个 IO 中断位，路由对应扫描系统处理器核 4 个中断号路由配置寄存器，共计 5 组路由配置寄存器 192 位路由配置，每 2 位对应 1 个中断资源，路由选择： 00:路由至扫描系统处理器核 0 中断号； 01:路由至扫描系统处理器核 1 中断号； 10:路由至扫描系统处理器核 2 中断号； 11:路由至扫描系统处理器核 3 中断号。
SCSYS_INT_MAP3[31:0]	
SCSYS_INT_MAP2[31:0]	
SCSYS_INT_MAP1[31:0]	
SCSYS_INT_MAP0[31:0]	

4 DDR 控制器

龙芯 2P0300 处理器内部集成的 DDR3/4 SDRAM 内存控制器。

4.1 访问地址

DDR 控制器包括两个地址空间，分别如下：

表 4-1 内存控制器地址空间分配

起始地址	结束名称	名称	说明
0x0FF0_0000	0x0FFF_FFFF	配置空间	当 mc_default_reg =1 时， 或 mc_default_reg = 0 且 mc_disable_reg = 0 时，为配置空间。 其它情况下为内存空间
其它		内存空间	使用 X2 的窗口配置路由至 DDR 的所有地址

具体的 mc_default_reg 和 mc_disable_reg 配置请参考主系统通用配置 0。功能概述

龙芯 2P0300 处理器内存处理器支持 1 个 CS（由 1 个片选信号实现），一共含有 22 位的地址总线（即：18 位的行列地址总线、2 位的逻辑 Bank 总线和 2 位逻辑 Bank Group 总线，其中行列地址总线与 RASn、CASn 和 Wen 复用）。

CPU 发送的内存请求物理地址可以根据控制器内部不同的配置进行多种不同的地址映射。

龙芯 2P0300 处理器中内存控制器具有如下特征：

1. 接口上命令、读写数据全流水操作；
2. 内存命令合并、排序提高整体带宽；
3. 配置寄存器读写端口，可以修改内存设备的基本参数；
4. 内建动态延迟补偿电路（DCC），用于数据的可靠发送和接收；
5. 支持 DDR3/4 SDRAM，且参数配置支持 x8 颗粒；
6. 控制器与 PHY 频率比 1/2 ；
7. 支持数据传输速率范围为 800Mbps-1600Mbps。

4.2 DDR 控制器寄存器

由于系统中可能使用不同类型的 DDR3/4 SDRAM，因此，在系统上电复位以后，需要对 DDR3/4 SDRAM 进行配置。在 JESD79-4 中规定了详细的配置操作和配置过程，在没有完成 DDR 的内存初始化操作之前，DDR 不可用。内存初始化操作执行顺序如下：

- 1) 系统复位，此时控制器内部所有寄存器内容将被清除为初始值。
- 2) 系统解复位。
- 3) 向配置寄存器地址发写指令，配置所有 DDR3/4 配置寄存器。所有寄存器都必须正确配置才可以正常工作。

4) 配置结束后内存控制器将自动对内存发起初始化指令。

在龙芯 2P0300 处理器设计中，DDR3/4 SDRAM 的配置在系统主板初始化完成以后，需要使用内存之前，进行内存类型的配置。具体的配置操作是对物理地址 0x0ff0 0000 相对应的配置寄存器写入相应的配置参数。一个寄存器可能会包括多个、一个、部分参数的数据。这些配置寄存器及其包含的参数意义如下表（寄存器中未使用的位均为保留位），具体的配置可以根据实际情况再决定：

表 4-2 DDR SDRAM 配置参数寄存器

Offset	63:55	55:48	47:40	39:32	31:24	23:16	15:8	7:0
PHY								
0x0000							version(RD)	
0x0008		switch_byte38		ddr3_mode			capability (RD)	
0x0010							dram_init(RD)	init_start
0x0018								
0x0020							preamble2	rdfifo_valid
0x0028		rdfifo_empty(RD)				Overflow(RD)		
0x0030		dll_value(RD)	dll_init_done(RD)	dll_lock_mode	dll_bypass	dll_adj_cnt	dll_increment	dll_start_point
0x0038				dll_dbl_fix			dll_close_disable	dll_ck
0x0040								
0x0048							clken_ckca	
0x0050								
0x0058							clken_ds_0	
0x0060								
0x0068							clken_ds_1	
0x0070								
0x0078							clken_ds_2	
0x0080								
0x0088							clken_ds_3	
0x0090								
0x0098							clken_ds_4	
0x00a0								
0x00a8							clken_ds_5	
0x00b0								
0x00b8							clken_ds_6	
0x00c0								
0x00c8							clken_ds_7	
0x00d0								
0x00d8							clken_ds_8	
0x00e0			vrefclk_inv	vref_sample		vref_num	vref_dly	dll_vref
.....								
0x0100					dll_1xdly_0	dll_1xgen_0	dll_wrdsq_0	dll_wrdsq_0
0x0108						dll_gate_0	dll_rdds1_0	dll_rdds0_0
0x0110	rdodt_ctrl_0	rdgate_len_0	rdgate_mode_0	rdgate_ctrl_0			dqs_oe_ctrl_0	dq_oe_ctrl_0
0x0118					dly_2x_0		redge_sel_0	rdds_phase_0(RD)
0x0120	w_bdly0_0[31:28]	w_bdly0_0[27:24]	w_bdly0_0[23:20]	w_bdly0_0[19:16]	w_bdly0_0[15:12]	w_bdly0_0[11:8]	w_bdly0_0[7:4]	w_bdly0_0[3:0]

0x0128		w_bdlly0_0[59:56]	w_bdlly0_0[55:52]	w_bdlly0_0[51:48]	w_bdlly0_0[47:44]	w_bdlly0_0[43:40]	w_bdlly0_0[39:36]	w_bdlly0_0[35:32]
0x0130	w_bdlly1_0[24:21]	w_bdlly1_0[20:18]	w_bdlly1_0[17:15]	w_bdlly1_0[14:12]	w_bdlly1_0[11:9]	w_bdlly1_0[8:6]	w_bdlly1_0[5:3]	w_bdlly1_0[2:0]
0x0138								w_bdlly1_0[27:26]
0x0140							rg_bdlly_0[7:4]	rg_bdlly_0[3:0]
0x0148								
0x0150	rdqsp_bdlly_0[31:28]	rdqsp_bdlly_0[27:24]	rdqsp_bdlly_0[23:20]	rdqsp_bdlly_0[19:16]	rdqsp_bdlly_0[15:12]	rdqsp_bdlly_0[11:8]	rdqsp_bdlly_0[7:4]	rdqsp_bdlly_0[3:0]
0x0158								rdqsp_bdlly_0[35:32]
0x0160	rdqsn_bdlly_0[31:28]	rdqsn_bdlly_0[27:24]	rdqsn_bdlly_0[23:20]	rdqsn_bdlly_0[19:16]	rdqsn_bdlly_0[15:12]	rdqsn_bdlly_0[11:8]	rdqsn_bdlly_0[7:4]	rdqsn_bdlly_0[3:0]
0x0168								rdqsn_bdlly_0[35:32]
0x0170	rdq_bdlly_0[24:21]	rdq_bdlly_0[20:18]	rdq_bdlly_0[17:15]	rdq_bdlly_0[14:12]	rdq_bdlly_0[11:9]	rdq_bdlly_0[8:6]	rdq_bdlly_0[5:3]	rdq_bdlly_0[2:0]
0x0178								rdq_bdlly_0[27:26]
0x0180					dll_1xdly_1	dll_1xgen_1	dll_wrdqs_1	dll_wrdq_1
0x0188						dll_gate_1	dll_rddqs_1	dll_rddqs0_1
0x0190	rdodt_ctrl_1	rdgate_len_1	rdgate_mode_1	rdgate_ctrl_1			dqs_oe_ctrl_1	dq_oe_ctrl_1
0x0198					dly_2x_1		redge_sel_1	rddqs_phase_1(RD)
0x01a0	w_bdlly0_1[31:28]	w_bdlly0_1[27:24]	w_bdlly0_1[23:20]	w_bdlly0_1[19:16]	w_bdlly0_1[15:12]	w_bdlly0_1[11:8]	w_bdlly0_1[7:4]	w_bdlly0_1[3:0]
0x01a8		w_bdlly0_1[59:56]	w_bdlly0_1[55:52]	w_bdlly0_1[51:48]	w_bdlly0_1[47:44]	w_bdlly0_1[43:40]	w_bdlly0_1[39:36]	w_bdlly0_1[35:32]
0x01b0	w_bdlly1_1[24:21]	w_bdlly1_1[20:18]	w_bdlly1_1[17:15]	w_bdlly1_1[14:12]	w_bdlly1_1[11:9]	w_bdlly1_1[8:6]	w_bdlly1_1[5:3]	w_bdlly1_1[2:0]
0x01b8								w_bdlly1_1[27:26]
0x01c0							rg_bdlly_1[7:4]	rg_bdlly_1[3:0]
0x01c8								
0x01d0	rdqsp_bdlly_1[31:28]	rdqsp_bdlly_1[27:24]	rdqsp_bdlly_1[23:20]	rdqsp_bdlly_1[19:16]	rdqsp_bdlly_1[15:12]	rdqsp_bdlly_1[11:8]	rdqsp_bdlly_1[7:4]	rdqsp_bdlly_1[3:0]
0x01d8								rdqsp_bdlly_1[35:32]
0x01e0	rdqsn_bdlly_1[31:28]	rdqsn_bdlly_1[27:24]	rdqsn_bdlly_1[23:20]	rdqsn_bdlly_1[19:16]	rdqsn_bdlly_1[15:12]	rdqsn_bdlly_1[11:8]	rdqsn_bdlly_1[7:4]	rdqsn_bdlly_1[3:0]
0x01e8								rdqsn_bdlly_1[35:32]
0x01f0	rdq_bdlly_1[24:21]	rdq_bdlly_1[20:18]	rdq_bdlly_1[17:15]	rdq_bdlly_1[14:12]	rdq_bdlly_1[11:9]	rdq_bdlly_1[8:6]	rdq_bdlly_1[5:3]	rdq_bdlly_1[2:0]
0x01f8								rdq_bdlly_1[27:26]
0x0200					dll_1xdly_2	dll_1xgen_2	dll_wrdqs_2	dll_wrdq_2
0x0208						dll_gate_2	dll_rddqs_2	dll_rddqs0_2
0x0210	rdodt_ctrl_2	rdgate_len_2	rdgate_mode_2	rdgate_ctrl_2			dqs_oe_ctrl_2	dq_oe_ctrl_2
0x0218					dly_2x_2		redge_sel_2	rddqs_phase_2(RD)
0x0220	w_bdlly0_2[31:28]	w_bdlly0_2[27:24]	w_bdlly0_2[23:20]	w_bdlly0_2[19:16]	w_bdlly0_2[15:12]	w_bdlly0_2[11:8]	w_bdlly0_2[7:4]	w_bdlly0_2[3:0]
0x0228		w_bdlly0_2[59:56]	w_bdlly0_2[55:52]	w_bdlly0_2[51:48]	w_bdlly0_2[47:44]	w_bdlly0_2[43:40]	w_bdlly0_2[39:36]	w_bdlly0_2[35:32]
0x0230	w_bdlly1_2[24:21]	w_bdlly1_2[20:18]	w_bdlly1_2[17:15]	w_bdlly1_2[14:12]	w_bdlly1_2[11:9]	w_bdlly1_2[8:6]	w_bdlly1_2[5:3]	w_bdlly1_2[2:0]
0x0238								w_bdlly1_2[27:26]
0x0240							rg_bdlly_2[7:4]	rg_bdlly_2[3:0]
0x0248								
0x0250	rdqsp_bdlly_2[31:28]	rdqsp_bdlly_2[27:24]	rdqsp_bdlly_2[23:20]	rdqsp_bdlly_2[19:16]	rdqsp_bdlly_2[15:12]	rdqsp_bdlly_2[11:8]	rdqsp_bdlly_2[7:4]	rdqsp_bdlly_2[3:0]
0x0258								rdqsp_bdlly_2[35:32]
0x0260	rdqsn_bdlly_2[31:28]	rdqsn_bdlly_2[27:24]	rdqsn_bdlly_2[23:20]	rdqsn_bdlly_2[19:16]	rdqsn_bdlly_2[15:12]	rdqsn_bdlly_2[11:8]	rdqsn_bdlly_2[7:4]	rdqsn_bdlly_2[3:0]
0x0268								rdqsn_bdlly_2[35:32]
0x0270	rdq_bdlly_2[24:21]	rdq_bdlly_2[20:18]	rdq_bdlly_2[17:15]	rdq_bdlly_2[14:12]	rdq_bdlly_2[11:9]	rdq_bdlly_2[8:6]	rdq_bdlly_2[5:3]	rdq_bdlly_2[2:0]
0x0278								rdq_bdlly_2[27:26]
0x0280					dll_1xdly_3	dll_1xgen_3	dll_wrdqs_3	dll_wrdq_3
0x0288						dll_gate_3	dll_rddqs_3	dll_rddqs0_3
0x0290	rdodt_ctrl_3	rdgate_len_3	rdgate_mode_3	rdgate_ctrl_3			dqs_oe_ctrl_3	dq_oe_ctrl_3
0x0298					dly_2x_3		redge_sel_3	rddqs_phase_3(RD)

0x02a0	w_bdlly0_3[31:28]	w_bdlly0_3[27:24]	w_bdlly0_3[23:20]	w_bdlly0_3[19:16]	w_bdlly0_3[15:12]	w_bdlly0_3[11:8]	w_bdlly0_3[7:4]	w_bdlly0_3[3:0]
0x02a8		w_bdlly0_3[59:56]	w_bdlly0_3[55:52]	w_bdlly0_3[51:48]	w_bdlly0_3[47:44]	w_bdlly0_3[43:40]	w_bdlly0_3[39:36]	w_bdlly0_3[35:32]
0x02b0	w_bdlly1_3[24:21]	w_bdlly1_3[20:18]	w_bdlly1_3[17:15]	w_bdlly1_3[14:12]	w_bdlly1_3[11:9]	w_bdlly1_3[8:6]	w_bdlly1_3[5:3]	w_bdlly1_3[2:0]
0x02b8								w_bdlly1_3[27:26]
0x02c0							rg_bdlly_3[7:4]	rg_bdlly_3[3:0]
0x02c8								
0x02d0	rdqsp_bdlly_3[31:28]	rdqsp_bdlly_3[27:24]	rdqsp_bdlly_3[23:20]	rdqsp_bdlly_3[19:16]	rdqsp_bdlly_3[15:12]	rdqsp_bdlly_3[11:8]	rdqsp_bdlly_3[7:4]	rdqsp_bdlly_3[3:0]
0x02d8								rdqsp_bdlly_3[35:32]
0x02e0	rdqsn_bdlly_3[31:28]	rdqsn_bdlly_3[27:24]	rdqsn_bdlly_3[23:20]	rdqsn_bdlly_3[19:16]	rdqsn_bdlly_3[15:12]	rdqsn_bdlly_3[11:8]	rdqsn_bdlly_3[7:4]	rdqsn_bdlly_3[3:0]
0x02e8								rdqsn_bdlly_3[35:32]
0x02f0	rdq_bdlly_3[24:21]	rdq_bdlly_3[20:18]	rdq_bdlly_3[17:15]	rdq_bdlly_3[14:12]	rdq_bdlly_3[11:9]	rdq_bdlly_3[8:6]	rdq_bdlly_3[5:3]	rdq_bdlly_3[2:0]
0x02f8								rdq_bdlly_3[27:26]
0x0300					dll_1xdly_4	dll_1xgen_4	dll_wrds_4	dll_wrds_4
0x0308						dll_gate_4	dll_rdds_4	dll_rdds_4
0x0310	rdodt_ctrl_4	rdgate_len_4	rdgate_mode_4	rdgate_ctrl_4			dqs_oe_ctrl_4	dq_oe_ctrl_4
0x0318					dly_2x_4		redge_sel_4	rdqs_phase_4(RD)
0x0320	w_bdlly0_4[31:28]	w_bdlly0_4[27:24]	w_bdlly0_4[23:20]	w_bdlly0_4[19:16]	w_bdlly0_4[15:12]	w_bdlly0_4[11:8]	w_bdlly0_4[7:4]	w_bdlly0_4[3:0]
0x0328		w_bdlly0_4[59:56]	w_bdlly0_4[55:52]	w_bdlly0_4[51:48]	w_bdlly0_4[47:44]	w_bdlly0_4[43:40]	w_bdlly0_4[39:36]	w_bdlly0_4[35:32]
0x0330	w_bdlly1_4[24:21]	w_bdlly1_4[20:18]	w_bdlly1_4[17:15]	w_bdlly1_4[14:12]	w_bdlly1_4[11:9]	w_bdlly1_4[8:6]	w_bdlly1_4[5:3]	w_bdlly1_4[2:0]
0x0338								w_bdlly1_4[27:26]
0x0340							rg_bdlly_4[7:4]	rg_bdlly_4[3:0]
0x0348								
0x0350	rdqsp_bdlly_4[31:28]	rdqsp_bdlly_4[27:24]	rdqsp_bdlly_4[23:20]	rdqsp_bdlly_4[19:16]	rdqsp_bdlly_4[15:12]	rdqsp_bdlly_4[11:8]	rdqsp_bdlly_4[7:4]	rdqsp_bdlly_4[3:0]
0x0358								rdqsp_bdlly_4[35:32]
0x0360	rdqsn_bdlly_4[31:28]	rdqsn_bdlly_4[27:24]	rdqsn_bdlly_4[23:20]	rdqsn_bdlly_4[19:16]	rdqsn_bdlly_4[15:12]	rdqsn_bdlly_4[11:8]	rdqsn_bdlly_4[7:4]	rdqsn_bdlly_4[3:0]
0x0368								rdqsn_bdlly_4[35:32]
0x0370	rdq_bdlly_4[24:21]	rdq_bdlly_4[20:18]	rdq_bdlly_4[17:15]	rdq_bdlly_4[14:12]	rdq_bdlly_4[11:9]	rdq_bdlly_4[8:6]	rdq_bdlly_4[5:3]	rdq_bdlly_4[2:0]
0x0378								rdq_bdlly_4[27:26]
0x0380					dll_1xdly_5	dll_1xgen_5	dll_wrds_5	dll_wrds_5
0x0388						dll_gate_5	dll_rdds_5	dll_rdds_5
0x0390	rdodt_ctrl_5	rdgate_len_5	rdgate_mode_5	rdgate_ctrl_5			dqs_oe_ctrl_5	dq_oe_ctrl_5
0x0398					dly_2x_5		redge_sel_5	rdqs_phase_5(RD)
0x03a0	w_bdlly0_5[31:28]	w_bdlly0_5[27:24]	w_bdlly0_5[23:20]	w_bdlly0_5[19:16]	w_bdlly0_5[15:12]	w_bdlly0_5[11:8]	w_bdlly0_5[7:4]	w_bdlly0_5[3:0]
0x03a8		w_bdlly0_5[59:56]	w_bdlly0_5[55:52]	w_bdlly0_5[51:48]	w_bdlly0_5[47:44]	w_bdlly0_5[43:40]	w_bdlly0_5[39:36]	w_bdlly0_5[35:32]
0x03b0	w_bdlly1_5[24:21]	w_bdlly1_5[20:18]	w_bdlly1_5[17:15]	w_bdlly1_5[14:12]	w_bdlly1_5[11:9]	w_bdlly1_5[8:6]	w_bdlly1_5[5:3]	w_bdlly1_5[2:0]
0x03b8								w_bdlly1_5[27:26]
0x03c0							rg_bdlly_5[7:4]	rg_bdlly_5[3:0]
0x03c8								
0x03d0	rdqsp_bdlly_5[31:28]	rdqsp_bdlly_5[27:24]	rdqsp_bdlly_5[23:20]	rdqsp_bdlly_5[19:16]	rdqsp_bdlly_5[15:12]	rdqsp_bdlly_5[11:8]	rdqsp_bdlly_5[7:4]	rdqsp_bdlly_5[3:0]
0x03d8								rdqsp_bdlly_5[35:32]
0x03e0	rdqsn_bdlly_5[31:28]	rdqsn_bdlly_5[27:24]	rdqsn_bdlly_5[23:20]	rdqsn_bdlly_5[19:16]	rdqsn_bdlly_5[15:12]	rdqsn_bdlly_5[11:8]	rdqsn_bdlly_5[7:4]	rdqsn_bdlly_5[3:0]
0x03e8								rdqsn_bdlly_5[35:32]
0x03f0	rdq_bdlly_5[24:21]	rdq_bdlly_5[20:18]	rdq_bdlly_5[17:15]	rdq_bdlly_5[14:12]	rdq_bdlly_5[11:9]	rdq_bdlly_5[8:6]	rdq_bdlly_5[5:3]	rdq_bdlly_5[2:0]
0x03f8								rdq_bdlly_5[27:26]
0x0400					dll_1xdly_6	dll_1xgen_6	dll_wrds_6	dll_wrds_6
0x0408						dll_gate_6	dll_rdds_6	dll_rdds_6
0x0410	rdodt_ctrl_6	rdgate_len_6	rdgate_mode_6	rdgate_ctrl_6			dqs_oe_ctrl_6	dq_oe_ctrl_6

0x0418					dly_2x_6		redge_sel_6	rddqs_phase_6(RD)
0x0420	w_bdy0_6[31:28]	w_bdy0_6[27:24]	w_bdy0_6[23:20]	w_bdy0_6[19:16]	w_bdy0_6[15:12]	w_bdy0_6[11:8]	w_bdy0_6[7:4]	w_bdy0_6[3:0]
0x0428		w_bdy0_6[59:56]	w_bdy0_6[55:52]	w_bdy0_6[51:48]	w_bdy0_6[47:44]	w_bdy0_6[43:40]	w_bdy0_6[39:36]	w_bdy0_6[35:32]
0x0430	w_bdy1_6[24:21]	w_bdy1_6[20:18]	w_bdy1_6[17:15]	w_bdy1_6[14:12]	w_bdy1_6[11:9]	w_bdy1_6[8:6]	w_bdy1_6[5:3]	w_bdy1_6[2:0]
0x0438								w_bdy1_6[27:26]
0x0440							rg_bdy_6[7:4]	rg_bdy_6[3:0]
0x0448								
0x0450	rdqsp_bdy_6[31:28]	rdqsp_bdy_6[27:24]	rdqsp_bdy_6[23:20]	rdqsp_bdy_6[19:16]	rdqsp_bdy_6[15:12]	rdqsp_bdy_6[11:8]	rdqsp_bdy_6[7:4]	rdqsp_bdy_6[3:0]
0x0458								rdqsp_bdy_6[35:32]
0x0460	rdqsn_bdy_6[31:28]	rdqsn_bdy_6[27:24]	rdqsn_bdy_6[23:20]	rdqsn_bdy_6[19:16]	rdqsn_bdy_6[15:12]	rdqsn_bdy_6[11:8]	rdqsn_bdy_6[7:4]	rdqsn_bdy_6[3:0]
0x0468								rdqsn_bdy_6[35:32]
0x0470	rdq_bdy_6[24:21]	rdq_bdy_6[20:18]	rdq_bdy_6[17:15]	rdq_bdy_6[14:12]	rdq_bdy_6[11:9]	rdq_bdy_6[8:6]	rdq_bdy_6[5:3]	rdq_bdy_6[2:0]
0x0478								rdq_bdy_6[27:26]
0x0480					dll_1xdly_7	dll_1xgen_7	dll_wrdqs_7	dll_wrdq_7
0x0488						dll_gate_7	dll_rddqs1_7	dll_rddqs0_7
0x0490	rdodt_ctrl_7	rdgate_len_7	rdgate_mode_7	rdgate_ctrl_7			dqs_oe_ctrl_7	dq_oe_ctrl_7
0x0498					dly_2x_7		redge_sel_7	rddqs_phase_7(RD)
0x04a0	w_bdy0_7[31:28]	w_bdy0_7[27:24]	w_bdy0_7[23:20]	w_bdy0_7[19:16]	w_bdy0_7[15:12]	w_bdy0_7[11:8]	w_bdy0_7[7:4]	w_bdy0_7[3:0]
0x04a8		w_bdy0_7[59:56]	w_bdy0_7[55:52]	w_bdy0_7[51:48]	w_bdy0_7[47:44]	w_bdy0_7[43:40]	w_bdy0_7[39:36]	w_bdy0_7[35:32]
0x04b0	w_bdy1_7[24:21]	w_bdy1_7[20:18]	w_bdy1_7[17:15]	w_bdy1_7[14:12]	w_bdy1_7[11:9]	w_bdy1_7[8:6]	w_bdy1_7[5:3]	w_bdy1_7[2:0]
0x04b8								w_bdy1_7[27:26]
0x04c0							rg_bdy_7[7:4]	rg_bdy_7[3:0]
0x04c8								
0x04d0	rdqsp_bdy_7[31:28]	rdqsp_bdy_7[27:24]	rdqsp_bdy_7[23:20]	rdqsp_bdy_7[19:16]	rdqsp_bdy_7[15:12]	rdqsp_bdy_7[11:8]	rdqsp_bdy_7[7:4]	rdqsp_bdy_7[3:0]
0x04d8								rdqsp_bdy_7[35:32]
0x04e0	rdqsn_bdy_7[31:28]	rdqsn_bdy_7[27:24]	rdqsn_bdy_7[23:20]	rdqsn_bdy_7[19:16]	rdqsn_bdy_7[15:12]	rdqsn_bdy_7[11:8]	rdqsn_bdy_7[7:4]	rdqsn_bdy_7[3:0]
0x04e8								rdqsn_bdy_7[35:32]
0x04f0	rdq_bdy_7[24:21]	rdq_bdy_7[20:18]	rdq_bdy_7[17:15]	rdq_bdy_7[14:12]	rdq_bdy_7[11:9]	rdq_bdy_7[8:6]	rdq_bdy_7[5:3]	rdq_bdy_7[2:0]
0x04f8								rdq_bdy_7[27:26]
0x0500					dll_1xdly_8	dll_1xgen_8	dll_wrdqs_8	dll_wrdq_8
0x0508						dll_gate_8	dll_rddqs1_8	dll_rddqs0_8
0x0510	rdodt_ctrl_8	rdgate_len_8	rdgate_mode_8	rdgate_ctrl_8			dqs_oe_ctrl_8	dq_oe_ctrl_8
0x0518					dly_2x_8		redge_sel_8	rddqs_phase_8(RD)
0x0520	w_bdy0_8[31:28]	w_bdy0_8[27:24]	w_bdy0_8[23:20]	w_bdy0_8[19:16]	w_bdy0_8[15:12]	w_bdy0_8[11:8]	w_bdy0_8[7:4]	w_bdy0_8[3:0]
0x0528		w_bdy0_8[59:56]	w_bdy0_8[55:52]	w_bdy0_8[51:48]	w_bdy0_8[47:44]	w_bdy0_8[43:40]	w_bdy0_8[39:36]	w_bdy0_8[35:32]
0x0530	w_bdy1_8[24:21]	w_bdy1_8[20:18]	w_bdy1_8[17:15]	w_bdy1_8[14:12]	w_bdy1_8[11:9]	w_bdy1_8[8:6]	w_bdy1_8[5:3]	w_bdy1_8[2:0]
0x0538								w_bdy1_8[27:26]
0x0540							rg_bdy_8[7:4]	rg_bdy_8[3:0]
0x0548								
0x0550	rdqsp_bdy_8[31:28]	rdqsp_bdy_8[27:24]	rdqsp_bdy_8[23:20]	rdqsp_bdy_8[19:16]	rdqsp_bdy_8[15:12]	rdqsp_bdy_8[11:8]	rdqsp_bdy_8[7:4]	rdqsp_bdy_8[3:0]
0x0558								rdqsp_bdy_8[35:32]
0x0560	rdqsn_bdy_8[31:28]	rdqsn_bdy_8[27:24]	rdqsn_bdy_8[23:20]	rdqsn_bdy_8[19:16]	rdqsn_bdy_8[15:12]	rdqsn_bdy_8[11:8]	rdqsn_bdy_8[7:4]	rdqsn_bdy_8[3:0]
0x0568								rdqsn_bdy_8[35:32]
0x0570	rdq_bdy_8[24:21]	rdq_bdy_8[20:18]	rdq_bdy_8[17:15]	rdq_bdy_8[14:12]	rdq_bdy_8[11:9]	rdq_bdy_8[8:6]	rdq_bdy_8[5:3]	rdq_bdy_8[2:0]
0x0578								rdq_bdy_8[27:26]
.....								
0x0700					leveling_cs	tLVL_DELAY	leveling_req(WR)	leveling_mode

0x0708							leveling_done(RD)	leveling_ready(RD)
0x0710	leveling_resp_7	leveling_resp_6	leveling_resp_5	leveling_resp_4	leveling_resp_3	leveling_resp_2	leveling_resp_1	leveling_resp_0
0x0718								leveling_resp_8
0x0720								
.....								
0x0800	dfe_ctrl_ds	pad_ctrl_ds				pad_ctrl_ck		
0x0808		pad_reset_po	pad_oplen_ca	pad_opdly_ca		pad_ctrl_ca		
0x0810	vref_ctrl_ds_3		vref_ctrl_ds_2		vref_ctrl_ds_1		vref_ctrl_ds_0	
0x0818	vref_ctrl_ds_7		vref_ctrl_ds_6		vref_ctrl_ds_5		vref_ctrl_ds_4	
0x0820							vref_ctrl_ds_8	
0x0828								
0x0830			pad_comp_o(RD)				pad_comp_i	
0x0838								
0x0840	pad_ctrl_ds_3		pad_ctrl_ds_2		pad_ctrl_ds_1		pad_ctrl_ds_0	
0x0848	pad_ctrl_ds_7		pad_ctrl_ds_6		pad_ctrl_ds_5		pad_ctrl_ds_4	
0x0850							pad_ctrl_ds_8	
.....								
0x0900		rdedge_soft		rd_phase(RD)			clk_inv	
0x0908							rdedge_inv	
CTL								
0x1000		tRP	tWLDQSEN	tMOD	tXPR		tCKE	tRESET
0x1008								tODTL
0x1010	tREFretention				tRFC		tREF	
0x1018	tCKESR	tXSRD	tXS		tRFC_dlr			tREF_IDLE
0x1020					tRDPDEN	tCPDED	tXPDLL	tXP
0x1028					tZQperiod	tZQCL	tZQCS	tZQ_CMD
.....								
0x1040	tRCD	tRRD_S_slr	tRRD_L_slr	tRRD_dlr				tRAS_min
0x1048				tRTP	tWR_CRC_DM	tWR	tFAW_slr	tFAW
0x1050	tWTR_S_CRC_DM	tWTR_L_CRC_DM	tWTR_S	tWTR		tCCD_dlr	tCCD_S_slr	tCCD_L_slr
0x1058								
0x1060			tPHY_WRLAT	tWL		tRDDATA	tPHY_RDLAT	tRL
0x1068				tCAL				tPL
0x1070			tW2P_sameba	tW2W_sameba	tW2R_sameba	tR2P_sameba	tR2W_sameba	tR2R_sameba
0x1078			tW2P_samebg	tW2W_samebg	tW2R_samebg	tR2P_samebg	tR2W_samebg	tR2R_samebg
0x1080			tW2P_samec	tW2W_samec	tW2R_samec	tR2P_samec	tR2W_samec	tR2R_samec
0x1088								
0x1090			tW2P_samecs	tW2W_samecs	tW2R_samecs	tR2P_samecs	tR2W_samecs	tR2R_samecs
0x1098				tW2W_diffcs	tW2R_diffcs		tR2W_diffcs	tR2R_diffcs
.....								
0x1100			cs_ref	cs_resync	cs_zqcl	cs_zq	cs_mrs	cs_enable
0x1108	cke_map				cs_map			
0x1110				cs2cid				cid_map
0x1118								
0x1120	mrs_done(RD)	mrs_req(WR)	pre_all_done(RD)	pre_all_req(WR)	cmd_cmd	status_cmd(RD)	cmd_req(WR)	command_mode
0x1128	cmd_cke	cmd_a			cmd_ba	cmd_bg	cmd_c	cmd_cs
0x1130								cmd_pda

0x1138						cmd_dq0		
0x1140	mr_3_cs_0		mr_2_cs_0		mr_1_cs_0		mr_0_cs_0	
0x1148	mr_3_cs_1		mr_2_cs_1		mr_1_cs_1		mr_0_cs_1	
0x1150	mr_3_cs_2		mr_2_cs_2		mr_1_cs_2		mr_0_cs_2	
0x1158	mr_3_cs_3		mr_2_cs_3		mr_1_cs_3		mr_0_cs_3	
0x1160	mr_3_cs_4		mr_2_cs_4		mr_1_cs_4		mr_0_cs_4	
0x1168	mr_3_cs_5		mr_2_cs_5		mr_1_cs_5		mr_0_cs_5	
0x1170	mr_3_cs_6		mr_2_cs_6		mr_1_cs_6		mr_0_cs_6	
0x1178	mr_3_cs_7		mr_2_cs_7		mr_1_cs_7		mr_0_cs_7	
0x1180	mr_3_cs_0_ddr4		mr_2_cs_0_ddr4		mr_1_cs_0_ddr4		mr_0_cs_0_ddr4	
0x1188			mr_6_cs_0_ddr4		mr_5_cs_0_ddr4		mr_4_cs_0_ddr4	
0x1190	mr_3_cs_1_ddr4		mr_2_cs_1_ddr4		mr_1_cs_1_ddr4		mr_0_cs_1_ddr4	
0x1198			mr_6_cs_1_ddr4		mr_5_cs_1_ddr4		mr_4_cs_1_ddr4	
0x11a0	mr_3_cs_2_ddr4		mr_2_cs_2_ddr4		mr_1_cs_2_ddr4		mr_0_cs_2_ddr4	
0x11a8			mr_6_cs_2_ddr4		mr_5_cs_2_ddr4		mr_4_cs_2_ddr4	
0x11b0	mr_3_cs_3_ddr4		mr_2_cs_3_ddr4		mr_1_cs_3_ddr4		mr_0_cs_3_ddr4	
0x11b8			mr_6_cs_3_ddr4		mr_5_cs_3_ddr4		mr_4_cs_3_ddr4	
0x11c0	mr_3_cs_4_ddr4		mr_2_cs_4_ddr4		mr_1_cs_4_ddr4		mr_0_cs_4_ddr4	
0x11c8			mr_6_cs_4_ddr4		mr_5_cs_4_ddr4		mr_4_cs_4_ddr4	
0x11d0	mr_3_cs_5_ddr4		mr_2_cs_5_ddr4		mr_1_cs_5_ddr4		mr_0_cs_5_ddr4	
0x11d8			mr_6_cs_5_ddr4		mr_5_cs_5_ddr4		mr_4_cs_5_ddr4	
0x11e0	mr_3_cs_6_ddr4		mr_2_cs_6_ddr4		mr_1_cs_6_ddr4		mr_0_cs_6_ddr4	
0x11e8			mr_6_cs_6_ddr4		mr_5_cs_6_ddr4		mr_4_cs_6_ddr4	
0x11f0	mr_3_cs_7_ddr4		mr_2_cs_7_ddr4		mr_1_cs_7_ddr4		mr_0_cs_7_ddr4	
0x11f8			mr_6_cs_7_ddr4		mr_5_cs_7_ddr4		mr_4_cs_7_ddr4	
0x1200			nc16_map	nc	channel_width	ba_xor_row_offset	addr_new	cs_place
0x1208						bg_xor_row_offset		addr_mirror
0x1210	addr_base_1				addr_base_0			
0x1218								
0x1220	addr_mask_1				addr_mask_0			
0x1228								
0x1230			cs_diff	c_diff	bg_diff	ba_diff	row_diff	col_diff
0x1238				CF_confbus_timeout				
0x1240	WRQthreshold	tRDQidle	wr_pkc_num	rwq_arb	retry	no_dead_inorder	placement_en	stb_en/pbuf
0x1248								tRWGNIdle
0x1250							rfifo_age	
0x1258	prior_age3		prior_age2		prior_age1		prior_age0	
0x1260	retry_cnt(RD)					rbuffer_max(RD)	rdfifo_depth	stat_en
0x1268								
.....								
0x1280	aw_512_align		rd_before_wr	ecc_enable		int_vector(RD)	int_trigger(RD)	int_enable
0x1288								
0x1290						int_cnt_fatal(RD)	int_cnt_err(RD)	int_cnt
0x1298	ecc_cnt_cs_7(RD)	ecc_cnt_cs_6(RD)	ecc_cnt_cs_5(RD)	ecc_cnt_cs_4(RD)	ecc_cnt_cs_3(RD)	ecc_cnt_cs_2(RD)	ecc_cnt_cs_1(RD)	ecc_cnt_cs_0(RD)
0x12a0	ecc_data_dir(RD)	ecc_code_dir(RD)	ecc_code_256(RD)					ecc_code_64(RD)
0x12a8	ecc_addr(RD)							
0x12b0	ecc_data[63:0](RD)							

0x12b8	ecc_data[127:64] (RD)							
0x12c0	ecc_data[191:128] (RD)							
0x12c8	ecc_data[255:192] (RD)							
.....								
0x1300							ref_num	ref_sch_en
0x1308							Status_sref(RD)	srefresh_req
.....								
0x1340	hardware_pd_7	hardware_pd_6	hardware_pd_5	hardware_pd_4	hardware_pd_3	hardware_pd_2	hardware_pd_1	hardware_pd_0
0x1348	power_sta_7(RD)	power_sta_6(RD)	power_sta_5(RD)	power_sta_4(RD)	power_sta_3(RD)	power_sta_2(RD)	power_sta_1(RD)	power_sta_0(RD)
0x1350	selfref_age		slowpd_age		fastpd_age		active_age	
0x1358				power_up				Age_step
0x1360	tCONF_IDLE				tLPMC_IDLE			
.....								
0x1380								zq_overlap
0x1388								zq_stat_en
0x1390	zq_cnt_1(RD)				zq_cnt_0(RD)			
0x1398	zq_cnt_3(RD)				zq_cnt_2(RD)			
0x13a0	zq_cnt_5(RD)				zq_cnt_4(RD)			
0x13a8	zq_cnt_6(RD)				zq_cnt_6(RD)			
.....								
0x13c0					odt_wr_cs_map			
0x13c8							odt_wr_length	odt_wr_delay
0x13d0					odt_rd_cs_map			
0x13d8							odt_rd_length	odt_rd_delay
.....								
0x1400				tRESYNC_length	tRESYNC_delay	tRESYNC_shift	tRESYNC_max	tRESYNC_min
.....								
0x1440					pre_predict		tm_cmdq_num	burst_length
0x1448								ca_timing
0x1450						wr/rd_dbi_en	ca_par_en	crc_en
0x1458							tCA_PAR	tWR_CRC
0x1460	bit_map_7	bit_map_6	bit_map_5	bit_map_6	bit_map_3	bit_map_2	bit_map_1	bit_map_0
0x1468	bit_map_15	bit_map_14	bit_map_13	bit_map_12	bit_map_11	bit_map_10	bit_map_9	bit_map_8
0x1470							bit_map_17	bit_map_16
0x1478								bitmap_mirror
0x1480				alertn_misc(RD)			alertn_cnt	alertn_clr
0x1488	alertn_addr(RD)							
.....								
0x1500	win0_base							
0x1508	win1_base							
0x1510	win2_base							
0x1518	win3_base							
0x1520	win4_base							
0x1528	win5_base							
0x1530	win6_base							
0x1538	win7_base							
.....								

0x1580	win0_mask							
0x1588	win1_mask							
0x1590	win2_mask							
0x1598	win3_mask							
0x15a0	win4_mask							
0x15a8	win5_mask							
0x15b0	win6_mask							
0x15b8	win7_mask							
.....								
0x1600	win0_mmap							
0x1608	win1_mmap							
0x1610	win2_mmap							
0x1618	win3_mmap							
0x1620	win4_mmap							
0x1628	win5_mmap							
0x1630	win6_mmap							
0x1638	win7_mmap							
.....								
0x1700							acc_hp	acc_en
0x1708	acc_fake_b				acc_fake_a			
0x1710								
0x1718								
0x1720	addr_base_acc_1				addr_base_acc_0			
0x1728								
0x1730	addr_mask_acc_1				addr_mask_acc_0			
0x1738								
MON								
0x2000								cmd_monitor
0x2008								
0x2010	cmd_fbck[63:0](RD)							
0x2018	cmd_fbck[127:64] (RD)							
0x2020					nw_switch_cnt(RD)			
.....								
0x2100								scheduler_mon
0x2108								
0x2110	sch_cmd_num(RD)							
0x2118	ba_conflict_all(RD)							
0x2120	ba_conflict_last1(RD)							
0x2128	ba_conflict_last2(RD)							
0x2130	ba_conflict_last3(RD)							
0x2138	ba_conflict_last4(RD)							
0x2140	ba_conflict_last5(RD)							
0x2148	ba_conflict_last6(RD)							
0x2150	ba_conflict_last7(RD)							
0x2158	ba_conflict_last8(RD)							
0x2160	rd_conflict(RD)							
0x2168	wr_conflict(RD)							

0x2170	rtw_conflict(RD)							
0x2178	wtr_conflict(RD)							
0x2180	rd_conflict_last1(RD)							
0x2188	wr_conflict_last1(RD)							
0x2190	rtw_conflict_last1(RD)							
0x2198	wtr_conflict_last1(RD)							
0x21a0	wr_rd_turnaround(RD)							
0x21a8	cs_turnaround(RD)							
0x21b0	bg_conflict(RD)							
.....								
0x2300						sm_leveling		sm_init
0x2308								
0x2310		sm_rank_03		sm_rank_02		sm_rank_01		sm_rank_00
0x2318		sm_rank_07		sm_rank_06		sm_rank_05		sm_rank_04
0x2320		sm_rank_11		sm_rank_10		sm_rank_09		sm_rank_08
0x2328		sm_rank_15		sm_rank_14		sm_rank_13		sm_rank_12
0x2330		sm_rank_19		sm_rank_18		sm_rank_17		sm_rank_16
0x2338		sm_rank_23		sm_rank_22		sm_rank_21		sm_rank_20
0x2340		sm_rank_27		sm_rank_26		sm_rank_25		sm_rank_24
0x2348		sm_rank_31		sm_rank_30		sm_rank_29		sm_rank_28
.....								
TST								
0x3000						lpbk_mode	lpbk_start	lpbk_en
0x3008	lpbk_correct(RD)				lpbk_counter(RD)			lpbk_error(RD)
0x3010	lpbk_data_en[63:0]							
0x3018								lpbk_data_en[71:64]
0x3020							lpbk_data_mask_en	
0x3028								
0x3030	Lpbk_dat_w0[63:0]							
0x3038	Lpbk_dat_w0[127:64]							
0x3040	Lpbk_dat_w1[63:0]							
0x3048	Lpbk_dat_w1[127:64]							
0x3050		lpbk_ecc_mask_w0	lpbk_dat_mask_w0				lpbk_ecc_w0	
0x3058		lpbk_ecc_mask_w1	lpbk_dat_mask_w1				lpbk_ecc_w1	
0x3060								prbs_23
0x3068						prbs_init		
.....								
0x3100					fix_data_pattern_index	bus_width	page_size	test_engine_en
0x3108			cs_diff_tst	c_diff_tst	bg_diff_tst	ba_diff_tst	row_diff_tst	col_diff_tst
0x3120	addr_base_tst							
0x3128								
0x3130	user_data_pattern							
0x3138								
0x3140	valid_bits[63:0]							
0x3148								valid_bits[71:64]
0x3150	ctrl[63:0]							
0x3158	ctrl[127:64]							

0x3160	obs[63:0] (RD)							
0x3168	obs[127:64] (RD)							
0x3170	obs[191:128] (RD)							
0x3178	obs[255:192] (RD)							
0x3180	obs[319:256] (RD)							
0x3188	obs[383:320] (RD)							
0x3190	obs[447:384] (RD)							
0x3198	obs[511:448] (RD)							
0x31a0	obs[575:512] (RD)							
0x31a8	obs[639:576] (RD)							
0x31b0					obs[671:640](RD)			
.....								
0x3200								
0x3208								
0x3220	tud_i0							
0x3228	tud_i1							
0x3230	tud_o(RD)							
.....								
0x3300	tst_300							
0x3308	tst_308							
0x3310	tst_310							
0x3318	tst_318							
0x3320	tst_320							
0x3328	tst_328							
0x3330	tst_330							
0x3338	tst_338							
0x3340	tst_340							
0x3348	tst_348							
0x3350	tst_350							
0x3358	tst_358							
0x3360	tst_360							
0x3368	tst_368							
0x3370	tst_370							
0x3378	tst_378							

5 GMAC 控制器

5.1 寄存器描述

龙芯 2P0300 集成了一个 GMAC 控制器，GMAC 控制器寄存器包括 GMAC 寄存器部分和 DMA 寄存器部分。

GMAC 的 GMAC 寄存器的起始地址是 0x1402_0000；GMAC 的 DMA 寄存器的起始地址是 0x1402_1000。

5.2 软件编程向导

DMA 初始化：

1. 软件重置(reset)GMAC
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 GMAC 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 GMAC 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 GMAC 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 GMAC 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 GMAC reg0 进行正确的配置
10. 设置 GMAC reg0[2]、[3]来启用 GMAC 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。

6 USB 控制器

6.1 总体概述

龙芯 2P0300 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 (High Speed) 的 USB 设备

支持一个 USB2.0 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。

6.2 控制器寄存器

表 6-1 USB 控制器地址空间分布

地址空间	名称	大小
0x1403,0000 - 0x1403,7fff	EHCI 寄存器	32KB
0x1403,8000 - 0x1403,ffff	OHCI 寄存器	32KB

7 OTG 控制器

7.1 概述

龙芯 2P0300 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 10 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 4 个 IN 方向的传输；

在 host 模式下，支持 12 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。

7.2 寄存器描述

表 7-1 OTG 控制器地址空间分布

地址空间	名称	大小
0x1410,0000 - 0x1413,ffff	OTG 寄存器	256KB

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。

8 SPI-FLASH 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。

8.1 SPI 控制器结构

龙芯 2P0300 集成的 SPI0-flash 控制器仅可作为主控端，所连接的是从设备。2P0300 共集成 3 个 SPI 控制器，其中 SPI0 支持 IO/memory 两种空间访问，SPI1~2 仅支持 IO 空间访问。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 0x1c000000，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，该启动功能仅 SPI0 控制器支持。2P0300 芯片 3 个 SPI 控制器空间分布，如下表。

表 8-1 SPI 控制器地址空间分布

地址空间	名称	大小
0x1c00,0000 - 0x1c0f,ffff	SPI0 Boot	1MB
0x1000,0000 - 0x13ff,ffff	SPI0 MEM	64MB
0x1401,0000 - 0x1401,ffff	SPI0 IO/CFG	64KB
0x1420,3000 - 0x1420,3fff	MAIN SPI IO	4KB
0x1520,3000 - 0x1520,3fff	SCAN SPI IO	4KB

对于 SPI 接口模块，使用时要注意将对应的芯片复用引脚设置为相应的接口功能。

与 SPI 接口相关的引脚复用设置可查询节 中的外设功能引脚复用关系表，并根据复用配置寄存器配置相应的 GPIO 引脚复用关系，实现对应设备功能引脚。

其结构如图 8-1 所示，由 AXI/APB 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)。

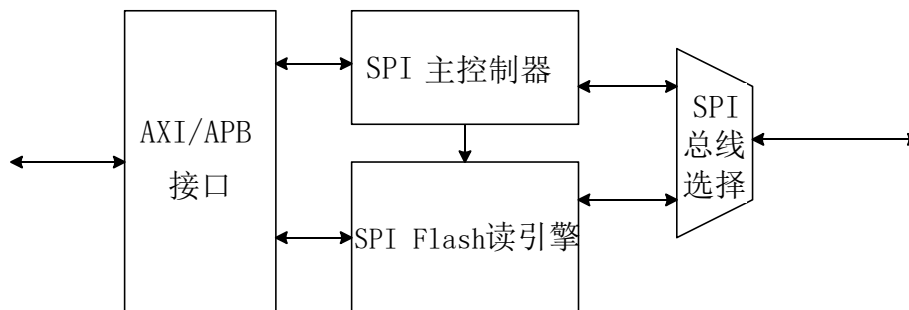


图8-1 SPI 控制器结构

8.2 配置寄存器

表 8-2 SPI0 配置寄存器列表

偏移	名称	描述	备注
0x0	SPCR/QSPI_CFG	控制寄存器/QSPI 配置寄存器	SPI0 控制器支持
0x1	SPSR/QSPI_STA	状态寄存器/QSPI 状态寄存器	SPI0 控制器支持
0x2	Tx/RxFIFO/ QSPI_DATA	TX/RX 数据寄存器/ QSPI 数据寄存器	SPI0 控制器支持
0x3	SPER/QSPI_MODE	外部寄存器/QSPI 模式切换寄存器	SPI0 控制器支持
0x4	SFC_PARAM/QSPI_TIMING	参数控制寄存器/QSPI 时序控制寄存器	SPI0 控制器支持
0x5	SFC_SOFTCS	片选控制寄存器	SPI0 控制器支持
0x6	SFC_TIMING	时序控制寄存器	SPI0 控制器支持
0x7	SFC_TIMING2	时序控制寄存器 2	SPI0 控制器支持
0x8	CTRL	自定义控制寄存器	SPI0 控制器支持
0x9	CMD	自定义命令寄存器	SPI0 控制器支持
0xa	BUF0	自定义数据寄存器 0	SPI0 控制器支持
0xb	BUF1	自定义数据寄存器 1	SPI0 控制器支持
0xc	TIMER0	自定义时序寄存器 0	SPI0 控制器支持
0xd	TIMER1	自定义时序寄存器 1	SPI0 控制器支持
0xe	TIMER2	自定义时序寄存器 2	SPI0 控制器支持
0xf	QSPI_SPACE_EN	QSPI 配置空间使能寄存器	SPI0 控制器支持

8.2.1 控制寄存器 (SPCR)

表 8-3 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
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

QSPI 控制器模式配置寄存器 (QSPI_CFG)

位域	名称	访问	初值	描述
7:6	Qmode	R/W	0	qspi mode: 高位为 CHOL, 低位 CHPA。
5	dummy	R/W	0	表征接下来传给 QSPI 的数据是 dummy
4:3	Qtype	R/W	0	QSPI 传输类型 00: standard spi; 01: dual spi; 10: quad spi。
2	duxen	R/W	0	开启双工模式。仅在 standard spi 模式下有效。开启后, 向 QSPI 写单字节数据的同时, QSPI 会回读一个字节。软件需要将这个字节读取。

位域	名称	访问	初值	描述
1:0	—	—	0	保留

8.2.2 状态寄存器 (SPSR)

表 8-4 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

QSPI 控制器模式配置/状态寄存器 (QSPI_STA)

位域	名称	访问	初值	描述
7:4	Qclk_div	R/W	0	QSPI 时钟分频系数, 分频值与 {spre, spr} 组合相同, 详见表 9-7
3	Qcs	R/W	1	QSPI 片选信号输出值
2	Qbusy	R	0	表征 QSPI 是否处于繁忙状态
1:0	—	—	0	保留

8.2.3 数据寄存器 (TxFIFO/RxFIFO)

表 8-5 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

QSPI 控制器模式数据寄存器 (QSPI_DATA)

位域	名称	访问	初值	描述
7:0	Qdat	R/W	0	写单字节数据, QSPI 转换为 IO 写设备; 读单字节数据, QSPI 转换为 IO 读设备。

8.2.4 外部寄存器 (SPER)

表 8-6 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

位域	名称	访问	初值	描述
1:0	spre	R/W	0	与 spr 一起设定分频的比率(I/O 访问模式下)

表 8-7 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

QSPI 控制器模式切换寄存器 (QSPI_MODE)

位域	名称	访问	初值	描述
7:1	-	-	-	保留
0	Switch	R/W	0	0:切换为 SPI 控制器; 1:切换为 QSPI 控制器

8.2.5 参数控制寄存器 (SFC_PARAM)

表 8-8 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 (Memory 访问模式下) 分频系数与 {spre, spr} 组合相同, 详见表 9-7
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。

QSPI 控制器模式时序寄存器 (QSPI_TIMING)

位域	名称	访问	初值	描述
7:4	-	-	-	保留
3:0	rx_sample_dly	R/W	0	读数据采样延迟配置 读数据采样阶段相对于当前采样沿, 推迟指定时钟周期延迟 (该时钟周期为 QSPI 控制器时钟)

8.2.6 片选控制寄存器 (SFC_SOFTCS)

表 8-9 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

8.2.7 时序控制寄存器 (SFC_TIMING)

表 8-10 SPI 时序控制寄存器 (SFC_TIMING)

位域	名称	访问	初值	描述
7	dly_mode	R/W	0	spi 接口模式控制, 为 1 时开启采样延迟模式且采样与发送时机错开半周期
6	samp_dly_fmc[4]	R/W	0	采样延迟, 用于调整 SPI FLASH 读引擎在 tFAST 配置为 1 时的读延迟
5	samp_dly_spi[4]	R/W	0	采样延迟, 用于调整 SPI 主控制器在 dly_mode 配置为 1 时的读延迟

位域	名称	访问	初值	描述
4	4byte_ad_en	R/W	0	地址扩展模式使能（4 字节地址） 1:使能 4 字节地址模式，扩展大容量设备 0:默认 3 字节地址模式，最高支持 16MB
3	quad_io	R/W	0	4 线 IO 模式使能，1 有效（注：SPI0 控制器仅支持 4 线 IO/Flash 读设备模式，QSPI0 支持 IO 读写模式）
2	tFAST	R/W	0	SPI flash 读采样模式 0: SCK 上沿采样，间隔半个 SPI 周期 1: SCK 下沿采样，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

8.2.8 时序控制寄存器 2 (SFC_TIMING2)

表 8-11 SPI 时序控制寄存器 2 (SFC_TIMING2)

位域	名称	访问	初值	描述
7:4	samp_dly_fmc[3:0]	R/W	0	采样延迟，用于调整 SPI FLASH 读引擎在 tFAST 配置为 1 时的读延迟 0: 无延迟 1: 延迟 1 个时钟周期 2: 延迟 2 个时钟周期
3:0	samp_dly_spi[3:0]	R/W	0	采样延迟，用于调整 SPI 主控制器在 dly_mode 配置为 1 时的读延迟 0: 无延迟 1: 延迟 1 个时钟周期 2: 延迟 2 个时钟周期

8.2.9 自定义控制寄存器 (CTRL)

表 8-12 自定义控制寄存器 (CTRL)

位域	名称	访问	初值	描述
7:4	nbyte	RW	0	一次传输的字节数
3:2	reserve	RW	0	保留
1	nbmode	RW	0	多字节传输模式
0	start	RW	0	开始多字节传输，完成后自动清零

8.2.10 自定义命令寄存器 (CMD)

表 8-13 自定义命令寄存器 (CMD)

位域	名称	访问	初值	描述
7:0	cmd	RW	0	设置发送给 spi flash 的命令

8.2.11 自定义数据寄存器 0 (BUF0)

表 8-14 自定义数据寄存器 0 (BUF0)

位域	名称	访问	初值	描述
7:0	buf0	RW	0	向 SPI 发送写命令时，该寄存器配置发送的第一个字节的数据；向 SPI 发送读命令时，该寄存器存储第一个读回来的数据。

8.2.12 自定义数据寄存器 1 (BUF1)

表 8-15 自定义数据寄存器 1 (BUF1)

位域	名称	访问	初值	描述
7:0	buf1	RW	0	向 SPI 发送写命令时，该寄存器配置发送的第二个字节的数据；向 SPI 发送读命令时，该寄存器存储第二个读回来的数据。

8.2.13 自定义时序寄存器 0 (TIMER0)

表 8-16 自定义时序寄存器 0 (TIMER0)

位域	名称	访问	初值	描述
7:0	time0	RW	0	自定义命令所需时间值的低 8 位

8.2.14 自定义时序寄存器 1 (TIMER1)

表 8-17 自定义时序寄存器 1 (TIMER1)

位域	名称	访问	初值	描述
7:0	Time1	RW	0	自定义命令所需时间值的中间 8 位

8.2.15 自定义时序寄存器 2 (TIMER2)

表 8-18 自定义时序寄存器 2 (TIMER2)

位域	名称	访问	初值	描述
7:0	Time2	RW	0	自定义命令所需时间值的高 8 位

8.2.16 QSPI 配置空间使能寄存器 (QSPI_SPACE_EN)

表 8-19 QSPI 配置空间使能寄存器 (QSPI_SPACE_EN)

位域	名称	访问	初值	描述
7:1	—	—	—	保留
0	Rssr	R/W	0	QSPI 控制器配置空间使能位 0:配置寄存器空间为 SPI flash/IO 寄存器空间; 1:配置寄存器空间为 QSPI IO 寄存器空间。

8.3 接口时序

8.3.1 SPI 主控制器接口时序

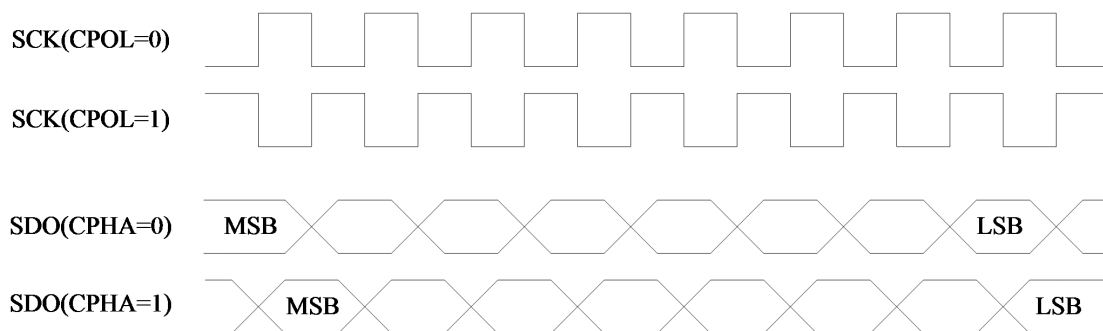


图8-2 SPI 主控制器接口时序

8.3.2 SPI Flash 访问时序

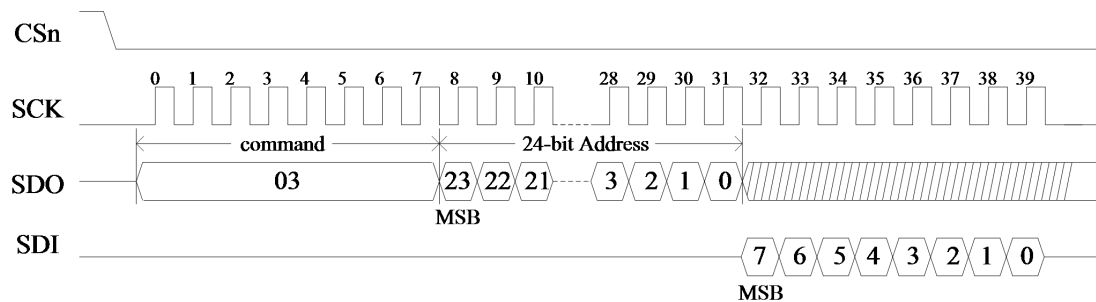


图8-3 SPI Flash 标准读时序

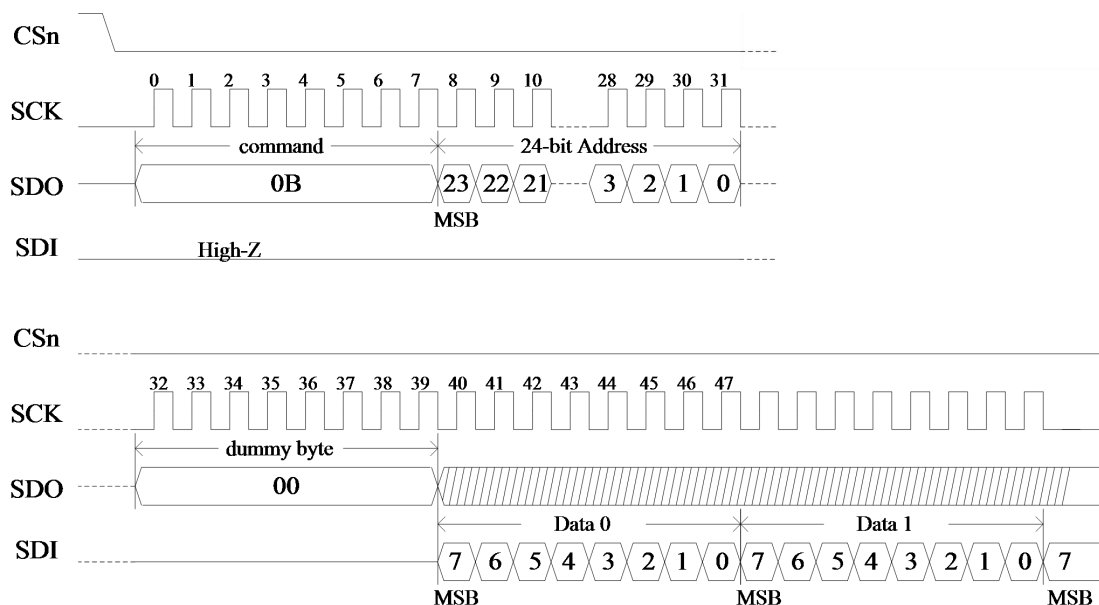


图8-4 SPI Flash 快速读时序

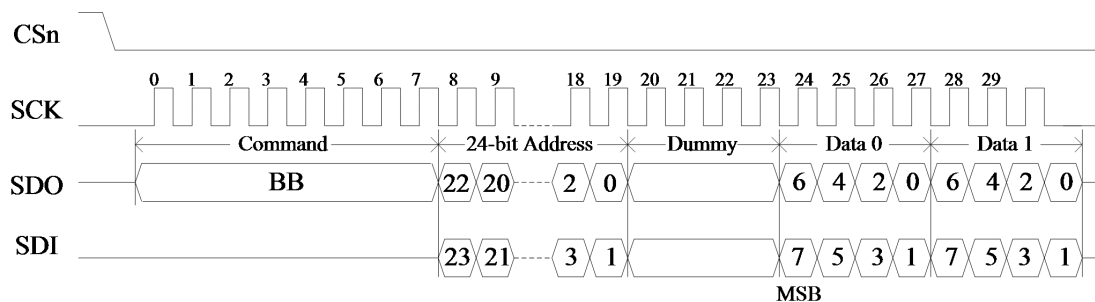


图8-5 SPI Flash 双向 I/O 读时序

8.4 使用指南

8.4.1 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。

8.4.2 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。

8.4.3 混合访问 SPI Flash 和 SPI 主控制器

1. 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。

8.4.4 混合访问 SPI 主控制器和 QSPI 主控制器

控制器复位后自动切换为 SPI flash/I/O 控制器模式, 按照原寄存器配置空间的软件无需更改即可配置使用。若使用 QSPI I/O 控制器功能, 需要首先配置 QSPI 配置空间使能寄存器(QSPI_SPACE_EN)，将寄存器配置空间切换为 QSPI 控制器配置空间。再通过配置 QSPI 控制器模式切换寄存器(QSPI_MODE)，将控制器切换至 QSPI 控制器模式。

为避免设备通信状态出错，以上切换操作均需通过读取 SPI/QSPI 当前状态并确认为 IDLE 时，再进行模式切换。

9 SPI-IO 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。

9.1 SPI 控制器结构

龙芯 2P0300 共集成 3 个 SPI 控制器，其中 SPI0 支持 IO/memory 两种空间访问，SPI1~2 仅支持 IO 空间访问。2P0300 芯片 2 个 SPI-IO 控制器空间分布，如下表。

表 9-1 SPI 控制器地址空间分布

地址空间	名称	大小
0x1420,3000 - 0x1420,3fff	MAIN SPI IO	4KB
0x1520,3000 - 0x1520,3fff	SCAN SPI IO	4KB

9.2 配置寄存器

表 9-2 SPI-IO 配置寄存器列表

偏移	名称	描述
0x00	CR1	控制寄存器 1
0x04	CR2	控制寄存器 2
0x08	CR3	控制寄存器 3
0x0c	CR4	控制寄存器 4
0x10	IER	中断寄存器
0x14	SR1	状态寄存器 1
0x18	SR2	状态寄存器 2
0x20	CFG1	配置寄存器 1
0x24	CFG2	配置寄存器 2
0x28	CFG3	配置寄存器 3
0x30	CRC1	CRC 寄存器 1
0x34	CRC2	CRC 寄存器 2
0x40	DR	数据寄存器

9.2.1 控制寄存器 1 (CR1)

表 9-3 SPI 控制寄存器 1 (CR1)

位域	名称	访问	初值	描述
31: 9	-	-	0	保留
8	SSREV	R/W	0	SS REVerse- 片选信号翻转 0: 片选 (SS) 输入输出正常 1: 片选 (SS) 输入输出翻转
7:3	-	-	0	保留

位域	名称	访问	初值	描述
2	AUTOSUS	R/W	0	AUTOSUSpend- 自动挂起 0: 在当前传输结束后挂起 1: 在当前传输结束后重新向 CTSIZE 载入TSIZE, 继续传输 注: 仅在主模式下使用。当前传输结束即: CTSIZE=0 且当前帧结束
1	CSTART	R/W	0	Control START of transfer- 传输开始控制 0: 在当前帧结束后挂起 1: 开始传输 注: 仅在主模式下使用。当 AUTOSUS=1 时, 会在 CTSIZE=0 时清 0
0	SPE	R/W	0	SPi Enable - SPI 使能。 0: 禁止传输, 此时 DR无法访问; 1: 可进行SPI 传输, 此时 CFGx 和 CRCx无法写入。

9.2.2 控制寄存器 2 (CR2)

表 9-4 SPI 控制寄存器 2 (CR2)

位域	名称	访问	初值	描述
31: 16	-	-	0	保留
15	TXDMAEN	R/W	0	TX DMA ENable - 发送侧 dma 使能 0: 数据通过 apb 写入DR 传输 1: 数据通过 dma自动传输, 此时 DR无法写入
14: 10	-	-	0	保留
9: 8	TXFTHLV	R/W	0	TX THreshold LeVel - 发送侧阈值 当 Txfifo 内的空余空间不少于 (TXFTHLV+1) 时, 发出 TXA 标志 (这个值会自动向上与 DSIZE 对齐)
7	RXDMAEN	R/W	0	Receive Side DMA ENable - 接收侧 dma 使能 0: 数据通过 apb 读取 DR 传输 1: 数据通过 dma自动传输, 此时 DR无法读取
6: 2	-	-	0	保留
1: 0	RXFTHLV	R/W	0	RX THreshold LeVel - 接收侧阈值 当 Rxfifo 内的数据不少于 (RXFTHLV+1) 时, 发出 RXA 标志。(这个值会自动向上与 DSIZE 对齐)

9.2.3 控制寄存器 3 (CR3)

表 9-5 SPI 控制寄存器 3 (CR3)

位域	名称	访问	初值	描述
31: 16	-	-	0	保留
15: 0	TSIZE	R/W	0	Transfer SIZE of frame - 单次传输帧数 当上一次传输结束 (CTSIZE=0), 继续下一次传输则会载入TSIZE。 即单次传输的帧数为 (TSIZE + 1)

9.2.4 控制寄存器 4 (CR4)

表 9-6 SPI 控制寄存器 4 (CR4)

位域	名称	访问	初值	描述
31: 16	–	–	0	保留
15: 0	CTSIZE	R/W	0	Counter of TSIZE – TSIZE 计数器 每次传输开始时会减 1, 表示本次传输剩余需要传输的帧数（不包括当前帧）。 减成 0 后会重新载入 TSIZE 中的值。

9.2.5 中断寄存器 (IER)

表 9-7 SPI 中断寄存器 (IER)

位域	名称	访问	初值	描述
31: 16	–	–	0	保留
15	EOTIE	R/W	0	EOT Int Enable – EOT 中断使能 使能后, 会在 SR1 中的 EOT 为 1 时产生中断
14: 12	–	–	0	保留
11	MODFIE	R/W	0	MODF Int Enable – MODF 中断使能 使能后, 会在 SR1 中的 MODF 为 1 时产生中断
10	CRCEIE	R/W	0	CRCE Int Enable – CRCE 中断使能 使能后, 会在 SR1 中的 CRCE 为 1 时产生中断
9	UDRIE	R/W	0	UDR Int Enable – UDR 中断使能 使能后, 会在 SR1 中的 UDR 为 1 时产生中断
8	OVRIE	R/W	0	OVR Int Enable – OVR 中断使能 使能后, 会在 SR1 中的 OVR 为 1 时产生中断
7	SUSPIE	R/W	0	SUSP Int Enable – SUSP 中断使能 使能后, 会在 SR1 中的 SUSP 为 1 时产生中断
6	–	–	0	保留
5	TXEIE	R/W	0	TXE Int Enable – TXE 中断使能 使能后, 会在 SR1 中的 TXE 为 1 时产生中断
4	RXEIE	R/W	0	RXE Int Enable – RXE 中断使能 使能后, 会在 SR1 中的 RXE 为 1 时产生中断
3	–	–	0	保留
2	DXAIE	R/W	0	DXA Int Enable – DXA 中断使能 使能后, 会在 SR1 中的 DXA 为 1 时产生中断
1	TXAIE	R/W	0	TXA Int Enable – TXA 中断使能 使能后, 会在 SR1 中的 TXA 为 1 时产生中断
0	RXAIE	R/W	0	RXA Int Enable – RXA 中断使能 使能后, 会在 SR1 中的 RXA 为 1 时产生中断

9.2.6 状态寄存器 1 (SR1)

表 9-8 SPI 状态寄存器 1 (SR1)

位域	名称	访问	初值	描述
31: 16	–	–	0	保留

位域	名称	访问	初值	描述
15	EOT	R	0	End Of Transfer - 发送结束 若一次发送结束（即 CTSIZE=0 且当前帧结束），置 1。通过向该位写 1 来清零 硬件置位软件清零
14: 12	—	—	0	保留
11	MODF	R	0	MODe Fault - 模式错误 若在多主模式下出现多主冲突，置 1。通过向该位写 1 来清零 硬件置位软件清零
10	CRCE	R	0	CRC Err - CRC 错误 若接收侧收到的 crc 比较错误，置 1。通过向该位写 1 来清零 硬件置位软件清零
9	UDR	R	0	UnDer Run - 发送侧数据下溢 若 txfifo 已空后仍然发送数据时，置 1。通过向该位写 1 来清零 硬件置位软件清零
8	OVR	R	0	OVer Run - 接收侧数据上溢 若 rxfifo 已满后仍然接收数据时，置 1。通过向该位写 1 来清零 硬件置位软件清零
7	SUSP	R	0	SUSPend - 挂起状态 仅在主模式下，挂起时置 1 硬件置位硬件清零
6	—	—	0	保留
5	TXE	R	0	TX fifo Empty - 发送侧 fifo 为空 当 txfifo 为空时置 1 硬件置位硬件清零
4	RXE	R	0	RX fifo Empty - 接收侧 fifo 为空 当 rxfifo 为空时置 1 硬件置位硬件清零
3	—	—	0	保留
2	DXA	R	0	DX dr Available - 两侧数据寄存器可访问 $dxa = rxa \& txa$ 硬件置位硬件清零
1	TXA	R	0	TX dr Available - 发送侧数据寄存器可访问 $txa = (txflv + (txfthlv + 1)) \leq 4$ 硬件置位硬件清零
0	RXA	R	0	RX dr Available - 接收侧数据寄存器可访问 $rxr = (rxflv - (rxfthlv + 1)) \geq 0$ 硬件置位硬件清零

9.2.7 状态寄存器 2 (SR2)

表 9-9 SPI 状态寄存器 2 (SR2)

位域	名称	访问	初值	描述
31: 11	—	—	0	保留
10: 8	TXFLV	R	0	TX Fifo LeVel - 发送侧 fifo 级别 表示txfifo 中储存的字节数，0x0~0x4
7: 3	—	—	0	保留
2: 0	RXFLV	R	0	RX Fifo LeVel - 接收侧 fifo 级别 表示rxfifo 中储存的字节数，0x0~0x4

9.2.8 配置寄存器 1 (CFG1)

表 9-10 SPI 配置寄存器 1 (CFG1)

位域	名称	访问	初值	描述
31: 13	—	—	0	保留
12: 8	DSIZE	R/W	7	Data SIZE of frame - 单帧数据大小 单帧数据大小为(DSIZE + 1) bits 注: 至少为 3, 即单帧数据大小 ≥ 4 bits
7	LSBFRST	R/W		LSB transfer FiRST - LSB 先发模式 0: 每一帧从 MSB 开始先发 1: 每一帧从 LSB 开始先发
6: 2	—	—	0	保留
1	CPHA	R/W	0	Clock PHase - 时钟相位 0: 从 SCK 第一个沿开始采样数据 1: 从 SCK 第二个沿开始采样数据
0	CPOL	R/W	0	Clock POLarity - 时钟极性 0: 空闲时 SCK 为低电平 1: 空闲时 SCK 为高电平

9.2.9 配置寄存器 2 (CFG2)

表 9-11 SPI 配置寄存器 2 (CFG2)

位域	名称	访问	初值	描述
31: 16	—	—	0	保留
15: 8	BRINT	R/W	2	Baut Rate ratio INTegeR - 波特率系数整数部分 主时钟频率/波特率 = (BRINT + BRDEC*2 ⁽⁻⁶⁾) 该比率应不小于 2.0
7:2	BRDEC	R/W	0	Baut Rate ratio DECimal - 波特率系数小数部分 注: 详见BRINT
1: 0	—	—	0	保留

9.2.10 配置寄存器 3 (CFG3)

表 9-12 SPI 配置寄存器 3 (CFG3)

位域	名称	访问	初值	描述
31: 10	—	—	0	保留

位域	名称	访问	初值	描述
9: 8	SSMODE	R/W	0	SS MODE - 片选模式 0: 一般模式--主模式下 SS 做输出: SS = (! SPE); 从模式下 SS 做输入: 收到 SS=0 时有效 1: 软件模式--主模式下 SS 无效: PIN_SS 可用作其他 IO; 从模式下 SS 无效: 默认收到内部 SS=0 2: 多主模式--主模式下 SS 做输入: 当输入为 0 时产生MODF 错误; 从模式下 SS 做输入: 收到 SS=0 时有效 注: SS 输出和输入时可用SSREV 反向, 软件模式的从模式下, SSREV 置 1 可使内部 SS=1
7: 4	-	-	0	保留
3	DOE	R/W	1	Data Output Enable - 数据输出使能 0: 数据输出禁止, 输出的 pin 脚可做其他 IO 1: 数据输出使能
2	DIE	R/W	1	Data Input Enable - 数据输入使能 0: 数据输入禁止, 输入的 pin 脚可做其他 IO 1: 数据输入使能
1	DIOSWP	R/W	0	Data In/Output SWaP - 数据输入输出交换 0: 主模式下 MISO 输入, MOSI 输出; 从模式下 MOSI 输入, MISO 输出 1: 主模式下 MOSI 输入, MISO 输出; 从模式下 MISO 输入, MOSI 输出
0	MSTR	R/W	0	MaSTeR - 主模式 0: SPI 从模式 1: SPI 主模式

9.2.11 CRC 寄存器 1 (CRC1)

表 9-13 SPI CRC 寄存器 1 (CRC1)

位域	名称	访问	初值	描述
31: 1	CRCPOLY	R/W	0	CRC POLYnomial - crc 多项式 表示crc 多项式的高位, 即抹除了最低位的 crcpoly, 而实际的多项式为{CRCPOLY, 1'b1} 目的是使用最低位作为 CRCEN, 因为只有在多项式最低位为 1 时 crc 才有意义
0	CRCEN	R/W	0	CRC ENable - crc 使能 表示本次传输使用crc 模式

9.2.12 CRC 寄存器 2 (CRC2)

表 9-14 SPI CRC 寄存器 2 (CRC2)

位域	名称	访问	初值	描述
31: 13	-	-	0	保留

位域	名称	访问	初值	描述
12:8	CRCSIZE	R/W	0	CRC SIZE - crc大小 表示一帧 crc 的数据大小，可以与数据大小不同 为(CRCSIZE+1)
7: 2	-	-	0	保留
1	TCRCINI	R/W	0	Tx CRC INItial value - 发送侧 crc 初始值 0: 表示txcrc 的初始值为全 0; 1: 表示txcrc 的初始值为全 1;
0	RCRCINI	R/W	0	Rx CRC INItial value - 接收侧 crc 初始值 0: 表示rxcrc 的初始值为全 0; 1: 表示rxcrc 的初始值为全 1;

9.2.13 数据寄存器 (DR)

表 9-15 SPI 数据寄存器 (DR)

位域	名称	访问	初值	描述
31: 0	DR	R/W	0	Data Reg - fifo 的读写接口: 读此寄存器将读出 RXFIFO 的数据，读出的数据字节数由 RXFTHLV 决定; 写此寄存器将把数据写入 TXFIFO，写入的数据字节数由 apb_wstrb 决定。

9.3 功能描述

9.3.1 PIN 脚配置

相关配置位:

CFG3 - MSTR、DIOSWP、DIE、DOE

可通过组合上述配置位，控制数据在 SCK、MISO、MOSI 两个 pin 脚上的传输方向:

表 9-16 PIN 脚配置位和流动方向关系

MSTR	DIOSWP	DIE	DOE	SCK	MISO	MOSI
0	0	1	1	IN	OUT	IN
0	0	1	0	IN	X	IN
0	0	0	1	IN	OUT	X
0	1	1	1	IN	IN	OUT
0	1	1	0	IN	IN	X
0	1	0	1	IN	X	OUT
1	0	1	1	OUT	IN	OUT
1	0	1	0	OUT	IN	IN
1	0	0	1	OUT	X	OUT
1	1	1	1	OUT	OUT	IN
1	1	1	0	OUT	X	IN
1	1	0	1	OUT	OUT	X

注: I 表示该脚接收 spi 的串行输入数据, 0 表示该脚发生时 spi 的串行输出数据, X 表示该脚无数据通路, 可用作 GPIO。

举例几种常见的几种配置：

表 9-17 PIN 脚常见配置

名称	主/从	传输状态	MSTR	DIOSWP	DIE	DIO	连接
全双工	主	双向	1	0	1	1	主 SCK-从 SCK
	从	双向	0	0	1	1	主 MISO-从 MISO 主 MOSI-从 MOSI
半双工	主	主到从	1	0	0	1	主 SCK-从 SCK 主 MOSI-从 MISO
		从到主	1	1	1	0	
	从	主到从	0	1	1	0	
		从到主	0	0	0	1	
单工（仅主到从）	主	主到从	1	0	0	1	主 SCK-从 SCK
	从	主到从	0	0	1	0	主 MOSI-从 MOSI
单工（仅从到主）	主	从到主	1	0	1	0	主 SCK-从 SCK
	从	从到主	0	0	0	1	主 MISO-从 MISO

9.3.2 SS 模式

相关配置位：

CFG3 - SSMODE、MSTR

CR1 - SSREV

通过配置上述配置位，可实现不同的 SS 模式。

表 9-18 PIN 脚配置位和流动方向关系

SSMODE	MSTR	SSREV	SS 行为
0：一般模式	0	0	SS 脚输入，且仅为低电平时有效
0：一般模式	0	1	SS 脚输入，且仅为高电平时有效
0：一般模式	1	0	SS 脚输出，ss_o = !SPE
0：一般模式	1	1	SS 脚输出，ss_o = SPE
1：软件模式	0	0	SS 脚无数据，内部 SS 永久无效
1：软件模式	0	1	SS 脚无数据，内部 SS 永久有效
1：软件模式	1	0	SS 脚无数据
1：软件模式	1	1	SS 脚无数据
2：多主模式	0	0	SS 脚输入，且仅为低电平时有效
2：多主模式	0	1	SS 脚输入，且仅为高电平时有效
2：多主模式	1	0	SS 脚输入，为低电平时触发 MODF（模式错误），MSTR 自动切换为 0
2：多主模式	1	1	SS 脚输入，为低电平时触发 MODF（模式错误），MSTR 自动切换为 0

9.3.3 波特率

相关配置位：

CFG2 - BRDEC、BRINT

BRDEC 和 BRINT 可组合成波特率比率，其中 BRINT 为整数部分，BRDEC 为小数部分：

主时钟频率/波特率 = (BRINT + BRDEC*2⁻⁶)

且这个比率不应小于 2.0。

该波特率为开始传输后理论上的频率，可能会出现 a) 传输挂起，sck 停止；b) 在不能被 2 整除时占空比不一定为 0.5。

注：波特率的含义为单位时间传输的 bit 数，也即 sck 的频率，单位为 Hz

9.3.4 传输格式

相关配置位：

CFG1 - CPOL、CPHA、LSBFRST、DSIZE

CFG1 中的 4 个配置能配置不同的传输格式：

CPOL：时钟极性

CPHA：时钟相位

LSBFRST：LSB 先发

DSIZE：单帧数据大小 = DSIZE + 1，(31 > DSIZE > 3)。

9.3.5 FIFO 阈值

相关配置位：

CR2 - RXFTHLV、TXFTHLV

RXFTHLV、TXFTHLV 可用于控制 FIFO 阈值，用于触发 RXA、TXA 标志。

标志可能会 a) 使 SR 中 RXA、TXA 位置 1，b) 触发中断，c) 触发 dma 请求。

当 rxfifo 内数据不少于 (RXFTHLV + 1) 字节时，发出 RXA 标志；当 txfifo 内空间不少于 (TXFTHLV + 1) 字节时，发出 TXA 标志；

注：在判断是否触发标志时，(RXFTHLV + 1) 和 (TXFTHLV + 1) 会自动向上与 DSIZE 对应的字节数对齐。

例：DSIZE = 11，每帧占 fifo 空间 2 字节。若 RXFTHLV=2，即 (RXFTHLV + 1)=3，则会向上对其至 4：当 rxfifo 内数据不少于 4 字节时发出 RXA 标志。

9.3.6 FIFO 中数据储存格式和访问格式

相关配置位：

CFG1 - DSIZE

CR2 - RXDMAEN、TXDMAEN

储存格式 DSIZE：

0x00 ~ 0x07：高位补 0 扩充成 1 字节

0x08 ~ 0x0F：高位补 0 扩充成 2 字节，低位字节先存

0x10 ~ 0x1F：高位补 0 扩充成 4 字节，低位字节先存

RXDMAEN、TXDMAEN 两个配置位决定了访问 RXFIFO 和 TXFIFO 的访问方式和访问格式：

表 9-19 FIFO 访问格式

DMAEN	RX/TX	访问方式	访问数据格式
0	RX	apb 总线向 DR 读数	由 RXFTHLV 决定，在小于阈值的字节数中，选择 1/2/4 字节格式中最大的格式读出
0	TX	apb 总线向 DR 读数	有 apb 总线的 wstrb 决定，其值只能为 0x1/0x3/0xf，对应 1/2/4/字节格式，其他值可能会出现错误
1	RX	dma 自动通过 axi 总线将 fifo 中的数写到指定地址	由 dma 中的配置位 PSIZE_R 决定，0x0/0x1/0x2 分别对应 1/2/4 字节格式
1	TX	TX dma 自动通过 axi 总线将指定地址的数读到 fifo 中	由 dma 中的配置位 PSIZE_T 决定，0x0/0x1/0x2 分别对应 1/2/4 字节格式

根据访问格式大小，可将数据按先存的字节放低位的方式取出数据。

9.3.7 单次传输帧数

相关配置位：

CR3 - TSIZE

CR4 - CTSIZE

TSIZE 实际表示（预载入单次传输帧数-1），CTSIZE 实际表示剩余传输帧数。

当下一帧即将开始时：若 CTSIZE!=0，CTSIZE 将会减 1；若 CTSIZE==0，则本次传输结束，并载入 TSIZE 中的值来进行下一次传输。

可以在恰当的时机，中途向 CTSIZE 写入值（0 或非 0），来达成提前结束本次传输或延长本次传输的目的。

注：一次传输结束时，若开启了 crc 功能，则会进行一次 crc 传输。

9.3.8 开始与挂起

相关配置位：

CR1 - CSTART、AUTOSUS

在 SPE=1 时，且主模式下可以配置上述配置位来控制 SPI 传输的启动和挂起

当 CSTART==1 时，将立即开始传输；

当 CSTART==0 时，将在当前帧结束时挂起传输。

当 AUTOSUS==0 时，在一次传输结束后，将立即开始下一次传输；

当 AUTOSUS==1 时，在一次传输结束后将自动挂起。

10 I²C 控制器

10.1 概述

I²C(Inter-integrated Circuit)总线接口连接微控制器和串行 I²C 总线。

它提供多主机功能，控制所有 I²C 总线特定的时序、协议、仲裁和定时。支持标准和快速两种模式。

根据特定设备的需要，可以使用 DMA 以减轻 CPU 的负担。

龙芯 2P0300 芯片 2 个 I²C 控制器空间分布，如下表。

表 10-1 I²C 控制器地址空间分布

地址空间	名称	大小
0x1420, 2000 - 0x1420, 2fff	MAIN I ² C 寄存器空间	4KB
0x1510, 2000 - 0x1520, 2fff	PRT I ² C 寄存器空间	4KB

10.2 主要特性

I²C 主控制器的结构，主要模块有，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。

- 1) 多主机功能：
 - a. 该模块既可做主设备也可做从设备；
- 2) I²C 主设备功能：
 - a. 产生时钟；
 - b. 产生起始和停止信号；
- 3) I²C 从设备功能：
 - a. 可编程的 I²C 地址检测；
 - b. 停止位检测；
- 4) 产生和检测 7 位地址和广播呼叫；
- 5) 支持不同的通讯速度：
 - a. 标准模式（高达 100 kHz）；
 - b. 快速模式（高达 400 kHz）；
- 6) 状态标志：
 - a. 发送器/接收器模式标志；
 - b. 字节发送结束标志；
 - c. I²C 总线忙标志；

- 7) 错误标志：
 - a. 主模式时的仲裁丢失；
 - b. 地址/数据传输后的应答 (ACK) 错误；
 - c. 检测到错位的起始或停止条件；
 - d. 禁止拉长时钟功能时的上溢或下溢；
- 8) 1 个中断输出：
 - a. 用于地址/数据通讯成功和错误；
- 9) 可选的拉长时钟功能：
 - a. 具单字节缓冲器的 DMA。

10.3 功能描述

10.3.1 模式选择

控制器可以工作在从发送、从接收、主发送、主接收等 4 种模式之一。其中无论主从，发送均表示将数据写到 SDA 线，接收均表示从 SDA 线获取数据。

控制器默认处于从模式，在主动发起“开始”命令或“恢复”命令时，自动切换到主模式；当仲裁丢失或总线出现产生“停止”条件时，自动切换回从模式。

控制器的时钟与 APB 总线时钟相同，由芯片 SYS/DDR-PLL 中 APB 设备时钟提供。

在从模式下，需要配置时钟频率 (CR2_FREQ)，用于产生合适的建立和保持时间。而在主模式下，数据线的建立和保持时间总是为一半的时钟线低电平时间。

10.3.2 从模式

从模式下，当检测到“开始”条件时，从 SDA 线接收地址，接收完成后与从地址寄存器 (OAR) 77 或广播地址 (当 CR2_ENGC 有效时) 进行比对。

若地址不命中，控制器无视本次总线传输 (不会进行应答) 并等待下一个“开始”条件。

若地址命中，若允许应答 (CR1_ACK 有效) 则会产生 I²C 总线应答；产生应答后，硬件置起地址状态位 (SR1_ADDR)，若事件中断使能 (CR2_ITEVTEN) 有效，还会产生中断。

发送状态位 (SR2_TRA) 表示从设备处于发送还是接收状态。

10.3.2.1 从发送模式

从设备拉低 SCL 线，直到地址状态位清除且数据寄存器填入需要发送的第一个数据。填入第一个数据之前，数据寄存器处于发送空状态 (SR1_TXE)。移位寄存器从数据寄存器取出第一个字节，开始移位并发送，取出后数据寄存器再次进入发送空状态。

当第一个发送的数据完成了应答后，若数据寄存器仍然处于空状态，则触发字节传输

结束状态（SR_BTFF）并等待。在字节传输结束状态被清除前，控制器进入等待状态并拉低 SCL 线。

若完成应答后，数据寄存器不处于空状态，则取出下一个字节并发送。即在前一个数据完成传输前，若查询到 SR1_TXE 状态，即可写入数据寄存器以免进入 SR1_BTFF 状态导致总线空闲。

若发送的数据接收到的应答为无应答（NACK），表示主设备不再接收数据，此时应答失败状态（SR1_AFF）会置起，控制器回到空闲状态，等待下一个“开始”条件。

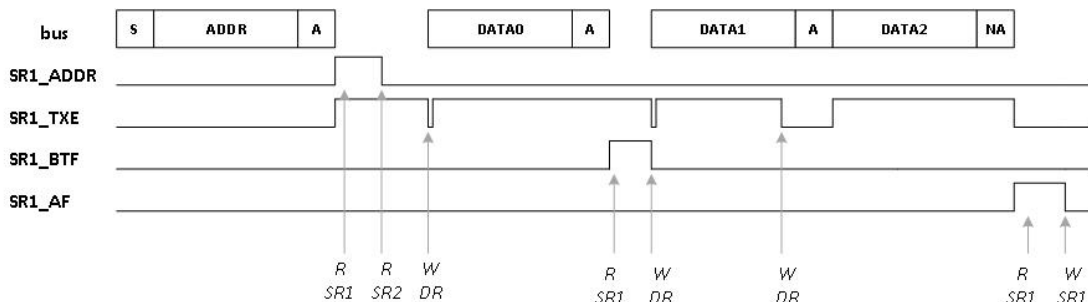


图10-1 从发送模式示意图

10.3.2.2 从接收模式

清除地址状态位后，从设备开始从 SDA 线接受数据放入移位寄存器。

收满一个字节后，若接收数据寄存器为空，则在应答周期完成后放入接收数据寄存器中，置起接收非空状态（SR1_RXNE），并继续进行下一个字节的接收。若应答配置（CR1_ACK）为无应答，则会结束本次传输，进入空闲状态等待下一个“开始”条件。

若接收数据寄存器不为空，则数据保留在移位寄存器中，并置起字节传输结束状态（SR1_BTFF）并等待。在字节传输结束状态被清除前，控制器进入等待状态并拉低 SCL 线。

当主设备发出“结束”条件时，控制器进入空闲状态，并置起停止状态位（SR1_STOPF）。

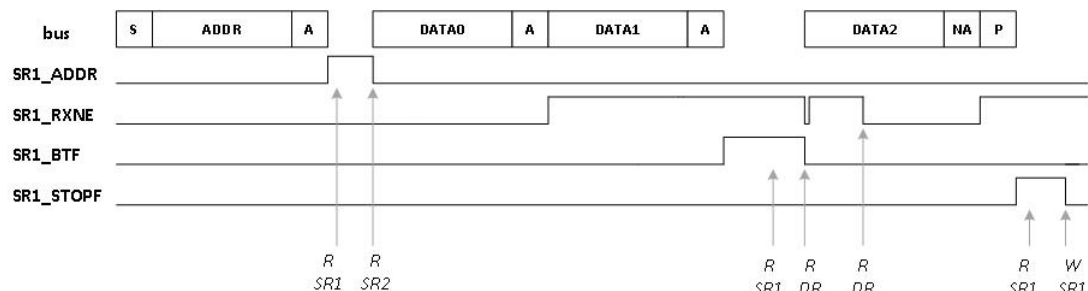


图10-2 从接收模式示意图

从接收模式存在一种可能：从设备还没有及时处理数据和停止条件，后续的传输就已经开始。这是由于单独的 SR1_RXNE 不会拉低 SCL，主设备可执行停止和新的开始。

需要注意，在图 12-2 中若最后一个字节未读取，则内部接收数据寄存器保持非空（即使 SR1_RXNE 已因写入 DR 或不在接收模式而被清除），在下一次接收模式时（无论主从），

会直接产生 SR1_BTIF 状态。也就是说，内部维护的接收非空状态（非 SR1_RXNE）只能由读 DR 而清除。

若 CR1_STOPF 也没有处理，则后续传输中会看到 SR1_STOPF 和 SR1_ADDR 同时置起。此时，若前一次访问为接收，则接收数据寄存器有前一次传输的数据。

为避免接收数据混乱，建议软件有足够快的处理速度，并及时读出接收的数据；处理时，先处理完 SR1_RXNE 再处理 SR1_STOPF；必要时，还应开启 DMA 方式保证数据接收。

10.3.3 主模式

主模式下，控制器初始化数据传输并提供时钟信号（SCL）。

当内部状态机接收到“开始”命令（CR1_START）或“恢复”命令（CR1_RECOVER）时，即进入主模式，从而支持在总线开始阶段即开始主设备仲裁检测。

10.3.3.1 主模式时钟生成

由于主模式需要产生时钟，故需要一些额外的配置：配置时钟控制寄存器（CCR）、配置上升时间寄存器（TRISE）。

其中，CCR 配置不同的时钟占空比和分频系数。TRISE 用来确定 SCL 上升时间的上限，在主模式发送 SCL 的上升沿后，等待 TRISE 时间再检查 SCL 输入的状态；若此时 SCL 输入为低，则表示从设备因为无法处理而拉低了 SCL，此时主模式的时钟发送暂停，直到 SCL 恢复为高。

等待 TRISE 的原因是，I²C 总线的上升沿依赖于电阻上拉，需要的事件较长。若主模式释放 SCL 后立刻进行检测，则可能由于上升速度不够而检测到错误的从设备拉低。最终导致时钟高电平事件变长，总线时钟周期不稳定。

10.3.3.2 开始条件

设置“开始”命令后，进入主模式（SR2_MSL），并生成总线的开始条件。“开始”命令配置后，硬件不会检查总线是否忙；软件应检查总线忙状态（SR2_BUSY），在非忙状态下设置“开始”命令，并检查是否存在仲裁丢失状态（SR1_ARLO）出现。

当成功生成总线的开始条件后，开始位状态（SR1_SB）置起。清除开始位状态的方法是：读取到开始位后，将从设备地址写到数据寄存器中。

10.3.3.3 从设备地址发送

移位寄存器从数据寄存器中取出从设备地址，开始发送到 SDA 线上。只支持 7 位地址模式，地址阶段发送的一个字节中，高 7 位为地址，最低位为发送或接收模式的选择。最低位为低时，将进行主发送模式；最低位为高时，将进行主接收模式。

当发送的地址在应答周期接收到有效应答时，会产生地址状态位；

若收到的为无应答，则会进行空闲状态等待新的命令（如再次开始或结束），此时

SCL 为低、SDA 为高。

与从模式类似，发送状态位（SR2_TRA）表示处于发送还是接收状态。

10.3.3.4 主发送模式

处理完地址状态位后，主设备通过移位寄存器将数据寄存器中的字节发送到 SDA 线上。

当需要发送数据，但发送数据寄存器没有有效的数时，会产生发送空状态（SR1_TXE）。若传输完一个数据字节后，仍出现发送空状态，则还会产生字节传输结束状态（SR1_BTF）。即第一个需要发送的字节未写入时，只有 TXE 状态，而后续字节未写入时，会有 BTF 状态。

等待 TXE 或 BTF 状态时，总线处于保持状态，即 SCL 为低，SDA 不变。

当需要结束传输时，写入“停止”命令（CR1_STOP）即可，主设备会在当前字节传输结束后发送停止条件。停止条件产生后，控制器自动返回从模式。

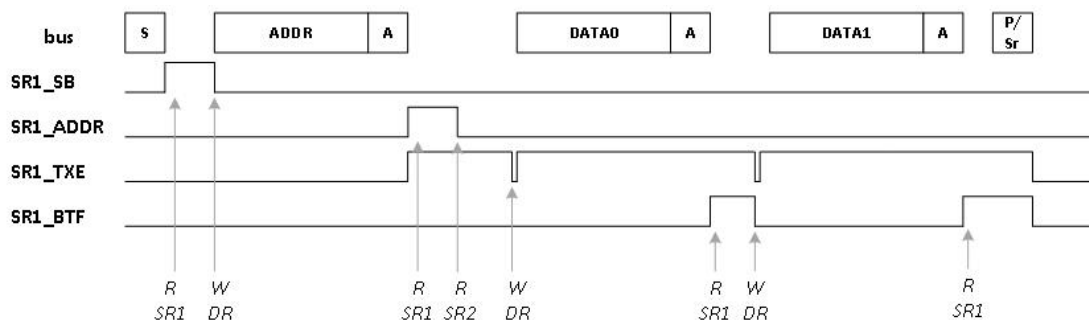


图10-3 主发送模式示意图

10.3.3.5 主接收模式

清除地址状态位后，从设备开始从 SDA 线接受数据放入移位寄存器。

收满一个字节后，若接收数据寄存器为空，则在应答周期完成后放入接收数据寄存器中，置起接收非空状态（SR1_RXNE），并继续进行下一个字节的接收。若应答配置（CR1_ACK）为无应答，则会结束本次传输，进入空闲状态等待下一个“开始”条件。

若接收数据寄存器不为空，则数据保留在移位寄存器中，并置起字节传输结束状态（SR1_BTF）并等待。在字节传输结束状态被清除前，控制器进入等待状态并拉低 SCL 线。

与从接收模式不同的是，主接收模式若需要结束传输，需要在最后一个字节的应答周期中给出无应答（NACK）条件。从设备收到无应答条件，释放总线，避免产生总线死锁现象。

发送完无应答条件后，控制器进入空闲状态，可接收再次开始或停止命令。开始或停止命令也可在最后一个字节传输过程中进行配置，传输完成后立刻执行。

为及时发送无应答条件，有下列注意事项：

- 1) 能确保在收到状态后、下一个字节传输过程中完成寄存器配置时：

接收到倒数第二个字节后，通过清除 CR1_ACK 配置位的方式发送无应答条件。若只需要传输一个字节，则在清除地址状态位后立刻清除应答配置位；事实上，由于应答配置位只对接收有效，在发送地址前后清除应答配置位均可达到相同目的。

2) 无法确保及时完成寄存器配置时：

这种条件下，需要在接收到倒数第三个字节后，就进行配置。当接收字节数大于 2 时，接收到倒数第三个字节后，不进行读出，直到倒数第二个字节也被接收在移位寄存器中；即此时接收数据寄存器中保存倒数第三个字节，移位寄存器中保存倒数第二个字节，此时 SR1_RXNE 和 SR1_BTF 状态位都置起；此时，清除应答配置位即可成功发送无应答条件。读取倒数第三个字节的同时，倒数第二个字节自动由移位寄存器进入接收数据寄存器，总线开始进行最后一个字节的传输。故在读取倒数第三个自己后配置开始或停止命令已经是安全的。

当接收字节数等于 2 时，情况有所不同。由于接收数据寄存器和移位寄存器共能接收 2 个字节，故需要特殊机制来保证最后一个自己的无应答。此时，需要使用应答位置配置位（CR1_POS）来获得延迟应答，CR1_POS 为 0 时，延迟应答为 0；CR1_POS 为 1 时，延迟应答会在下一次接收应答时更新为应答配置位（CR1_ACK）的值。故在地址状态处理的前后位置，即可配置 CR1_POS 并清除 CR1_ACK，此时 2 个字节的应答会分别为有应答和无应答。接收到 SR1_BTF 状态时，表示 2 个字节都接收完成，读取数据寄存器两次即可。重新开始或停止命令在收到 SR1_BTF 状态后进行配置。

当接收字节数为 1 时，在清除地址状态前即清除应答配置位，以免错过配置时间。

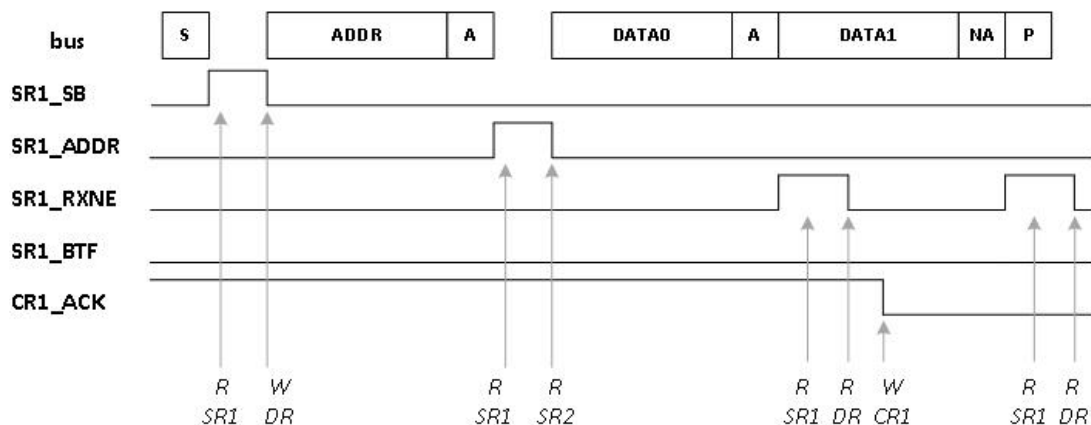


图10-4 主接收模式（及时）示意图

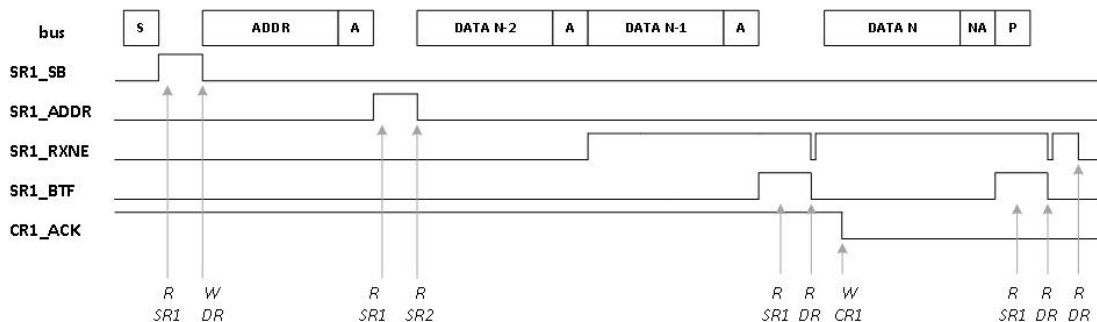


图10-5 主接收模式（非及时、3 字节）示意图

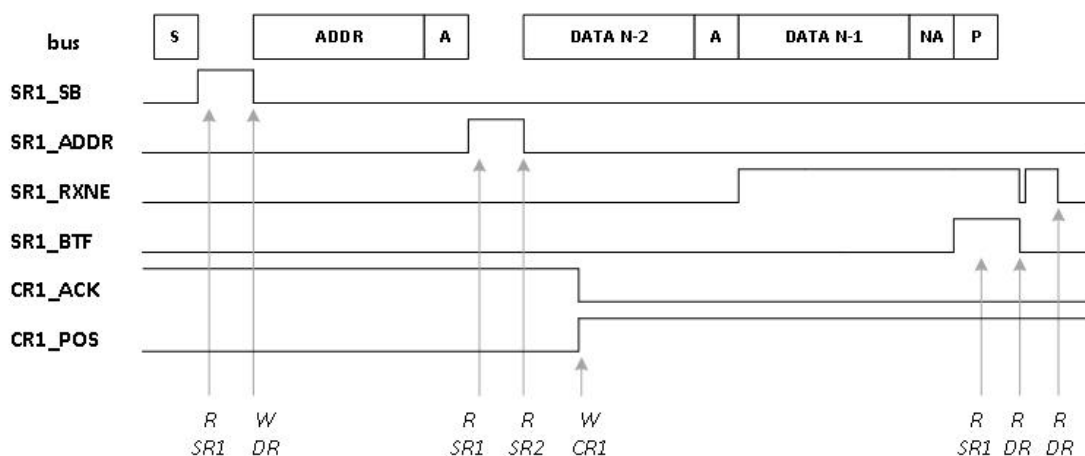


图10-6 主接收模式（非及时、2 字节）示意图

10.3.4 错误条件

下列为可能导致传输失败的错误条件。

1) 总线错误（SR1_BERR）

当控制器处于地址或数据传输状态时，总线上出现了开始或停止条件。在从模式中，当前传输数据被丢弃，总线被释放；错误的开始条件会被视为正常的重新开始，等待地址或停止；错误的停止条件会被视为正常的停止。在主模式中，当前传输继续，软件可依据状态进行操作；若软件不中断当前传输，则后续有可能出现仲裁丢失而停止。

2) 应答失败（SR1_AF）

产生应答失败时，表示发送的地址或数据未被接收。此时，内部状态回到空闲状态。从模式时，总线被释放，等待新的开始条件；主模式时，等待软件配置重新开始或停止命令。

3) 仲裁丢失（SR1_ARLO）

当主模式欲发送的 SDA 为高，而 SDA 线为低时，产生仲裁丢失状态，表明其他主设备赢得总线控制权。此时，总线被释放，控制器回到从模式，进入空闲状态。即使获胜的主设备访问的是本设备，也无法进行应答，需等待下一个开始条件。

4) 溢出错误 (SR1_OVR)

溢出错误分为上溢和下溢两种，都是由于从模式下时钟伸展特性被禁用 (CR1_NOSTRETCH) 时，上溢由于数据寄存器非空时仍接收写入，下溢由于数据寄存器非满时仍读出发送。

当产生上溢时，最后一个接收的字节被丢弃；软件应清除非空状态，并以一定方法使发送方重发。当产生下溢时，数据寄存器的字节将被发送两次；用户应保证接收方能识别并无视该错误。当需要发送第一个字节时，必须在数据阶段的第一个 SCL 上升沿之前写入数据寄存器。

当与不支持时钟伸展的主设备进行传输时，用户必须自行定义合理的校验方式，来识别可能的溢出错误；并尽量保证及时对数据进行读出或写入。

10.3.5 DMA 请求

当需要进行数据传输时，可使用 DMA 进行传输，使用时置起 DMA 使能位 (CR2_DMAEN)，同时控制发送和接收两个通道。发送请求连接到 DMA 通道 1，接收请求连接到 DMA 通道 2。开启 DMA 时，内部的数据寄存器空满状态（而非状态寄存器的状态）控制 DMA 请求信号。

其中，发送通道在内部 TXE 时置起请求信号。内部 TXE 在复位信号有效、控制器未使能时，以及总线的开始和停止条件后复位为 1；在写数据寄存器后置为 0；在内部加载到移位寄存器（包括发送字节和主模式发送地址）时置为 1。

接收通道在内部 RXNE 时置起请求信号。内部 RXNE 在复位信号有效、控制器为使能时复位为 0；在接收到字节（不含从模式接收地址）后且非 RXNE 时置为 1；在读数据寄存器后置为 0。

DMA 通道配置方法如下，配置完成后在需要开启时打开 DMA 使能。

- 1) 外设基地址设为 I²C 的数据寄存器地址
- 2) 内部基地址设为发送或接收缓冲区的地址
- 3) 方向依据发送或接收配置为外设为目标或外设为源
- 4) 缓冲区大小设为需要传输的字节数
- 5) 外设地址自增设为关闭
- 6) 内存地址自增设为开启
- 7) 外设数据大小设为字节
- 8) 内存数据大小设为字节
- 9) 模式设为普通
- 10) 优先级根据需要进行设定
- 11) 内存到内存开关设为关闭

主接收模式时，由于最后 1 个或 2 个字节需要特殊控制以正确发送 NACK，DMA 只能用于传输多于 2 个的字节，剩余字节必须用 PIO 方式进行。

10.4 中断

中断状态和对应的使能控制位如下表：

表 10-2 I²C 中断请求

中断事件	标记位	使能位
开始条件已发送（主模式）	SR1_SB	CR2_ITEVTEN
地址已发送（主模式）或地址命中（从模式）	SR1_ADDR	CR2_ITEVTEN
收到停止条件（从模式）	SR1_STOPF	CR2_ITEVTEN
字节传输结束	SR1_STF	CR2_ITEVTEN
接收数据寄存器非空	SR1_RXNE	CR2_ITEVTEN and CR2_ITBUFEN
发送数据寄存器空	SR1_TXE	CR2_ITEVTEN and CR2_ITBUFEN
总线错误	SR1_BERR	CR2_ITERREN
仲裁丢失（主模式）	SR1_ARLO	CR2_ITERREN
应答失败	SR1_AF	CR2_ITERREN
溢出	SR1_OVR	CR2_ITERREN

10.5 寄存器定义

表 10-3 I²C 控制器寄存器列表

地址偏移	寄存器名称	描述
0x00	I2C_CR1	I ² C 控制寄存器 1
0x04	I2C_CR2	I ² C 控制寄存器 2
0x08	I2C_OAR	I ² C 从地址寄存器
0x10	I2C_DR	I ² C 数据寄存器
0x14	I2C_SR1	I ² C 状态寄存器 1
0x18	I2C_SR2	I ² C 状态寄存器 2
0x1c	I2C_CCR	I ² C 时钟控制寄存器
0x20	I2C_TRISE	I ² C 上升时间寄存器

10.5.1 I²C 控制寄存器（I2C_CR1）

偏移量： 0x00

复位值： 0x00000000

表 10-4 I²C 控制寄存器

位域	名称	访问	描述
31:16	Reserved	—	—
15	CR1_SWRST	RW	软复位 0：未复位； 1：复位。 该复位只复位内部控制逻辑，不复位寄存器，应配合 CR1_PE 使用软件可配置和清除该位。
14	CR1_RECOVER	RW	总线恢复命令 发出 9 个 SCL 时钟和一个停止条件，可用于解除死锁状态（SCL 为高，从设备将 SDA 拉低）。 软件可设置和清除该位，硬件会在发出停止条件后或 CR1_PE 为 0 时清除该位。

位域	名称	访问	描述
13:12	Reserved	—	—
11	CRI_POS	RW	数据接收的应答位置 0: CRI_ACK 位控制当前移位寄存器正在接收的字节的应答; 1: CRI_ACK 位控制将由移位寄存器接收的下一个字节的应答。 硬件行为是在接收字节的应答周期完成时, 将 CRI_ACK 表示的值保存到寄存器中作为应答值; 该位无效时, 应答寄存器复位为有应答; 故开启时, 第一个应答为有应答, 下一个为在第一个应答周期保存的 CRI_ACK 表示的值。 软件可设置和清除该位, 硬件会在 CRI_PE 为 0 时清除该位。
10	CRI_ACK	RW	应答使能 0: 无应答返回 (在应答拍为高电平); 1: 收到一个字节后返回有应答 (地址命中或接收到数据后, 在应答拍为低电平)。 软件可设置和清除该位, 硬件会在 CRI_PE 为 0 时清除该位。
9	CRI_STOP	RW	停止生成命令 0: 无命令; 1: 主模式下在当前字节传输 (不含地址阶段) 或开始/恢复命令完成后, 生成总线停止条件; 从模式下在当前字节传输完成后 (不含地址阶段) 释放总线, 回到空闲状态。 停止命令在主接收模式未输出无应答条件时存在总线状态不确定的风险。 软件可设置和清除该位, 硬件会在总线停止条件后清除该位。
8	CRI_START	RW	开始生成命令 0: 无命令; 1: 主模式下在当前字节传输 (不含地址阶段) 后, 生成重复开始条件; 从模式下在当前字节传输 (不含地址阶段) 后或空闲时, 生成开始条件并进入主模式。 开始命令无视总线状态, 软件应自行检查总线状态 (通过 SR2_BUSY) 并准备处理可能的仲裁丢失错误。 软件可设置和清除该位, 硬件会在总线开始条件后或 CRI_PE 为 0 时清除该位。
7	CRI_NOSTRETCH	RW	时钟伸展无效 (从模式) 0: 时钟伸展有效; 1: 时钟伸展无效 软件可配置和清除该位。
6	CRI_ENGC	RW	广播呼叫使能 使能广播呼叫, 地址 0x00 将得到应答。 软件可配置和清除该位。
5:1	Reserved	—	—
0	CRI_PE	RW	控制器使能 1: 使能。 在工作中清除该位会立刻重置控制器状态机, 无视总线。 软件可配置和清除该位。

10.5.2 I²C 控制寄存器 2 (I2C_CR2)

偏移量: 0x04

复位值: 0x00000000

本寄存器不含硬件置位或复位。

表 10-5 I²C 控制寄存器 2

位域	名称	访问	描述
31:12	Reserved	—	—

位域	名称	访问	描述
11	CR2_DMAEN	RW	DMA 请求使能 1: 使能。
10	CR2_ITBUFEN	RW	缓冲类中断使能 1: 使能缓冲类中断, 包括 TXE 和 RXNE
9	CR2_ITEVTEN	RW	事件类中断使能 1: 使能事件类中断, 包括 SB、ADDR、STOPF、BTF、TXE、RXNE
8	CR2_ITERREN	RW	错误类中断使能 1: 错误类中断使能, 包括 BERR、ARLO、AF、OVR
7:0	CR2_FREQ	RW	设备时钟频率 表示 APB 接口时钟频率 (MHz), 用于生成数据建立和保持事件 (从模式)。

10.5.3 I²C 从地址寄存器 (I2C_OAR)

偏移量: 0x08

复位值: 0x00000000

本寄存器不含硬件置位或复位。

表 10-6 I²C 从地址寄存器

位域	名称	访问	描述
31:8	Reserved	—	—
7:1	OAR	RW	从设备地址 作为从模式地址阶段的[7:1]位地址进行地址命中的判断
0	Reserved	—	—

10.5.4 I²C 数据寄存器 (I2C_DR)

偏移量: 0x10

复位值: 0x00000000

表 10-7 I²C 数据寄存器

位域	名称	访问	描述
31:8	Reserved	—	—
7:0	DR	RW	字节数据寄存器 写入时, 写到发送数据寄存器 (TXR) 中; 读出时, 读出接收数据寄存器 (RXR) 的值。从模式接收的地址不会放入 RXR 中。

10.5.5 I²C 状态寄存器 (I2C_SR1)

偏移量: 0x14

复位值: 0x00000000

表 10-8 I²C 状态寄存器

位域	名称	访问	描述
31:12	Reserved	—	—

位域	名称	访问	描述
11	SR1_OVR	R	溢出状态位 软件向该位写 0 清除状态，硬件在 CR1_PE 为 0 时清除该位。 详细说明见错误条件。
10	SR1_AF	R	应答失败状态位 软件向该位写 0 清除状态，硬件在 CR1_PE 为 0 时清除该位。 详细说明见错误条件
9	SR1_ARLO	R	仲裁丢失状态位 软件向该位写 0 清除状态，硬件在 CR1_PE 为 0 时清除该位。 详细说明见错误条件。
8	SR1_BERR	R	总线错误状态位 软件向该位写 0 清除状态，硬件在 CR1_PE 为 0 时清除该位。 详细说明见错误条件。
7	SR1_TXE	R	发送数据寄存器空 发送模式时数据寄存器为空。 在地址阶段不会置位，收到无应答（NACK）条件时不会置位。 软件写数据寄存器后清除，硬件在开始或停止条件后或在 CR1_PE 为 0 时清除。 该位在写数据寄存器清除后可能很快再次置起，此时可再次写入。
6	SR1_RXNE	R	接收数据寄存器非空 接收模式时数据寄存器非空，即收到了有效的数。 在地址阶段不会置位，在出现仲裁丢失时不会置位。 软件写或读数据寄存器后清除，硬件在 CR1_PE 为 0 时清除。 该位在读数据寄存器清除后可能很快再次置起，因为移位寄存器中的数放到了接收数据寄存器，此时可再次读出。
5	Reserved	—	—
4	SR1_STOPF	R	从模式停止位检测 从模式时，在正常的应答后检测到总线停止条件后置位。 收到无应答（NACK）时不会置位。 要清除该位，软件在读出该位为 1 后，再写 I2C_CR1 寄存器；硬件在 CR1_PE 为 0 时清除。
3	Reserved	—	—
2	SR1_BTF	R	字节传输结束状态 当 CR1_NOSTRETCH 为 0 时可置位。接收模式时，接收到一个字节且接收数据寄存器非空时置位； 发送模式时，发送完一个数据字节后且发送数据寄存器为空时置位。 在收到无应答（NACK）条件时不会置位。 要清除该位，软件在读出该位为 1 后，再写或读数据寄存器；硬件收到开始或停止状态，或在 CR1_PE 为 0 时清除。
1	SR1_ADDR	R	地址阶段成功 主模式时发送地址成功，或从模式时接收地址命中。 收到 NACK 时不会置位。 要清除该位，软件在读出该位为 1 后，再读 I2C_SR2 寄存器；硬件在 CR1_PE 为 0 时清除。
0	SR1_SB	R	开始条件成功 主模式成功生成开始条件后置位。 要清除该位，软件在读出该位为 1 后，再写数据寄存器（地址）；硬件在 CR1_PE 为 0 时清除。

10.5.6 I²C 状态寄存器 2 (I2C_SR2)

偏移量： 0x18

复位值： 0x00000000

表 10-9 I²C 状态寄存器 2

位域	名称	访问	描述
31:5	Reserved	—	—
4	SR2_GENCALL	R	广播地址命中 当 CR1_ENGC 有效时，接收到了广播地址后置位。 硬件接收到开始或停止条件后清除，或在 CR1_PE 为 0 时清除。
3	Reserved	—	—
2	SR2_TRA	R	发送模式 0：接收模式；1：发送模式。 当内部状态机为发送相关的状态时，该位为 1，否则均为 0。 当 CR1_PE 为 0，或各种会导致状态机离开发送状态的事件均会清除该位。
1	SR2_BUSY	R	总线忙 硬件检测到 SCL 或 SDA 线上为低时置位，检测到停止条件后清除。 该位在 CR1_PE 为 0 时仍会更新。
0	SR2_MSL	R	主从模式 0：从模式；1：主模式。 默认为从模式，从模式下收到开始或恢复命令后变为主模式。 主模式接收到停止条件或仲裁丢失时回到从模式，在 CR1_PE 为 0 时为从模式。

10.5.7 I²C 时钟控制寄存器（I2C_CCR）

偏移量： 0x1c

复位值： 0x00000000

本寄存器不含硬件置位或复位。

表 10-10 I²C 时钟控制寄存器

位域	名称	访问	描述
31:16	Reserved	—	—
15	CCR_F_S	RW	主模式选择 0：标准模式；1：快速模式。
14	CCR_DUTY	RW	占空比控制 0：快速模式 SCL 高低电平时间为 1:2； 1：快速模式 SCL 高低电平时间为 9:16。
13:12	Reserved	—	—
11:0	CCR_CCR	RW	时钟分频控制 定义 SCL 的高电平时间 T _{high} 和低电平时间 T _{low} 。 标准模式时： T _{high} =CCR_CCR*T_PCLK T _{low} =CCR_CCR*T_PCLK 快速模式且 CCR_DUTY 为 0 时： T _{high} =CCR_CCR*T_PCLK T _{low} =2*CCR_CCR*T_PCLK 快速模式且 CCR_DUTY 为 1 时： T _{high} =9*CCR_CCR*T_PCLK T _{low} =16*CCR_CCR*T_PCLK 例如，要配置标准模式 100KHz 时钟，APB 时钟为 50MHz 时： CCR_CCR = 5000ns/20ns = 250

10.5.8 I²C 上升时间寄存器 (I2C_TRISE)

偏移量: 0x20

复位值: 0x00000000

本寄存器不含硬件置位或复位。

表 10-11 I²C 上升时间寄存器

位域	名称	访问	描述
31:6	Reserved	—	—
5:0	TRISE	RW	SCL 最大上升时间（主模式） 根据总线协议规定的最大 SCL 上升时间与 APB 时钟进行配置。例如，标准模式下最大上升事件为 1000ns，APB 时钟频率 50MHz，则 TRISE 配置为 $TRISE = 1000ns/20ns + 1 = 51$。当除法结果不是整数时，向下取整即可。 该位域用于在合适的时间判断是否出现时钟伸展，从而有助于保持 SCL 频率稳定。

11 打印接口

11.1 概述

打印接口模块集成在芯片打印系统下，并可共享给扫描系统与主系统。主要用于将内存中的打印数据搬运并转换成指定格式（串/并）输出至打印机芯，支持黑白与红黑（至多2通道）打印。其中，主要包含以下功能模块：

- (1) DMA 控制器，用于从内存中搬运打印数据；
- (2) 像素数据控制器，负责将像素数据转换为指定格式（1，2，4，8 比特色深）；
- (3) 调制单元，实现并行数据的串行解码与 sync 信号的处理；
- (4) 交换台单元，主要用于 sync 信号与输出数据的来源选择。

打印接口模块基地址为 0x1510c000。

11.2 配置寄存器

表 11-1 打印控制器地址空间分布

寄存器偏移地址	寄存器名称	寄存器大小
0x0000 - 0x03ff	打印控制器 0 (Channel 0)	1KB
0x0400 - 0x07ff	打印控制器 1 (Channel 1)	1KB
0x2000 - 0x20ff	打印接口配置	256B

12 扫描接口

12.1 概述

扫描接口模块集成在芯片扫描系统下，并可共享给打印系统与主系统。主要用于生成扫描仪中 CIS、LED 和 AFE 的控制时序，并将采回的数据存至指定内存区域。扫描接口模块支持灰度与彩色扫描，同时支持分段式（Segment）CIS 的数据拼接。支持的 AFE 数据输入宽度有 4、8 和 16 位，并可根据不同 AFE 特性对输入数据进行对齐。支持至多 12 个分段（单色 12 个，彩色 4 个），并可在 DMA 搬运时将不同颜色不同分段的数据进行重组，便于后续图像处理。其中，主要包含以下功能模块：

- (1) CIS 控制单元，用于生成 CIS 及 LED 所需时序；
- (2) AFE 控制单元，主要用于生成 AFE 控制时序并接收像素数据；
- (3) 数据拼接单元，将接收到的像素数据以颜色和分段进行重组并添加 tag；
- (4) DMA 控制器，根据 tag 将像素数据发送至指定内存区域；
- (5) 中断处理单元，负责根据各模块状态产生中断。

扫描接口模块基地址为 0x15206000。

12.2 配置寄存器

表 12-1 SCAN 控制器地址空间分布

寄存器偏移地址	寄存器名称	寄存器大小
0x0000 - 0x01ff	CIS 控制单元配置	512B
0x0200 - 0x02ff	AFE 控制单元配置	256B
0x0300 - 0x03ff	数据处理单元配置	256B
0x0400 - 0x04ff	扫描接口配置	128B

13 PMIO

13.1 概述

龙芯 2P0300 集成 3 组可编程多功能 IO（Programmable Multifunction IO，PMIO），其中含有多个功能模块，可通过 SoC 内部或外部输入产生输出。打印系统集成了一个双 Bank 全功能 PMIO，扫描系统集成了一个单 Bank 全功能 PMIO，主系统包含一个单 Bank 精简版 PMIO。

PMIO 的核心功能模块，可由软件或其他 PMIO 功能模块控制，并产生复杂的输出。其中包含：

核心功能模块：（精简版 PMIO 仅支持 PWM 功能）

1. PWM 输出：包含 Timer、Clock 和 PWM Generator，在触发后生成相应波形；
2. 输入捕捉：SIN、DIN，用于测量输入信号的速度或位置；
3. IO 功能：FIFO，在触发后将寄存器中的值串行输出或将输入记录至寄存器中；
4. 其他功能：Access（读写对应寄存器），Bypass（输入直连输出）。

触发控制：

1. PMIO 内部，其他 PMIO，或外部输入事件均可作为触发信号。
2. 触发信号的采样/屏蔽控制

一个全功能 PMIO Bank 含有 24 个功能模块（Function Unit），其中 16 个为可输出功能模块，8 个为内部功能模块；一个精简版 PMIO Bank 仅含有 16 个可输出的功能模块。控制器地址空间分布如下表所示。

13.2 配置寄存器

表 13-1 各系统 PMIO 控制器地址空间分布

各系统 PMIO 基地址	寄存器名称	寄存器大小
0x1420_6000	主系统 Bank0 寄存器	2KB
0x1420_6f00	主系统 PMIO 寄存器	2KB
0x1510_8000	打印系统 Bank0 寄存器	2KB
0x1510_8800	打印系统 Bank1 寄存器	2KB
0x1510_8f00	打印系统 PMIO 寄存器	2KB
0x1520_4000	扫描系统 Bank0 寄存器	2KB
0x1520_4f00	扫描系统 PMIO 寄存器	2KB

14 JPEG 编码器

14.1 概述

龙芯 2P0300 集成一个以 RGB 为输入，JPEG 为输出的 JPEG 编码器，该编码器直接生成 JPEG 文件格式的 SOI 到 EOI 部分字段。JPEG 编码器内部集成了 DMA 读写接口，作为输入的 RGB 原始图像需要按照 Zero[31:24], R[23:16], G[15:8], B[7:0] 的格式按 32 位对齐放入内存，供 Y 分量使用的 DC 量化表和供 Cb, Cr 分量使用的 AC 量化表各存放了 64 个量化值（Q），可以将 32768/Q 的值按照 32 位对齐放入内存。需要注意的是，其中只有[15:0]是有效的。编码器采用固定的哈夫曼编码值，支持垂直方向 4:4:4，4:2:2，4:1:1 和灰度四种采样形式。

为实现扫描模块与 JPEG 之间的数据转换，在 JPEG 编码器内部集成了一个以扫描得到的 R/G/B 行数据为输入，ARGB 为输出的数据格式转换模块，该模块将扫描模块得到的按照 R/G/B 行排列的数据转换为 ARGB 数据。

控制器的基址为 0x14050000。

14.2 寄存器描述

表 14-1 JPEG 编码器寄存器列表

偏移地址	名称	描述
0x00	CMD	命令寄存器
0x04	STS	状态寄存器
0x08	Q_TABLE_ADDR	量化表寄存器
0x0c	BLOCK_OF_IMAGE	块数量寄存器
0x10	IMAGE_SIZE	图像尺寸寄存器
0x14	RGB_IN_ADDR	输入地址寄存器
0x18	JPEG_OUT_ADDR	输出地址寄存器
0x1c	ENCODING_BYTE	编码字节数寄存器
0x20	TRAILING_BIT_CNT	末尾字有效位数寄存器
0x24	TRAILING_WORD	跟随字寄存器
0x100	IT_CMD	格式转换命令寄存器
0x104	IT_STS	格式转换状态寄存器
0x108	IT_SIZE	格式转换尺寸寄存器
0x10c	IT_IN_ADDR	格式转换输入地址寄存器
0x110	IT_OUT_ADDR	格式转换输出地址寄存器

14.2.1 命令寄存器（CMD）

偏 移：0x00

复位值：32' h0

表 14-2 命令寄存器

位域	名称	访问	描述
31	soft_reset	RW	软复位 高有效，复位后需要手动解除
30: 17	-	-	保留
16	finish_int_en	RW	结束中断使能 置 1 使能中断
15: 7	-	-	保留
6	axi_order	RW	AXI 读取数据顺序 1 : AXI 128 位数据视为按照 {pixel3, pixel2, pixel1, pixel0} 排列, pixel3 在最高位 0 : AXI 128 位数据视为按照 {pixel0, pixel1, pixel2, pixel3} 排列, pixel0 在最高位
5	clear_dc	RW	清除 DC 值 1: 清除 DC 差分值, 清除后不可进行接续编码 0: 解除清除状态
4	need_load_q_tbl	RW	需要装载量化表 1: 装载内存中的量化表 0: 默认使用全 1 量化表
3	bit_bound	RW	末尾按位边界生产 1: 编码末尾按位边界生成 0: 末尾编码在字节边界前自动补若干个 1
2: 1	sample_type	RW	采样形式 00: 4: 4: 4 01: 4: 2: 2 10: 4: 1: 1 11: 灰度
0	start	RW	开始编码 全部配置完成后, 置 1 开始编码

14.2.2 状态寄存器 (STS)

偏 移: 0x04

复位值: 32'h0

表 14-3 状态寄存器

位域	名称	访问	描述
31: 1	-	-	保留
0	jpeg_enc_done	RW1C	JPEG 编码完成 编码完成后此位变成 1, 写 1 清零, 清零后写 0 解除

14.2.3 量化表地址寄存器 (Q_TABLE_ADDR)

偏 移: 0x08

复位值: 32'h0

表 14-4 量化表地址寄存器

位域	名称	访问	描述
31: 0	addr	RW	地址 量化表在内存的地址, 按 16 字节对齐

14.2.4 块数量寄存器 (BLOCK_OF_IMAGE)

偏 移: 0x0c

复位值: 32' h0

表 14-5 块数量寄存器

位域	名称	访问	描述
31: 0	amount	RW	数量 原始 RGB 图像中包含 8*8 块的数量

14.2.5 图像尺寸寄存器 (IMAGE_SIZE)

偏 移: 0x10

复位值: 32' h0

表 14-6 寄存器

位域	名称	访问	描述
31:16	Reserved	RO	-
15: 0	image_width	RW	图片宽度 原始 RGB 图像宽度, 必须可以被 8 像素整除

14.2.6 输入地址寄存器 (RGB_IN_ADDR)

偏 移: 0x14

复位值: 32' h0

表 14-7 输入地址寄存器

位域	名称	访问	描述
31: 0	rgb_in_addr	RW	输入地址 原始 RGB 图像在内存的地址, 按 16 字节对齐

14.2.7 输出地址寄存器 (JPEG_OUT_ADDR)

偏 移: 0x18

复位值: 32' h0

表 14-8 输出地址寄存器

位域	名称	访问	描述
31: 0	jpeg_out_addr	RW	输出地址 输出 jpeg 编码在内存的地址, 按 4 字节对齐 编码完成后硬件自动配置为本次编码输出地址末尾用于连续编码, 若无需连续编码可软件清除

14.2.8 编码字节数寄存器 (ENCODING_BYTE)

偏 移: 0x20

复位值: 32' h0

表 14-9 编码字节数寄存器

位域	名称	访问	描述
31: 0	encoding_byte	RW	输出 JPEG 编码的字节数 注：若有接续编码，接续编码所在 4 字节也会被计算在内

14.2.9 末尾字有效位数寄存器 (TRAILING_BIT_CNT)

偏 移：0x20

复位值：32' h0

表 14-10 末尾字节位数寄存器

位域	名称	访问	描述
31: 5	-	-	保留
4: 0	bit_cnt	RW	编码末尾字包含有效位个数，用于连续编码，编码完成后硬件自动配置，若无需连续编码可软件清除

14.2.10 跟随字寄存器 (TRAILING_WORD)

偏 移：0x2c

复位值：32' h0

表 14-11 跟随字节寄存器

位域	名称	访问	描述
31: 0	trailing_word	RW	跟随字 上次编码的末尾字，用于连续编码，编码完成后硬件自动配置，若无需连续编码可软件清除 如需软件配置，请将输出数据按 32 位对齐的最后一个字（不满一个字时末尾补 0）按照 {word[7:0], word[15:8], word[23:16], word[31:24]} 的顺序重排写入，若数据包含 0xff，编码时会自动补齐 0x00，请勿将 0x00 重复计算

14.2.11 格式转换命令寄存器 (IT_CMD)

偏 移：0x100

复位值：32' h0

表 14-12 格式转换命令寄存器

位域	名称	访问	描述
31	soft_reset	RW	软复位 高有效，复位后需软件清 0
30:17	Reserved	RO	-
16	finish_int_en	RW	结束中断使能 置 1 使能中断
15:2	Reserved	RO	-
1	data_type	RW	数据类型 0：原始数据为 R/G/B LINE 数据 1：原始数据为 Y LINE 数据
0	start	RW	配置完成后，置 1 开始转换

14.2.12 格式转换状态寄存器 (IT_STS)

偏 移: 0x104

复位值: 32' h0

表 14-13 格式转换命令寄存器

位域	名称	访问	描述
31: 1	Reserved	RO	–
0	trans_done	RW1C	数据转换完成 转换完成后置 1, 写 1 清零, 清零后写 0 解除

14.2.13 格式转换尺寸寄存器 (IT_SIZE)

偏 移: 0x108

复位值: 32' h0

表 14-14 格式转换尺寸寄存器

位域	名称	访问	描述
31: 16	image_height	RW	图片高度 原始图像高度
15:0	image_width	RW	图片宽度 原始图像宽度, 必须可以被 8 像素整除

14.2.14 格式转换输入地址寄存器 (IT_IN_ADDR)

偏 移: 0x10c

复位值: 32' h0

表 14-15 格式转换输入地址寄存器

位域	名称	访问	描述
31: 0	in_addr	RW	输入地址 原始图像在内存地址, 32 字节对齐

14.2.15 格式转换命令寄存器 (IT_CMD)

偏 移: 0x110

复位值: 32' h0

表 14-16 格式转换输出地址寄存器

位域	名称	访问	描述
31: 0	out_addr	RW	输出地址 输出数据在内存地址, 4 字节对齐

14.3 编程示例

JPEG 编码配置操作流程:

- (1) 命令寄存器配置采用格式: CMD = 0x10; //装载量化表, 按照 4: 4: 4 采样
- (2) 配置量化表地址: Q_TABLE_ADDR = 0x0; //量化表在内存地址为 0x0

- (3) 配置图像块大小: `BLOCK_OF_IMAGE = 0x1000;` //原始 RGB 图像包含 4096 个 8*8 块
- (4) 配置原图像宽度: `IMAGE_SIZE = 0x800;` //原始 RGB 图像宽度为 2048
- (5) 配置原图像大小: `RGB_IN_ADDR = 0x200;` //原始 RGB 图像在内存地址为 0x200
- (6) 配置 JPEG 格式文件存放地址: `JPEG_OUT_ADDR = 0x40000;` //输出 JPEG 编码地址为 0x40000
- (7) 启动 JPEG 编码: `CMD = CMD | 0x1;` //开始编码
- (8) 等待 JPEG 编码结束标志: `while(~(STS & 0x1));` //等待编码结束状态位
- (9) 读取 JPEG 格式数据

15 JBIG85 解码器

15.1 概述

JBIG85 解码模块基于 ITU-T T.85 协议规范，负责对二值图像进行对应数据的解码工作。JBIG 模块包括主要包括两个子模块：主控制模块、JBIG 算法解码模块。各 JBIG 模块共用同一个系统中断，可通过配置实现中断的与模式/或模式。结构框图如图 15-1 所示：

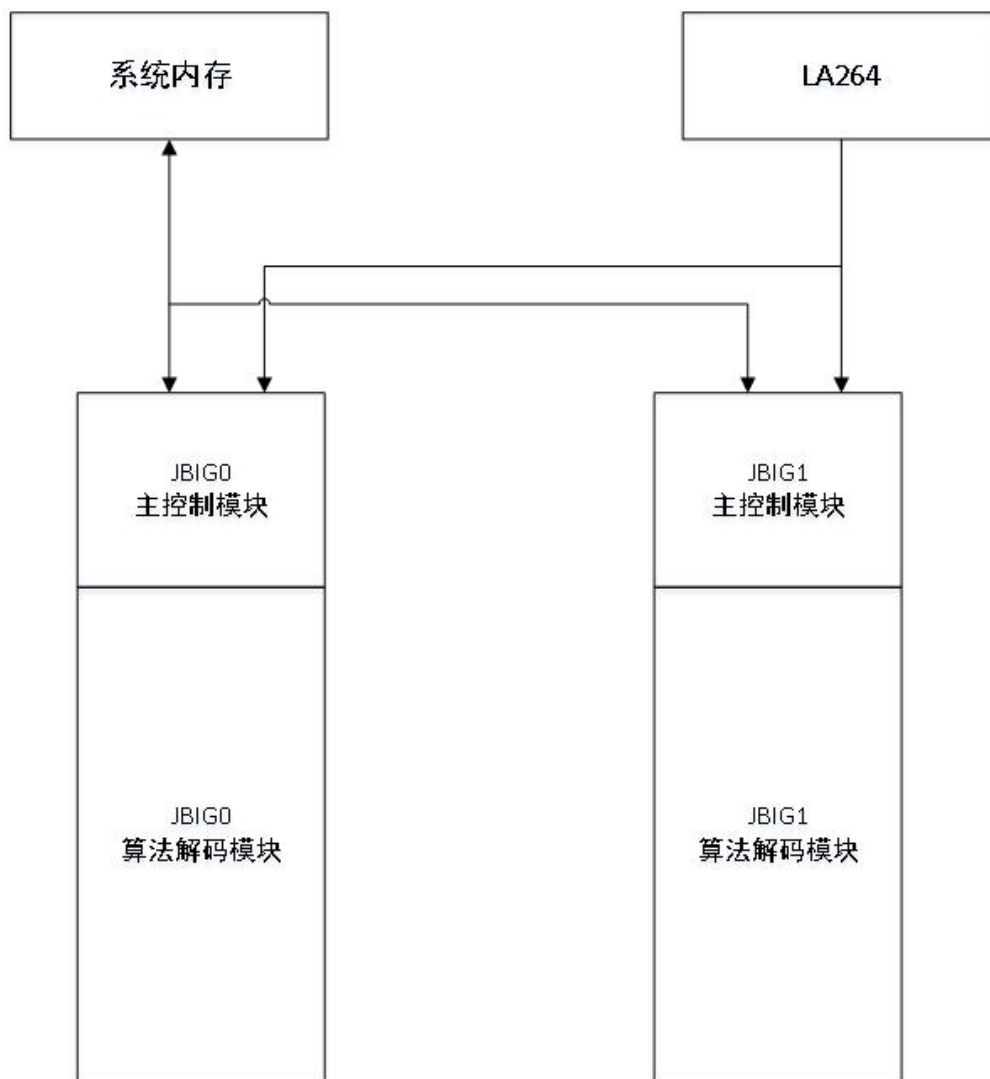


图15-1 JBIG85 解码结构示意图

JBIG85 主控制模块包括三个功能：寄存器配置、输入数据读取，输出数据写回。可配置寄存器通过 APB 协议进行寄存器读写，寄存器项包括模块使能寄存器、输入数据起始地址、输入数据长度、输出数据起始地址、多模块的中断传递类型、解码错误状态寄存器。输入数据保存在内存中，数据地址由 APB 进行配置，主控制模块通过 AXI 协议对数据进行读取，并将读取的数据发送至 JBIG 解码模块进行解码。输出数据通过 AXI 协议进行写回，写回基地址同样由 APB 进行配置，并且所有写回数据按照图片行数据进行 256 像素单位对

齐。

JBIG85 算法解码模块主要实现对经由 jbig85 压缩算法压缩过的图片数据。

控制器的基址为 0x14040000。

15.2 寄存器描述

JBIG85 模块有 2 个相同的 JBIG 解码单元，两个模块均具有相同的配置寄存器格式。

表 15-1 JBIG 配置地址

偏移地址	设备	备注
0x0000	JBIG0	两个模块寄存器配置方式相同， 区别在于寄存器基地址偏移不同
0x4000	JBIG1	

JBIG85 每个模块包括 12 个可以通过配置总线进行读写的寄存器。

表 15-2 JBIG 配置寄存器

偏移地址	位宽	寄存器名称	读写	说明
0x0	32	jbig_top_cfg	R/W	JBIG 模块的整体配置位，
0x4	32	pscd_addr	R/W	当前 JBIG 模块待解码原始数据的存储地址
0x8	32	pscd_data_length	R/W	当前 JBIG 模块待解码数据的长度，单位为字节
0xc	32	output_addr	R/W	当前 JBIG 模块解码后数据存储地址
0x10	2	start	R/W	开始当前 JBIG 模块的解码，高电平有效
0x14	1	error_processed_reg	R/W	错误清除位，出现错误并处理结束后，写 1 清除
0x18	32	error_code_reg	R	错误码状态寄存器
0x1c	1	apb_clear	W	清空当前 JBIG 模块的所有配置项，高电平有效
0x20	3	int_reg_clear	W	清除当前 JBIG 模块的中断位，高电平有效
0x24	3	int_reg	R	读取当前 JBIG 模块的中断位
0x30	32	state_for_int	R/W	基于内部状态产生中断
0x34	32	line_for_int	R/W	基于已解码行数产生中断

(1) jbig_top_cfg 寄存器

偏移地址：0x0。

表 15-3 jbig_top_cfg 寄存器

位域	名称	复位值	访问	说明
31:13	Reserved	-	RO	保留
12	line_stop_en	1' b0	R/W	使能行数暂停 注：必须使能 line_int_en 才可使用
11	line_int_en	1' b0	R/W	使能行数中断
10	state_int_en	1' b0	R/W	使能内部状态机状态中断
9	page_end_int_en	1' b0	R/W	页完成中断使能

位域	名称	复位值	访问	说明
8	int_oen	1' b0	R/W	2 个 JBIG 模块的中断位合并为 1 位输出， 0: 将 2 位中断位做 AND 操作，所有模块均为中断，才输出中断，可用作没有数据错误的正常打印方式。 1: 将 2 位中断位做 OR 操作，所有模块有一个中断，则立即输出中断，可用作有数据错误的调试方式。
7:6	Reserved	—	RO	保留
5:4	int_en	2' b0	R/W	配置 JBIG 模块的中断使能位 1: 打开使能 0: 关闭使能 [5]表示 JBIG1（对应 JBIG1 偏移地址寄存器位域[5]） [4]表示 JBIG0（对应 JBIG0 偏移地址寄存器位域[4]）
3:2	Reserved	—	RO	保留
1:0	jbig_en	2' b0	R/W	配置 JBIG 模块的使能位 1: 打开使能 0: 关闭使能 [1]表示 JBIG1（对应 JBIG1 偏移地址寄存器位域[1]） [0]表示 JBIG0（对应 JBIG0 偏移地址寄存器位域[0]）

(2) start 寄存器

偏移地址：0x10

表 15-4 start 寄存器

位域	名称	复位值	访问	说明
1	continue	1' b0	R_W	用于 line_stop 暂停输出后继续解码
0	start	1' b0	R/W	开始解码

(3) error_code_reg 寄存器

偏移地址：0x18

JBIG 错误码，数据 s->buffer 为 jbig85 数据前 20 字节数据，为当前 jbig85 数据的描述符，x0/y0/10/mx/options 为通过 buffer 计算得来的数据。

表 15-5 error_code_reg 寄存器

位域	名称	复位值	访问	说明
31:16	Reserved	—	—	
15:8	jbig1_err_code	8' b0	R/W	JBIG1 的错误码，位域含义等同 [7:0]（对应 JBIG1 偏移地址寄存器位域 [15:8]）

位域	名称	复位值	访问	说明
7:0	jbig0_err_code	8' b0	R/W	JBIG0 的错误码（对应 JBIG0 偏移地址寄存器位域 [7:0]），数据 s 为 jbig85 数据的描述符，具体表示参考 jbig85_c_code [7:4] == 4' d6, 输入数据包含错误输入 [3:0] == 4' d1: s->buffer[1] < s->buffer[0] [3:0] == 4' d2: s->buffer[3] != 0 [3:0] == 4' d3: s->buffer[18]&0xf0 != 0 [3:0] == 4' d4: s->buffer[19]&80 != 0 [3:0] == 4' d5: s->buffer[2] == 0 [3:0] == 4' d6: s->x0 == 0 [3:0] == 4' d7: s->y0 == 0 [3:0] == 4' d8: s->l0 == 0 [3:0] == 4' d9: s->mx > 127 [7:4] == 4' d7, 输入数据不符合 JBIG85 格式 [3:0] == 4' d8: s->buffer[0] != 0 [3:0] == 4' d9: s->buffer[1] != 0 [3:0] == 4' d10: s->buffer[2] != 1 [3:0] == 4' d11: s->buffer[17] != 0 [3:0] == 4' d12: s->buffer[18] != 0 [3:0] == 4' d13: s->options &0x17 != 0

(4) int_reg_clear 寄存器

偏移地址：0x20

表 15-6 int_reg_clear 寄存器

位域	名称	复位值	访问	说明
2	line_int_clear	1' b0	W	清除 line_int 中断状态
1	state_int_clear	1' b0	W	清除 state_int 中断状态
0	int_reg_clear	1' b0	W	清除 int_reg 中断状态

(5) int_reg 寄存器

偏移地址：0x24

表 15-7 int_reg 寄存器

位域	名称	复位值	访问	说明
2	line_int	1' b0	R	行中断标志
1	state_int	1' b0	R	状态中断标志
0	int_reg	1' b0	R	解码完成标志

(6) state_for_int 寄存器

偏移地址：0x30

表 15-8 state_for_int 寄存器

位域	名称	复位值	访问	说明
31:16	state1	16' b0	R/W	对应 0x2c 寄存器状态
15:0	state0	16' b0	R/W	对应 0x28 寄存器状态

(7) line_for_int 寄存器

偏移地址：0x34

表 15-9 line_for_int 寄存器

位域	名称	复位值	访问	说明
31:16	reserved	-	-	
15:0	line_for_int	16' b0	R/W	行中断数

15.3 编程说明

操作 1：正常打印配置方式

1. 逐个配置 2 个 JBIG 模块的 apb_clear 寄存器，清空所有 jbig 配置项；
2. 逐个配置 2 个 JBIG 模块的 jbig_cfg 寄存器；
3. 配置需要启用的 JBIG 模块的 pscd_addr 寄存器，将待解压缩的 jbig 数据地址写入

该寄存器；

4. 配置需要启用的 JBIG 模块的 pscd_length 寄存器，将待解压的 jbig 数据长度写入该寄存器；

5. 配置需要启用的 JBIG 模块的 output_addr 寄存器，将解压后的数据地址写入该寄存器；

6. 配置需要启用的 JBIG 模块的 start 寄存器，开始解压缩。

操作 2：打印结束后获取状态

1. 读取 int_reg 寄存器，如果值为 1，表示打印结束或者出现错误；
2. 读取 error_code_reg 寄存器，如果值为 0，表示没有错误，为正常结束；
3. 配置 int_reg_clear 寄存器；
4. 重复正常打印配置方式，继续下一页打印。

操作 3：打印错误及状态获取

1. 读取 int_reg 寄存器，如果值为 1，表示打印结束或者出现错误；

2. 读取 error_code_reg 寄存器，如果值不为 0，表示有错误，对照错误码表，确定错误类型；
3. 配置 int_reg_clear 寄存器；
4. 修改错误，重复正常打印配置方式，继续下一页打印。

16 UART 控制器

16.1 概述

龙芯 2P0300 集成了 5 个 UART 控制器，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。

16.2 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。

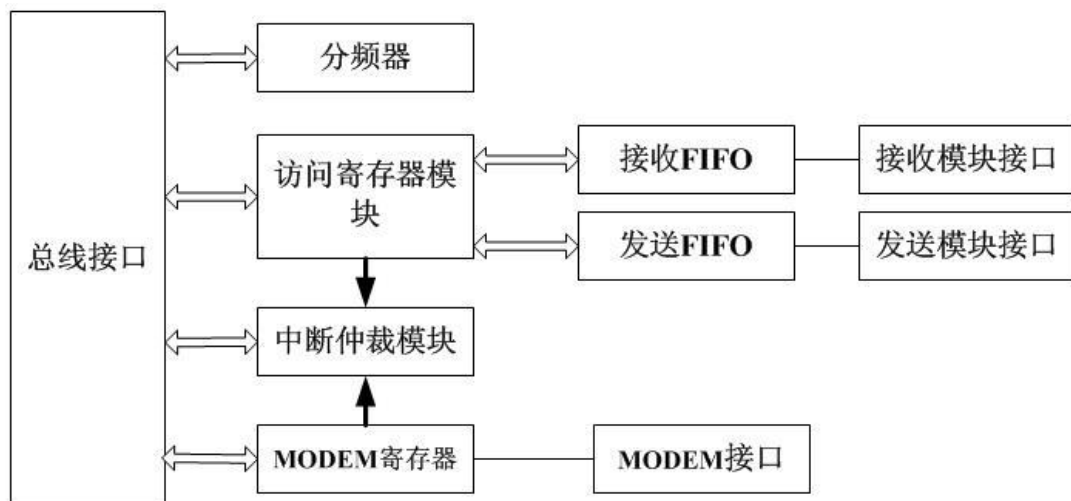


图16-1 UART 控制器结构

主要模块功能及特征描述如下：

1) 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。

2) 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。

3) 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。

4) 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。

16.3 寄存器描述

龙芯 2P0300 中有 5 个 UART，每个 UART 只是访问基址不一样，其功能寄存器保持一致。

MSYS_UART0 寄存器物理地址基址为 0x14200000。

MSYS_UART1 寄存器物理地址基址为 0x14200800。

PRT_UART0 寄存器物理地址基址为 0x15100000。

PRT_UART1 寄存器物理地址基址为 0x15100800。

SCA_UART0 寄存器物理地址基址为 0x15200000。

UART 接口模块配置寄存器，如下表所示。

表 16-1 UART 配置寄存器列表

地址偏移	寄存器名称	描述
0x00	DAT/DL_L	数据寄存器/分频系数低 8 位
0x01	IER/DL_H	中断使能寄存器/分频系数高 8 位
0x02	IIR	中断标识寄存器
0x02	FCR/DL_D	FIFO 控制寄存器/分频系统小数位
0x03	LCR	线路控制寄存器
0x04	MCR/SPC	Modem 控制寄存器/采样点配置寄存器
0x05	LSR	线路状态寄存器
0x06	MSR	Modem 状态寄存器

对于 UART 接口模块，使用时要注意将对应的芯片复用引脚设置为相应的接口功能。

与 UART 接口相关的引脚复用设置可查询外设功能引脚复用关系表，配置相应的 GPIO 引脚复用关系，实现对应设备功能引脚。

16.3.1 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

表 16-2 数据传输寄存器

位域	位域名称	位宽	访问	描述
7:0	DATA	8	W	数据传输寄存器

16.3.2 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

表 16-3 中断使能寄存器

位域	位域名称	位宽	访问	描述
7	ACTSE	1	RW	CTS 自动流控使能 ‘0’ - 关闭 ‘1’ - 打开 打开时，若 CTSn 的输入为 0，则暂停发送
6	ARTSE	1	RW	RTS 自动流控使能 ‘0’ - 关闭 ‘1’ - 打开 打开时，若 rxfifo 为满，则 RTSn 的输出为 0
5	TXDE	1	RW	发送状态 DMA 使能 ‘0’ - 关闭 ‘1’ - 打开 打开时，若 txfifo 未满，向 dma 发送发送请求
4	RXDE	1	RW	接收状态 DMA 使能 ‘0’ - 关闭 ‘1’ - 打开 打开时，若 rxfifo 未满，向 dma 发送接收请求
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

16.3.3 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

表 16-4 中断源寄存器

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（低有效）

表 16-5 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 st	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 nd	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 nd	接收超时	在 FIFO 至少有一个字符，但在 4 个字符时	读接收 FIFO

Bit 3	Bit 2	Bit 1	优先级	中断类型	中断源	中断复位控制
					间内没有任何操作，包括读和写操作	
0	0	1	3 rd	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 th	Modem 状态	CTS, DSR, RI or DCD.	读 MSR

16.3.4 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

表 16-6 FIFO 控制寄存器

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

16.3.5 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

表 16-7 线路控制寄存器

位域	位域名称	位宽	访问	描述
7	dlab	1	RW	分频锁存器/采样点寄存器访问位 '1' - 访问操作分频锁存器/采样点寄存器 '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

位域	位域名称	位宽	访问	描述
				‘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

16.3.6 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

表 16-8 Modem 控制寄存器

位域	位域名称	位宽	访问	描述
7: 5	Reserved	3	RW	保留
4	Loop	1	RW	回环模式控制位 ‘0’ - 正常操作 ‘1’ - 回环模式。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	RW	在回环模式中连到 DCD 输入
2	OUT1	1	RW	在回环模式中连到 RI 输入
1	RTSC	1	RW	RTS 信号控制位
0	DTRC	1	RW	DTR 信号控制位

16.3.7 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

表 16-9 线路状态寄存器

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

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

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

16.3.8 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

表 16-10 Modem 状态寄存器

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

16.3.9 分频锁存器

中文名： 分频锁存器低 8 位

寄存器位宽： [7: 0]

偏移量： 0x00

复位值： 0x00

表 16-11 分频锁存器低 8 位寄存器

位域	位域名称	位宽	访问	描述
7:0	DL_L	8	RW	存放分频锁存器的低 8 位数值

中文名： 分频锁存器高 8 位

寄存器位宽： [7: 0]

偏移量： 0x01

复位值： 0x00

表 16-12 分频锁存器高 8 位寄存器

位域	位域名称	位宽	访问	描述
7:0	DL_H	8	RW	存放分频锁存器的高 8 位数值

中文名： 分频锁存器小数位

寄存器位宽： [7: 0]

偏移量： 0x02

复位值： 0x00

表 16-13 分频锁存器小数位寄存器

位域	位域名称	位宽	访问	描述
7:0	DL_D	8	RW	存放分频锁存器的小数部分二进制值 例如，分频小数值为 0.45，DL_D=0x73(即： 0x100*0.45)

UART 分频由 DL_H、DL_L、DL_D 三个寄存器组成分频值寄存器 DL，则 UART 波特率为：
 $CLK_{in}/16*DL$ 。例如，输入为 10MHz 时钟，目标波特率为 115200，则 $DL=5.42535$ ，故，
 DL_H=0x0，DL_L=0x5，DL_D=0x6c(即：0x100*0.42535)。波特率配置值与期望值的误差在
 5%以内，否则无法识别所有数据位，将导致串口显示乱码。

16.3.10 采样点寄存器 (SPC)

中文名： 采样点寄存器

寄存器位宽： [7: 0]

偏移量: 0x04
复位值: 0x77

表 16-14 采样点寄存器

位域	位域名称	位宽	访问	描述
7	Reserved	1	RW	保留
6:4	ws1	3	RW	配置采样窗口内采样点后的周期数，对应内部时钟周期为 ws1+1
3	Reserved	1	RW	保留
2: 0	ws0	3	RW	配置采样窗口内采样点前的周期数，对应内部时钟周期为ws1+0

17 eMMC 控制器

17.1 功能概述

龙芯 2P0300 集成了两个 SDIO/eMMC 控制器。eMMC 控制器特性如下：

- 支持 eMMC 启动
- 8 位预分频逻辑（频率=系统时钟/(p+1)）
- DMA 数据传输模式
- 专用独立 DMA 通道
- 1 位/4 位/8 位的总线模式

17.2 访问地址及引脚复用

SDIO/eMMC 控制器内部寄存器的物理地址构成如下：

表 17-1 eMMC 内部寄存器物理地址构成

地址	设备	备注
0x1421_0000	SDIO0/eMMC0	32KB 大小寄存器配置空间
0x1421_8000	SDIO1/eMMC1	32KB 大小寄存器配置空间

对于 eMMC 模块，使用时要注意将对应的引脚设置为相应的功能。

与 eMMC 相关的引脚复用设置可查询节 中的 eMMC 功能引脚复用关系，并配置相应 GPIO 引脚复用配置寄存器实现。

17.3 寄存器描述

eMMC 控制器的寄存器详细说明如下：

表 17-2 EMMC_CON 寄存器

寄存器名称	偏移地址	读/写 (R/W)	功能描述	复位值
EMMC_CON	0x00	R/W	eMMC 控制寄存器	0x0

表 17-3 EMMC_CON 寄存器位域描述

EMMC_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	7:1	0x0	-
enclk	0	0x0	SD 时钟输出使能

表 17-4 EMMC_PRE 寄存器

寄存器名称	地址	读/写 (R/W)	功能描述	复位值
EMMC_PRE	0x04	R/W	eMMC 预分频寄存器	0x1

表 17-5 EMMC_PRE 寄存器位域描述

EMMC_PRE	位	缺省值	描述
emmc_clk_rev_en	31	0x1	SDR 模式时，该位为 1，表示控制器输出的 eMMC 数据与 eMMC 时钟下降沿对齐；该位为 0，表示控制器输出的 eMMC 数据与 eMMC 时钟上升沿对齐。DDR 模式，该位必须置为 1。
Reserved	30:8	0x0	-
emmc_pre	7:0	0x1	eMMC 时钟预分频值，输出频率=PCLK/预分频值

表 17-6 EMMC_CMD_ARG 寄存器

寄存器名称	偏移地址	读/写 (R/W)	功能描述	复位值
EMMC_CMD_ARG	0x08	R/W	eMMC 命令参数寄存器	0x0

表 17-7 EMMC_CMD_ARG 寄存器位域描述

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

表 17-8 EMMC_CMD_CON 寄存器

寄存器名称	偏移地址	读/写 (R/W)	功能描述	复位值
EMMC_CMD_CON	0x0c	R/W	eMMC 命令控制寄存器	0x0

表 17-9 EMMC_CMD_CON 寄存器位域描述

EMMC_CMD_CON	位	缺省值	描述
Reserved	31:18	0x0	
func_num_abort	17:15	0x0	eMMC 卡时中断的功能号，用于多块读写时，硬件自动发送停止命令。如果 auto_stop_en 为 0，则此位无效
emmc_en	14	0x0	eMMC 使能信号。用于多块读写时，硬件自动发送停止命令，为 1 时发送 CMD52，为 0 是发送 CMD12。如果 auto_stop_en 为 0，则此位无效
check_on	13	0x0	是否检查 CRC，为 1 时有效
Auto_stop_en	12	0x0	硬件自动发送停止命令，多块读写时，是否硬件自动发送停止命令，为 1 时有效
Reserved	11	0x0	-
long_rsp	10	0x0	是否为 136 位长响应，为 1 时表示长消息回复

EMMC_CMD_CON	位	缺省值	描述
Wait_rsp	9	0x0	决定是否主机等待响应，为 1 是表示等待消息回复
CMST	8	0x0	命令开始，置 1 时开始，命令结束后硬件自动清零
cmd_index	7:0	0x0	带开始 2 位的命令索引（共 8 位）

表 17-10 EMMC_CMD_STA 寄存器

寄存器名称	偏移地址	读/写 (R/W)	功能描述	复位值
EMMC_CMD_STA	0x10	RO	eMMC 命令状态寄存器	0x0

表 17-11 EMMC_CMD_STA 寄存器位域描述

EMMC_CMD_STA	位	缺省值	描述
Reserved	31:15	0x0	–
cmd_sent_fin	14	0x0	命令发送完成（包含响应）标志位，为 1 表示命令发送完成及响应完成
auto_stop	13	0x0	硬件自动发送停止命令标志位，为 1 表示硬件自动发送停止命令，为 0 则没有
rsp_crc_err	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_end	11	0x0	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），或者 R1b 类型的命令，忙等待超时，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）

表 17-12 EMMC_RSP0 寄存器

寄存器名称	偏移地址	读/写 (R/W)	功能描述	复位值
EMMC_RSP0	0x14	RO	eMMC 命令响应寄存器 0	0x0

表 17-13 EMMC_RSP0 寄存器位域描述

EMMC_RSP0	位	缺省值	描述
sdi_resp0	31:0	0x0	卡状态[31:0]（短），卡状态[127:96]（长）长响应的配置间 sdi_cmd_con[10]

表 17-14 EMMC_RSP1 寄存器

寄存器名称	偏移地址	读/写 (R/W)	功能描述	复位值
EMMC_RSP1	0x18	RO	eMMC 命令响应寄存器 1	0x0

表 17-15 EMMC_RESP1 寄存器位域描述

EMMC_RESP1	位	缺省值	描述
sdi_resp1	31:0	0x0	未使用（短），卡状态[95:64]（长）长响应的配置间 sdi_cmd_con[10]

表 17-16 EMMC_RSP2 寄存器

寄存器名称	偏移地址	读/写 (R/W)	功能描述	复位值
EMMC_RSP2	0x1c	RO	eMMC 命令响应寄存器 2	0x0

表 17-17 EMMC_RSP2 寄存器位域描述

EMMC_RESP2	位	缺省值	描述
sdi_resp2	31:0	0x0	未使用（短），卡状态[63:32]（长）长响应的配置间 sdi_cmd_con[10]

表 17-18 EMMC_RSP3 寄存器

寄存器名称	偏移地址	读/写 (R/W)	功能描述	复位值
EMMC_RSP3	0x20	RO	eMMC 命令响应寄存器 3	0x0

表 17-19 EMMC_RSP3 寄存器位域描述

EMMC_RESP3	位	缺省值	描述
sdi_resp3	31:0	0x0	未使用（短），卡状态[31:0]（长）长响应的配置间 sdi_cmd_con[10]

表 17-20 EMMC_DTIMER 寄存器

寄存器名称	偏移地址	读/写 (R/W)	功能描述	复位值
EMMC_DTIMER	0x24	R/W	eMMC 命令数据超时寄存器	0x0

表 17-21 EMMC_DTIMER 寄存器位域描述

EMMC_DTIMER	位	缺省值	描述
Reserved	31:24	0x0	
sdi_dtimer	23:0	0x0	数据超时计数值，用分频后的时钟计数

表 17-22 EMMC_BSIZE 寄存器

寄存器名称	偏移地址	读/写 (R/W)	功能描述	复位值
EMMC_BSIZE	0x28	R/W	eMMC 块大小寄存器	0x0

表 17-23 EMMC_BSIZE 寄存器位域描述

EMMC_BSIZE	位	缺省值	描述
Reserved	31:12	0x0	
sdi_bsize	11:0	0x0	块大小值 (0~4095)

表 17-24 EMMC_DAT_CON 寄存器

寄存器名称	偏移地址	读/写 (R/W)	功能描述	复位值
EMMC_DAT_CON	0x2c	R/W	eMMC 数据控制寄存器	0x0

表 17-25 EMMC_DAT_CON 寄存器位域描述

EMMC_DAT_CON	位	缺省值	描述
Reserved	31:21	0x0	-
wide_mode_8b	26	0x0	wide_mode_8b 为 1, wide_mode 为 0, 表示八线模式 (eMMC 模式有效)
resume_rw	20	0x0	eMMC 挂起回复读写标志位。为 1 时, eMMC 挂起后恢复之前的写操作; 为 0 时, 恢复之前的读操作
IO_resume	19	0x0	eMMC 恢复请求。在 eMMC 设备进入挂起状态后, 将此位写 1, 并且 IO_suspend 位写 0 后, eMMC 设备恢复之前的操作。
IO_suspend	18	0x0	eMMC 挂起请求。写 1 后控制器会在合适的时机发送 CMD52 命令, 通知 eMMC 设备进入挂起状态。恢复操作时需要将此位写 0。
RwaitReq	17	0x0	读等待请求。写 1 后控制器会在合适的时机将 DAT2 拉低, 通知 eMMC 设备进入读等待状态。写 0 后恢复之前的读操作。
wide_mode	16	0x0	位宽选择位。为 1 表示 4 线模式, 为 0 表示单线模式。
DMA_en	15	0x0	DMA 使能。为 1 时表示使能 DMA, 为 0 表示禁止 DMA
DTST	14	0x0	数据传输开始, 写 1 时数据传输开始, 数据传输结束后硬件清零。
Reserved	13:12	0x0	-
Blk_num	11:0	0x0	读写操作的块数。

表 17-26 EMMC_DAT_CNT 寄存器

寄存器名称	偏移地址	读/写 (R/W)	功能描述	复位值
EMMC_DAT_CNT	0x30	R/W	eMMC 数据计数寄存器	0x0

表 17-27 EMMC_DAT_CNT 寄存器位域描述

EMMC_DAT_CNT	位	缺省值	描述
Reserved	31:24	0x0	
blk_num_cnt	23:12	0x0	当前传输块的字节数
blk_cnt	11:0	0x0	当前传输的块数

表 17-28 EMMC_DAT_STA 寄存器

寄存器名称	偏移地址	读/写 (R/W)	功能描述	复位值
EMMC_DAT_STA	0x34	R0	eMMC 数据状态寄存器	0x0

表 17-29 EMMC_DAT_STA 寄存器位域描述

EMMC_DAT_STA	位	缺省值	描述
Reserved	31:17	0x0	-
suspend_on	16	0x0	为 1 时表示正在挂起状态
rst_suspend	15	0x0	为 1 表示正在挂起复位。用于 eMMC 设备挂起后，控制器复位 FIFO 和 DMA 请求
R1b_tout	14	0x0	为 1 表示 R1b 类型命令超时
data_start	13	0x0	为 1 表示数据传输开始
R1b_fin	12	0x0	检测到带 busy 状态的命令完成。当发送带 busy 状态的命令时，此位为 0；当 busy 状态结束时变成 1
auto_stop	11	0x0	为 1 时表示硬件正在自动发送停止命令
Reserved	10	0x0	-
r_wait_req	9	0x0	读等待发生。发送读等待请求信号到 eMMC 卡
emmc_int	8	0x0	eMMC 中断标志位。为 1 表示检测到中断
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（比如编程时）。为 1 时标志忙结束
busy_fin	3	0x0	编程错误标志位（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。

表 17-30 EMMC_FIFO_STA 寄存器

寄存器名称	偏移地址	读/写 (R/W)	功能描述	复位值
EMMC_FIFO_STA	0x38	R0	eMMC FIFO 状态寄存器	0x0

表 17-31 EMMC_FIFO_STA 寄存器位域描述

EMMC_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
rx_empty	7	0x0	Rx FIFO 空标志位
Reserved	6:0	0x0	-

表 17-32 EMMC_INT_MASK 寄存器

寄存器名称	偏移地址	读/写 (R/W)	功能描述	复位值
EMMC_INT_MASK	0x3c	R/W	eMMC 中断寄存器	0x0

表 17-33 EMMC_INT_MASK 寄存器位域描述

EMMC_INT_MASK	位	缺省值	描述
Reserved	31:10	0x0	-
Rlb_fin_int	9	0x0	检测到 busy 结束中断，写 1 清零
rsp_crc_int	8	0x0	命令响应 CRC 错误中断，写 1 清零
cmd_tout_int	7	0x0	命令超时中断，写 1 清零
cmd_fin_int	6	0x0	发送完成中断，硬件清零
emmc_int	5	0x0	检测到 eMMC 中断，写 1 清零
prog_err_int	4	0x0	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

表 17-34 EMMC_DAT 寄存器

寄存器名称	偏移地址	读/写 (R/W)	功能描述	复位值
EMMC_DAT	0x40	RO	eMMC 命令数据寄存器	0x0

表 17-35 EMMC_DAT 寄存器位域描述

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

表 17-36 EMMC_INT_EN 寄存器

寄存器名称	偏移地址	读/写 (R/W)	功能描述	复位值
EMMC_INT_EN	0x64	R/W	eMMC 中断寄存器使能寄存器	0x0

表 17-37 EMMC_INT_EN 寄存器位域描述

EMMC_INT_EN	位	缺省值	描述
Reserved	31:10	0x0	
Rlb_fin_int_en	9	0x0	Busy 结束中断使能，为 1 时有效
rsp_crc_int_en	8	0x0	命令响应 CRC 错误中断使能，为 1 时有效

EMMC_INT_EN	位	缺省值	描述
cmd_tout_int_en	7	0x0	命令超时中断使能，为 1 时有效
cmd_fin_int_en	6	0x0	命令发送完成中断使能，为 1 时有效
emmc_int_en	5	0x0	eMMC 中断使能，为 1 时有效
prog_err_int_en	4	0x0	SD 卡编程错误中断使能，为 1 时有效
crc_sta_int_en	3	0x0	数据发送后从设备返回 CRC 错误中断使能，为 1 时有效
dat_cec_int_en	2	0x0	数据接收 CRC 错误中断使能，为 1 时有效
dat_tout_int_en	1	0x0	数据超时中断使能，为 1 时有效
dat_fin_int_en	0	0x0	数据完成中断使能，为 1 时有效

表 17-38 DLL_MASTER_VAL 寄存器

寄存器名称	地址	读/写 (R/W)	功能描述	复位值
DLL_MASTER_VAL	0xf0	R	DLL master 锁定值	0x0

表 17-39 DLL_MASTER_VAL 寄存器位域描述

DLL_MASTER_VAL	位	缺省值	描述
reserved	31:9	0x0	—
dll_init_done	8	0x0	DLL master 锁定完成标志位
pm_dll_value	7: 0	0x0	DLL master 锁定值

表 17-40 DLL_CON 寄存器

寄存器名称	地址	读/写 (R/W)	功能描述	复位值
DLL_CON	0xf4	R/W	DLL 控制寄存器	0x0

表 17-41 DLL_CON 寄存器位域描述

DLL_CON	位	缺省值	描述
reserved	31:30	0x0	—
resync_dll_rd	29	0x0	内部采样时钟 DLL 重同步使能位
dll_bypass_rd	28	0x0	采样时钟 DLL bypass。该位置 1，DLL 控制器将不对 DLL 参数值进行微调。
resync_dll_pad	27	0x0	pad 时钟 DLL 重同步使能位
dll_bypass_pad	26	0x0	pad 时钟 DLL bypass。该位置 1，DLL 控制器将不对 DLL 参数值进行微调。
pm_init_start	25	0x0	DLL master 初始化开始位
pm_dll_lock_mode	24	0x0	DLL master 锁定模式。0，锁定一个周期；1，锁定半个周期
pm_dll_start_point	23:16	0x0	DLL master 初始化的起点值。该值应该低于一个周期的延迟值
pm_dll_increment	15:8	0x0	DLL master 初始化的步进值
pm_dll_adj_cnt	7:0	0x0	刷新锁定值的时间间隔

表 17-42 PARAM_DELAY 寄存器

寄存器名称	地址	读/写 (R/W)	功能描述	复位值
PARAM_DELAY	0xf8	R/W	DLL 延迟参数寄存器。理想情况下, DLL 一级延迟为 100ps, 共 256 级	0x0

表 17-43 PARAM_DELAY 寄存器位域描述

PARAM_DELAY	位	缺省值	描述
reserved	31:16	0x0	—
clk_rd_delay	15:8	0x0	内部采样时钟延迟参数, 用于调整控制器采样数据的采样点。DDR 模式下, 该值一般比 clk_pad_delay 大。
clk_pad_delay	7:0	0x0	时钟延迟参数。DDR 模式下, 此时的时钟与内部参考时钟的相位关系应该为 90°。例如, 内部参考时钟为 50MHz (20ns), 此时的 clk_pad_delay 值 应该为 50 (50*100ps)。

表 17-44 SDIO_EMMC_SEL 寄存器

寄存器名称	地址	读/写 (R/W)	功能描述	复位值
SDIO_EMMC_SEL	0xfc	R/W	总线模式选择	0x0

表 17-45 SDIO_EMMC_SEL 寄存器位域描述

SDIO_EMMC_SEL	位	缺省值	描述
reserved	31:4	0x0	—
bus_sel	1	0x0	SDIO 与 eMMC 总线模式选择。0 表示 eMMC 总线模式; 1 表示 SDIO 总线模式
data_mode	0	0x0	数据模式选择。0, 表示 SDR 数据模式; 1, 表示 DDR 数据模式

17.4 专用 DMA 控制器

17.4.1 结构描述

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

DMA 的传送数据的过程由三个阶段组成:

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

该 DMA 控制器限定为以字 (4Byte) 为单位的数据搬运。

DMA 控制器支持 64 位地址空间, 这主要通过 dma_64bit 来控制, 当该位设置为 1 时表示 DMA 控制器工作在 64 位地址空间, 反之为 32 位地址空间。在 64 位地址模式下, 需要扩

展 DMA_ORDER_ADDR 和 DMA_SADDR 为 64 位寄存器。

17.4.2 DMA 描述符

DMA_ORDER_ADDR_LOW

偏移地址： 0x0

复位值： 0x00000000

表 17-46 DMA_ORDER_ADDR_LOW 寄存器

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

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

DMA_SADDR

偏移地址： 0x4

复位值： 0x00000000

表 17-47 DMA_SADDR 寄存器

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

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

DMA_DADDR

偏移地址： 0x8

复位值： 0x00000000

表 17-48 DMA_DADDR 寄存器

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

DMA_LENGTH

偏移地址： 0xc

复位值： 0x00000000

表 17-49 DMA_LENGTH 寄存器

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

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

DMA_STEP_LENGTH

偏移地址：0x10

复位值：0x00000000

表 17-50 DMA_STEP_LENGTH 寄存器

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

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

DMA_STEP_TIMES

偏移地址：0x14

复位值：0x00000000

表 17-51 DMA_STEP_TIMERS 寄存器

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

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

DMA_CMD

偏移地址：0x18

复位值：0x00000000

表 17-52 DMA_CMD 寄存器

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

说明：dma_single_trans_over=1 指一次 DMA 操作执行结束，此时 length=0 且 step_times=1，开始取下个 DMA 操作的描述符。下个 DMA 操作的描述符地址保存在 DMA_ORDER_ADDR 寄存器中，如果 DMA_ORDER_ADDR 寄存器中 dma_order_en=0，则 dma_trans_over=1，整个 dma 操作结束，没有新的描述符要读；如果 dma_order_en=1，则 dma_trans_over 置为 0，开始读下个 dma 描述符。dma_int 为 DMA 的中断，如果没有中断屏蔽，在一次配置的 DMA 操作结束后发生中断。CPU 处理完中断后可以直接将其置低，也可以等到 DMA 进行下次传输时自动置低。dma_int_mask 为对应 dma_int 的中断屏蔽。dma_read_state 说明了 DMA 当前的读状态。dma_write_state 说明了 DMA 当前的写状态。DMA 写状态(WRITE_STATE[3:0])描述，DMA 包括以下几个写状态：

表 17-53 DMA 写状态描述

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：

表 17-54 DMA 读状态描述

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

DMA_ORDER_ADDR_HIGH

偏移地址： 0x20

复位值：0x00000000

表 17-55 DMA_ADDR_HIGH 寄存器

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

DMA_SADDR_HIGH

偏移地址：0x24

复位值：0x00000000

表 17-56 DMA_SADDR_HIGH 寄存器

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

17.5 软件配置流程

17.5.1 eMMC 正常读写流程

eMMC 正常读写，软件配置流程如下

1. 配置 emmc_con，使能时钟；
2. 配置 emmc_pre，输出预设频率的时钟；
3. 配置 emmc_bsize，设置一块数据的大小，单位为字节；
4. 配置 emmc_dtimer，设置超时计数，最大为 100ms；
5. 配置 emmc_int_en，设置中断使能，设置读写完成及错误中断使能；
6. 配置 emmc_dat_con，设置线宽数据模式；
7. 配置 bus_sel，设置 DDR 或 SDR 模式；
8. eMMC 设备初始化：初始化流程需要对命令通道进行操作，先配置 emmc_cmd_arg，填写命令参数，再 emmc_cmd_con，开始发送命令，通过检测 emmc_int_msk 检查命令发送完成中断；命令完成后读 emmc_rsp0~3，读 eMMC 发回来的回复；
9. 初始化完成后，可发送数据相关命令进行数据操作。配置 DMA（注：eMMC 控制器基址加上 0x800 为 DMA 读，基址加上 0x400 为 DMA 写）与 eMMC 控制器。命令发送同样时通过配置 emmc_cmd_arg 和 emmc_cmd_con 来完成。数据收发是否完成，通过检测 emmc_int_msk 来判断。

17.5.2 eMMC 初始化流程

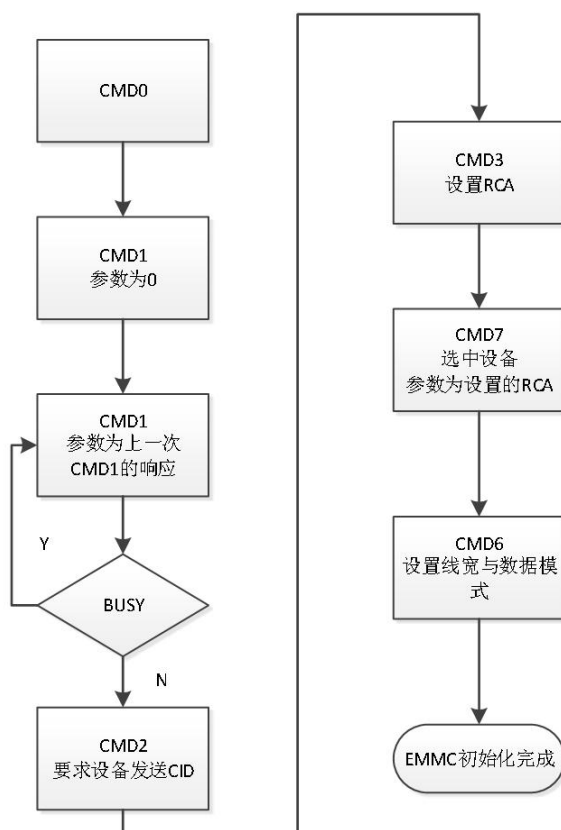


图17-1 eMMC 初始化流程图

17.5.3 DDR 模式设置

在开启 DDR 模式前，需要先初始化 eMMC 设备，同时设置 eMMC 设备为 DDR 模式；然后初始化 DLL，设置延迟值，最后将控制器设置为 DDR 模式。流程如下：

1. DLL 设置：pm_dll_lock_mode 置 1 锁定半个周期时钟；pm_dll_start_point 置为 0x10（该值应该大于 0 小于半周期）；pm_dll_adj_cnt 置为 0xff；pm_dll_increment，pm_dll_bypass，pm_dll_resync 置为 1；pm_init_start 置 1 开始 DLL 初始化；

2. DLL 锁定：DLL 设置完成后，检测 dll_init_done 寄存器，该值为 1 表示 DLL 完成锁定；

3. 配置延迟值：将 DLL 锁定值 pm_dll_value 除以 2 后的值写入 clk_pad_delay；clk_rd_delay 应该比 pm_dll_value/2 大，具体值视不同芯片和环境条件（PCB 走线，工作温度等）而定，软件需要根据实际情况对 clk_rd_delay 进行调整；

4. data_mode 置 1，开启 DDR 模式。

18 SDIO 控制器

18.1 功能概述

龙芯 2P0300 集成了两个 SDIO/eMMC 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：

- 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
- 专用独立 DMA 通道
- 1 位/4 位（宽总线）的 SD 模式

18.2 访问地址及引脚复用

SDIO 控制器内部寄存器的物理地址构成如下：

表 18-1 SDIO 内部寄存器物理地址构成

地址	设备	备注
0x1421_0000	SDIO0	32KB 大小寄存器配置空间
0x1421_8000	SDIO1	32KB 大小寄存器配置空间

对于 SDIO 模块，使用时要注意将对应的引脚设置为相应的功能。

与 SDIO 相关的引脚复用设置可查询节 中的 SDIO 功能引脚复用关系，并配置相应 GPIO 引脚复用配置寄存器实现。

18.3 寄存器描述

SDIO 控制器的寄存器详细说明如下：

表 18-2 SDI_CON 寄存器

寄存器名称	偏移地址	读/写(R/W)	功能描述	复位值
SDI_CON	0x00	R/W	SDIO 控制寄存器	0x0

表 18-3 SDI_CON 寄存器位域描述

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	7:1	0x0	-
enclk	0	0x0	SD 时钟输出使能

表 18-4 SDI_PRE 寄存器

寄存器名称	地址	读/写 (R/W)	功能描述	复位值
SDI_PRE	0x04	R/W	SDIO 预分频寄存器	0x1

表 18-5 SDI_PRE 寄存器位域描述

SDI_PRE	位	缺省值	描述
sdio_clk_rev_en	31	0x1	SDR 模式时，该位为 1，表示控制器输出的 sdio 数据与 sdio 时钟下降沿对齐；该位为 0，表示控制器输出的 sdio 数据与 sdio 时钟上升沿对齐。DDR 模式，该位必须置为 1。
Reserved	30:8	0x0	
Sdi_pre	7:0	0x1	SDIO 时钟预分频值，输出频率=PCLK/预分频值

表 18-6 SDI_CMD_ARG 寄存器

寄存器名称	偏移地址	读/写 (R/W)	功能描述	复位值
SDI_CMD_ARG	0x08	R/W	SDIO 命令参数寄存器	0x0

表 18-7 SDI_CMD_ARG 寄存器位域描述

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

表 18-8 SDI_CMD_CON 寄存器

寄存器名称	偏移地址	读/写 (R/W)	功能描述	复位值
SDI_CMD_CON	0x0c	R/W	SDIO 命令控制寄存器	0x0

表 18-9 SDI_CMD_CON 寄存器位域描述

SDI_CMD_CON	位	缺省值	描述
Reserved	31:18	0x0	-
func_num_abort	17:15	0x0	SDIO 卡时中断的功能号，用于多块读写时，硬件自动发送停止命令。如果 auto_stop_en 为 0，则此位无效
sdio_en	14	0x0	SDIO 使能信号。用于多块读写时，硬件自动发送停止命令，为 1 时发送 CMD52，为 0 是发送 CMD12。如果 auto_stop_en 为 0，则此位无效
check_on	13	0x0	是否检查 CRC，为 1 时有效
Auto_stop_en	12	0x0	硬件自动发送停止命令，多块读写时，是否硬件自动发送停止命令，为 1 时有效
Reserved	11	0x0	
long_rsp	10	0x0	是否为 136 位长响应，为 1 时表示长消息回复

SDI_CMD_CON	位	缺省值	描述
Wait_rsp	9	0x0	决定是否主机等待响应，为 1 是表示等待消息回复
CMST	8	0x0	命令开始，置 1 时开始，命令结束后硬件自动清零
cmd_index	7:0	0x0	带开始 2 位的命令索引（共 8 位）

表 18-10 SDI_CMD_STA 寄存器

寄存器名称	偏移地址	读/写 (R/W)	功能描述	复位值
SDI_CMD_STA	0x10	RO	SDIO 命令状态寄存器	0x0

表 18-11 SDI_CMD_STA 寄存器位域描述

SDI_CMD_STA	位	缺省值	描述
Reserved	31:15	0x0	—
cmd_sent_fin	14	0x0	命令发送完成（包含响应）标志位，为 1 表示命令发送完成及响应完成
auto_stop	13	0x0	硬件自动发送停止命令标志位，为 1 表示硬件自动发送停止命令，为 0 则没有
rsp_crc_err	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_end	11	0x0	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），或者 R1b 类型的命令，忙等待超时，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）

表 18-12 SDI_RSP0 寄存器

寄存器名称	偏移地址	读/写 (R/W)	功能描述	复位值
SDI_RSP0	0x14	RO	SDIO 命令响应寄存器 0	0x0

表 18-13 SDI_RSP0 寄存器位域描述

SDI_RESP0	位	缺省值	描述
sdi_resp0	31:0	0x0	卡状态[31:0]（短），卡状态[127:96]（长）长响应的配置间 sdi_cmd_con[10]

表 18-14 SDI_RSP1 寄存器

寄存器名称	偏移地址	读/写 (R/W)	功能描述	复位值
SDI_RSP1	0x18	RO	SDIO 命令响应寄存器 1	0x0

表 18-15 SDI_RSP1 寄存器位域描述

SDI_RESP1	位	缺省值	描述
sdi_respl	31:0	0x0	未使用（短），卡状态[95:64]（长）长响应的配置间 sdi_cmd_con[10]

表 18-16 SDI_RSP2 寄存器

寄存器名称	偏移地址	读/写 (R/W)	功能描述	复位值
SDI_RSP2	0x1c	RO	SDIO 命令响应寄存器 2	0x0

表 18-17 SDI_RSP2 寄存器位域描述

SDI_RESP2	位	缺省值	描述
sdi_resp2	31:0	0x0	未使用（短），卡状态[63:32]（长）长响应的配置间 sdi_cmd_con[10]

表 18-18 SDI_RSP3 寄存器

寄存器名称	偏移地址	读/写 (R/W)	功能描述	复位值
SDI_RSP3	0x20	RO	SDIO 命令响应寄存器 3	0x0

表 18-19 SDI_RSP3 寄存器位域描述

SDI_RESP3	位	缺省值	描述
sdi_resp3	31:0	0x0	未使用（短），卡状态[31:0]（长）长响应的配置间 sdi_cmd_con[10]

表 18-20 SDI_DTIMER 寄存器

寄存器名称	偏移地址	读/写 (R/W)	功能描述	复位值
SDI_DTIMER	0x24	R/W	SDIO 命令数据超时寄存器	0x0

表 18-21 SDI_DTIMER 寄存器位域描述

SDI_DTIMER	位	缺省值	描述
Reserved	31:24	0x0	-
sdi_dtimer	23:0	0x0	数据超时计数值，用分频后的时钟计数

表 18-22 SDI_BSIZE 寄存器

寄存器名称	偏移地址	读/写 (R/W)	功能描述	复位值
SDI_BSIZE	0x28	R/W	SDIO 块大小寄存器	0x0

表 18-23 SDI_BSIZE 寄存器位域描述

SDI_BSIZE	位	缺省值	描述
Reserved	31:12	0x0	
sdi_bsize	11:0	0x0	块大小值 (0~4095)

表 18-24 SDI_DAT_CON 寄存器

寄存器名称	偏移地址	读/写 (R/W)	功能描述	复位值
SDI_DAT_CON	0x2c	R/W	SDIO 数据控制寄存器	0x0

表 18-25 SDI_DAT_CON 寄存器位域描述

SDI_DAT_CON	位	缺省值	描述
Reserved	31:21	0x0	-
resume_rw	20	0x0	SDIO 挂起回复读写标志位。为 1 时，SDIO 挂起后恢复之前的写操作；为 0 时，恢复之前的读操作
IO_resume	19	0x0	SDIO 恢复请求。在 SDIO 设备进入挂起状态后，将此位写 1，并且 IO_suspend 位写 0 后，SDIO 设备恢复之前的操作。
IO_suspend	18	0x0	SDIO 挂起请求。写 1 后控制器会在合适的时机发送 CMD52 命令，通知 SDIO 设备进入挂起状态。恢复操作时需要将此位写 0。
RwaitReq	17	0x0	读等待请求。写 1 后控制器会在合适的时机将 DAT2 拉低，通知 SDIO 设备进入读等待状态。写 0 后恢复之前的读操作。
wide_mode	16	0x0	位宽选择位。为 1 表示 4 线模式，为 0 表示单线模式。
DMA_en	15	0x0	DMA 使能。为 1 时表示使能 DMA，为 0 表示禁止 DMA
DTST	14	0x0	数据传输开始，写 1 时数据传输开始，数据传输结束后硬件清零。
Reserved	13:12	0x0	-
Blk_num	11:0	0x0	读写操作的块数。

表 18-26 SDI_DAT_CNT 寄存器

寄存器名称	偏移地址	读/写 (R/W)	功能描述	复位值
SDI_DAT_CNT	0x30	R/W	SDIO 数据计数寄存器	0x0

表 18-27 SDI_DAT_CNT 寄存器位域描述

SDI_DAT_CNT	位	缺省值	描述
Reserved	31:24	0x0	-
blk_num_cnt	23:12	0x0	当前传输块的字节数
blk_cnt	11:0	0x0	当前传输的块数

表 18-28 SDI_DAT_STA 寄存器

寄存器名称	偏移地址	读/写 (R/W)	功能描述	复位值
SDI_DAT_STA	0x34	RO	SDIO 数据状态寄存器	0x0

表 18-29 SDI_DAT_STA 寄存器位域描述

SDI_DAT_STA	位	缺省值	描述
Reserved	31:17	0x0	-
suspend_on	16	0x0	为 1 时表示正在挂起状态
rst_suspend	15	0x0	为 1 表示正在挂起复位。用于 SDIO 设备挂起后，控制器复位 FIFO 和 DMA 请求
R1b_tout	14	0x0	为 1 表示 R1b 类型命令超时
data_start	13	0x0	为 1 表示数据传输开始
R1b_fin	12	0x0	检测到带 busy 状态的命令完成。当发送带 busy 状态的命令时，此位为 0；当 busy 状态结束时变成 1
auto_stop	11	0x0	为 1 时表示硬件正在自动发送停止命令
Reserved	10	0x0	-
r_wait_req	9	0x0	读等待发生。发送读等待请求信号到 SDIO 卡
SDIO_int	8	0x0	SDIO 中断标志位。为 1 表示检测到中断
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（比如编程时）。为 1 时标志忙结束
busy_fin	3	0x0	编程错误标志位（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。

表 18-30 SDI_FIFO_STA 寄存器

寄存器名称	偏移地址	读/写 (R/W)	功能描述	复位值
SDI_FIFO_STA	0x38	RO	SDIO FIFO 状态寄存器	0x0

表 18-31 SDI_FIFO_STA 寄存器位域描述

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	-

SDI_FIFO_STA	位	缺省值	描述
rx_full	8	0x0	Rx FIFO 满标志
rx_empty	7	0x0	Rx FIFO 空标志位
Reserved	6:0	0x0	-

表 18-32 SDI_INT_MASK 寄存器

寄存器名称	偏移地址	读/写 (R/W)	功能描述	复位值
SDI_INT_MASK	0x3c	R/W	SDIO 中断寄存器	0x0

表 18-33 SDI_INT_MASK 寄存器位域描述

SDI_INT_MASK	位	缺省值	描述
Reserved	31:10	0x0	
Rlb_fin_int	9	0x0	检测到 busy 结束中断，写 1 清零
rsp_crc_int	8	0x0	命令响应 CRC 错误中断，写 1 清零
cmd_tout_int	7	0x0	命令超时中断，写 1 清零
cmd_fin_int	6	0x0	发送完成中断，硬件清零
SDIO_int	5	0x0	检测到 SDIO 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

表 18-34 SDI_DAT 寄存器

寄存器名称	偏移地址	读/写 (R/W)	功能描述	复位值
SDI_DAT	0x40	RO	SDIO 命令数据寄存器	0x0

表 18-35 SDI_DAT 寄存器位域描述

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）

表 18-36 SDI_INT_EN 寄存器

寄存器名称	偏移地址	读/写 (R/W)	功能描述	复位值
SDI_INT_EN	0x64	R/W	SDIO 中断寄使能寄存器	0x0

表 18-37 SDI_INT_EN 寄存器位域描述

SDI_INT_EN	位	缺省值	描述
Reserved	31:10	0x0	-
Rlb_fin_int_en	9	0x0	Busy 结束中断使能，为 1 时有效
rsp_crc_int_en	8	0x0	命令响应 CRC 错误中断使能，为 1 时有效
cmd_tout_int_en	7	0x0	命令超时中断使能，为 1 时有效

SDI_INT_EN	位	缺省值	描述
cmd_fin_int_en	6	0x0	命令发送完成中断使能，为1时有效
SDIO_int_en	5	0x0	SDIO 中断使能，为1时有效
prog_err_int_en	4	0x0	SD 卡编程错误中断使能，为1时有效
crc_sta_int_en	3	0x0	数据发送后从设备返回 CRC 错误中断使能，为1时有效
dat_cec_int_en	2	0x0	数据接收 CRC 错误中断使能，为1时有效
dat_tout_int_en	1	0x0	数据超时中断使能，为1时有效
dat_fin_int_en	0	0x0	数据完成中断使能，为1时有效

表 18-38 DLL_MASTER_VAL 寄存器

寄存器名称	地址	读/写 (R/W)	功能描述	复位值
DLL_MASTER_VAL	0xf0	R	DLL master 锁定值	0x0

表 18-39 DLL_MASTER_VAL 寄存器位域描述

DLL_MASTER_VAL	位	缺省值	描述
reserved	31:9	0x0	
dll_init_done	8	0x0	DLL master 锁定完成标志位
pm_dll_value	7: 0	0x0	DLL master 锁定值

表 18-40 DLL_CON 寄存器

寄存器名称	地址	读/写 (R/W)	功能描述	复位值
DLL_CON	0xf4	R/W	DLL 控制寄存器	0x0

表 18-41 DLL_CON 寄存器位域描述

DLL_CON	位	缺省值	描述
reserved	31:30	0x0	-
resync_dll_rd	29	0x0	内部采样时钟 DLL 重同步使能位
dll_bypass_rd	28	0x0	采样时钟 DLL bypass。该位置 1，DLL 控制器将不对 DLL 参数值进行微调。
resync_dll_pad	27	0x0	pad 时钟 DLL 重同步使能位
dll_bypass_pad	26	0x0	pad 时钟 DLL bypass。该位置 1，DLL 控制器将不对 DLL 参数值进行微调。
pm_init_start	25	0x0	DLL master 初始化开始位
pm_dll_lock_mode	24	0x0	DLL master 锁定模式。0，锁定一个周期；1，锁定半个周期
pm_dll_start_point	23:16	0x0	DLL master 初始化的起点值。该值应该低于一个周期的延迟值
pm_dll_increment	15:8	0x0	DLL master 初始化的步进值

DLL_CON	位	缺省值	描述
pm_dll_adj_cnt	7:0	0x0	刷新锁定值的时间间隔

表 18-42 PARAM_DELAY 寄存器

寄存器名称	地址	读/写 (R/W)	功能描述	复位值
PARAM_DELAY	0xf8	R/W	DLL 延迟参数寄存器。理想情况下, DLL 一级延迟为 100ps, 共 256 级	0x0

表 18-43 PARAM_DELAY 寄存器位域描述

PARAM_DELAY	位	缺省值	描述
reserved	31:16	0x0	—
clk_rd_delay	15:8	0x0	内部采样时钟延迟参数, 用于调整控制器采样数据的采样点。DDR 模式下, 该值一般比 clk_pad_delay 大。
clk_pad_delay	7:0	0x0	时钟延迟参数。DDR 模式下, 此时的时钟与内部参考时钟的相位关系应该为 90°。例如, 内部参考时钟为 50MHz (20ns), 此时的 clk_pad_delay 值 应该为 50 (50*100ps)。

表 18-44 SDIO_EMMC_SEL 寄存器

寄存器名称	地址	读/写 (R/W)	功能描述	复位值
SDIO_EMMC_SEL	0xfc	R/W	总线模式选择	0x0

表 18-45 SDIO_EMMC_SEL 寄存器位域描述

SDIO_EMMC_SEL	位	缺省值	描述
reserved	31:2	0x0	—
bus_sel	1	0x0	SDIO 与 eMMC 总线模式选择。0 表示 SDIO 总线模式; 1 表示 eMMC 总线模式。切换至 eMMC 模式时, eMMC 最多只能支持四线模式。
data_mode	0	0x0	数据模式选择。0, 表示 SDR 数据模式; 1, 表示 DDR 数据模式

18.4 软件编程指南

18.4.1 SD Memory 卡软件编程说明

SD Memory 卡要想正常工作, 必须先初始化。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. 配置 sdi_int_en, 使能命令、数据完成及其他中断。

4. 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)。

5. 进行数据操作之前需要配置 Bsize 寄存器，Dtimer 寄存器。

6. 数据的操作必须要配置 DMA，配置 dat_con 寄存器并配置 DMA (注：SDIO 控制器基址加上 0x800 为 DMA 读，基址加上 0x400 为 DMA 写，寄存器与 eMMC 专用 DMA 相同)。

7. 读 sdi_int_msk 寄存器，检测是否传输完成，是否有错误。

8. 没有错误则完成一次数据传输，不需要软件发送停止命令。

18.4.2 SDIO 卡软件编程说明

SDIO 卡的初始化流程和 SD memory 卡不同，其初始化流程如下：

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。

3. 时钟来调整时序。

4. 配置 sdi_int_en, 使能命令、数据完成及其他中断。

5. 初始化流程如下：

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 (对 CCCR 的操作)：

CMD52 (复位)

CMD5 (等待上电完成)

CMD3 (获取 RCA)

CMD7 (选择相应 RCA 的卡)

CMD52（配置是否用 4bit 数据线传输）

CMD52（配置读写数据的块大小）

CMD52（打开 IO 中断使能）进行数据操作之前需要配置 Bsize 寄存器，Dtimer 寄存器。

6. 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）。

7. 发送读写数据的命令时，如果需要硬件自动发送停止命令，需要配置 auto_stop 和 sdio_en。读写数据时需要先读写支持的 Function，配置相应的 FBR 的指针寄存器，再发送多块读写（CMD53）或者单块读写（CMD52）命令进行读写。

8. 读 sdi_int_msk 寄存器，检测是否传输完成，是否有错误。

9. 没有错误则完成一次数据传输，不需要软件发送停止命令。

10. 如果检测到 IO 中断，控制器会置起响应的中断，但是不会停止当前的操作。

11. 对于读等待，控制器会在合适的时机将 DAT2 拉低，通知 SDIO 卡停止发送数据。所以如果当前正在传输数据过程中，控制器可能会在当前一块数据传输结束后，发出读等待信号，这时才不会接收下一块数据。

12. 对于挂起和恢复操作。有可能出现挂起嵌套情况，比如说操作 1 被操作 2 中断而挂起，然后操作 2 被操作 3 中断而挂起。挂起时中断的现场需要软件保存（如当前的读写标志位，当前传输的块数，地址等），进行入栈操作。恢复时需要软件再按相应的顺序出栈。

18.4.3 DDR 模式设置

在开启 DDR 模式前，需要先初始化 sdio 设备，同时设置 sdio 设备为 DDR 模式；然后初始化 DLL，设置延迟值，最后将控制器设置为 DDR 模式。流程如下：

1. DLL 设置：pm_dll_lock_mode 置 1 锁定半个周期时钟；pm_dll_start_point 置为 0x10（该值应该大于 0 小于半周期）；pm_dll_adj_cnt 置为 0xff；pm_dll_increment，pm_dll_bypass，pm_dll_resync 置为 1；pm_init_start 置 1 开始 DLL 初始化；
2. DLL 锁定：DLL 设置完成后，检测 dll_init_done 寄存器，该值为 1 表示 DLL 完成锁定；
3. 配置延迟值：将 DLL 锁定值 pm_dll_value 除以 2 后的值写入 clk_pad_delay；clk_rd_delay 应该比 pm_dll_value/2 大，具体值视不同芯片和环境条件（PCB 走线，工作温度等）而定，软件需要根据实际情况对 clk_rd_delay 进行调整；
4. data_mode 置 1，开启 DDR 模式。

18.5 支持 SDIO 型号

本节列出经过验证的可支持的 SDIO 卡型号，其它类型的 SDIO 卡未经验证，不保证与本控制器兼容。

Mem 卡：Kingston SD-C02G SDC/2GB

IO 卡（wifi）：maxwell sd8686

19 PWM 控制器

19.1 概述

龙芯 2P0300 集成 16 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和一路待测脉冲输入信号。系统时钟高达 125MHz，计数寄存器和参考寄存器均 32 位数据宽度。

19.2 访问地址及引脚复用

PWM 控制器内部寄存器的物理地址构成如下：

表 19-1 PWM 寄存器地址

地址	设备	备注
0x1420_7000	MAIN_PWM0~7	每个 PWM 占用 16B 寄存器配置空间： 0x1420_7000-pwm0, 0x1420_7010-pwm1, ...
0x1510_7000	PRT_PWM0~7	每个 PWM 占用 16B 寄存器配置空间： 0x1510_7000-pwm0, 0x1510_7010-pwm1, ...

对于 PWM 模块，使用时要注意将对应的引脚设置为相应的功能。与 PWM 相关的引脚复用设置可查询 PWM 功能引脚复用关系，并配置相应 GPIO 引脚复用配置寄存器实现。

19.3 寄存器描述

每路控制器共有三个寄存器，具体描述如下：

表 19-2 PWM 寄存器列表

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

表 19-3 PWM 控制寄存器设置

位域	名称	访问	复位值	说明
0	EN	R/W	0	计数器使能位 置 1 时：CNTR 用来计数 置 0 时：CNTR 停止计数（输出保持）
2: 1		Reserved	2' b0	预留
3	OE	R/W	0	脉冲输出使能控制位, 低有效 置 0 时：脉冲输出使能 置 1 时：脉冲输出屏蔽
4	SINGLE	R/W	0	单脉冲控制位 置 1 时：脉冲仅产生一次 置 0 时：脉冲持续产生

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

19.4 功能说明

19.4.1 脉宽调制功能

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

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

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

如果对两个缓冲寄存器都写 0，则输出为低电平；如果对 low_buffer 写 0，对 full_buffer 写 1，则输出高电平；如果写入 Low_buffer 的值不小于 full_buffer，则输

出低电平。但这三类数值都是不推荐的。

此外，缓冲寄存器的数值写入应当先于 CTRL 控制寄存器。

19.4.2 脉冲测量功能

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

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

待测脉冲应当是周期信号，且脉冲周期不应超出 32 位计数器能计量的范围。

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

若出现持续的周期超过 0xFFFF_FFF9 的脉冲，控制寄存器 INT 位会被置 1，表示待测脉冲超出了计量范围。

19.4.3 防死区功能

每组 8 路 PWM 都配备了防死区功能，可以防止 8 路脉冲输出同时发生跳变。

以 4 路 PWM 为例，将四路模块分别标记为 PWM_0、PWM_1、PWM_2、PWM_3，它们的优先级为 $0>1>2>3$ ，当对应 PWM 使能了防死区配置位后，若要同时产生跳变，在 PWM_0 跳变之后 PWM_1 才能跳变（低优先级的信号被“抹去”一个或多个系统时钟），依此类推。该优先级是固化的，不可配置。

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

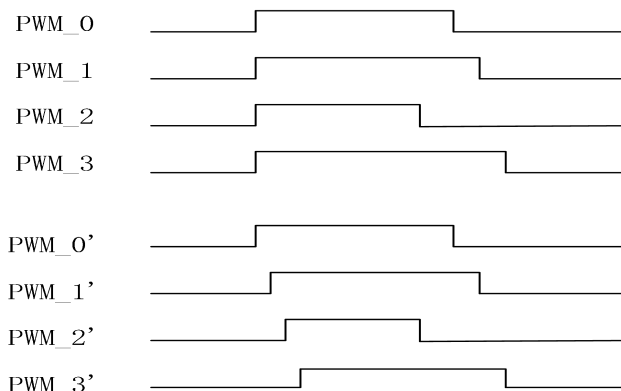


图19-1 防死区功能

20 HPET 控制器

20.1 概述

龙芯 2P0300 芯片集成 6 个 HPET 定时控制器，HPET（High Precision Event Timer，高精度事件定时器）定义了一组新的定时器，这组定时器被操作系统使用，用来给线程调度，内核以及多媒体定时器服务器等产生中断。操作系统可以将不同的定时器分配给不同的应用程序使用。通过配置，每个定时器都能独立产生中断。

这组定时器由一个向上累加的主计时器（up-counter）以及一组比较器构成。这个计时器以各系统下 APB 总线频率向上累加，因此当软件两次读取计时器的值时，除非遇到计时器溢出，否则第二次读取的值总是比第一次读取的值大。而每个定时器都包含一个 match 寄存器以及一个比较器。当 match 寄存器的值与主计时器相等时，那么定时器产生中断。部分定时器可产生周期性中断。

HPET 模块包括一个主计数器（main count）以及三个比较器（comparator），且他们的宽度都是 32 位。在这三个比较器中，有且仅有一个比较器支持周期性中断（periodic-capable）；这三个比较器都支持非周期性中断。

20.2 访问地址

HPET 控制器内部寄存器的物理地址构成如下：

表 20-1 HPET 寄存器地址

地址	设备	备注
0x1420_4000	MSYS_HPETO	2KB 寄存器配置空间
0x1420_4800	MSYS_HPET1	2KB 寄存器配置空间
0x1510_b000	PRT_HPETO	2KB 寄存器配置空间
0x1510_b800	PRT_HPET1	2KB 寄存器配置空间
0x1520_5000	SCA_HPETO	2KB 寄存器配置空间
0x1520_5800	SCA_HPET1	2KB 寄存器配置空间

20.3 寄存器描述

下表列出了 HPET 的寄存器：

表 20-2 HPET 寄存器

寄存器偏移地址	寄存器	类型
000-007h	General Capabilities and ID Register	只读
008-00Fh	Reserved	
010-017h	General Configuration Register	读/写
018-01Fh	Reserved	R/WC
020-027h	General Interrupt Status Register	R/W
028-0EFh	Reserved	

寄存器偏移地址	寄存器	类型
0F0-0F7h	Main Counter Value Register	R/W
100-107h	Timer 0 Configuration and Capability Register	R/W
108-10Fh	Timer 0 Comparator Value Register	R/W
110-11Fh	Reserved	
120-127h	Timer 1 Configuration and Capability Register	R/W
128-12Fh	Timer 1 Comparator Value Register	R/W
130-13Fh	Reserved	
140-147h	Timer 2 Configuration and Capability Register	R/W
148-14Fh	Timer 2 Comparator Value Register	R/W
150-15Fh	Reserved	

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

20.3.1 General Capabilities and ID Register

表 20-3 General Capabilities ID 寄存器

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

20.3.2 General Configuration Register

表 20-4 General Configuration 寄存器

位	名称	描述	读写特性
63: 1	Reserved		

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

20.3.3 General Interrupt Status Register

表 20-5 General Interrupt Status 寄存器

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

各个定时器的中断触发模式由各自 Configuration and Capabilities 寄存器的 Tn_TYPE_CNF 位确定。

20.3.4 Main Counter Value Register

表 20-6 Main Counter Value 寄存器

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

20.3.5 Timer N Configuration and Capabilities Register

表 20-7 Timer N Configuration Capabilities 寄存器

位	名称	描述	读写特性
63: 9	Reserved		
8	Tn_32MODE_CNF	Timer n 32-bit 模式 (N 为 0-2)。当定时器为 32 位时, 这位为 0, 且只读	RO
7	Reserved		RO

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

20.3.6 Timer N Comparator Value Register

表 20-8 Timer N Comparator Value 寄存器

位	名称	描述	读写特性
63: 32	Reserved		

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

21 DMA 控制器

21.1 概述

龙芯 2P0300 集成通用 DMA 可以实现从内存到内存，设备到内存，内存到设备的数据传输。其内部有 8 个通道，每个通道都支持内存到内存传输，但是不同设备被分配到了不同的通道上。

每个通道在配置寄存器都可以配置其优先级，每当遇到两个优先级不同的通道发出 DMA 传输请求时，高优先级会赢得仲裁，低优先级请求会在传输高优先级请求的传输间隙开启。相同优先级的通道，编号越小优先级越高。

DMA 具备循环模式，可用于处理循环缓冲区和连续数据传输，如 ADC 扫描模式，当传输个数变为 0 时，该通道所有寄存器会被复位为传输开启时的值。

DMA 支持存储器到存储器模式，不需要外设请求，在配置 DMA_CCRx 的 EN 位之后开始传输。此模式不能和循环模式同时开启。

21.2 访问地址

表 21-1 DMA 寄存器地址

寄存器基址	设备	备注
0x1420_8000	MAIN-APB-DMA	主系统 APB 设备 DMA 寄存器配置空间，大小共 4KB，仅供主系统 APB 设备使用（MAIN-UART/I ² C/SPI）。
0x1510_a000	PRT-APB-DMA	打印系统 APB 设备 DMA 寄存器配置空间，大小共 4KB，仅供打印系统 APB 设备使用（PRT-UART/I ² C/ADC）。

21.3 寄存器描述

部分表格内以 x 代替通道编号 0-7。

表 21-2 DMA 寄存器列表

偏移	名称	描述
0x00	DMA_ISR	DMA 中断状态寄存器
0x04	DMA_IFCR	DMA 中断标志清除寄存器
0x08	DMA_CCR0	DMA 通道 0 配置寄存器
0x0c	DMA_CNDTR0	DMA 通道 0 传输数量寄存器
0x10	DMA_CPAR0	DMA 通道 0 外设地址寄存器
0x14	DMA_CMAR0	DMA 通道 0 储存地址寄存器
0x18		
0x1c	DMA_CCR1	DMA 通道 1 配置寄存器
0x20	DMA_CNDTR1	DMA 通道 1 传输数量寄存器
0x24	DMA_CPAR1	DMA 通道 1 外设地址寄存器

偏移	名称	描述
0x28	DMA_CMAR1	DMA 通道 1 储存地址寄存器
0x2c		
0x30	DMA_CCR2	DMA 通道 2 配置寄存器
0x34	DMA_CNDTR2	DMA 通道 2 传输数量寄存器
0x38	DMA_CPAR2	DMA 通道 2 外设地址寄存器
0x3c	DMA_CMAR2	DMA 通道 2 储存地址寄存器
0x40		
0x44	DMA_CCR3	DMA 通道 3 配置寄存器
0x48	DMA_CNDTR3	DMA 通道 3 传输数量寄存器
0x4c	DMA_CPAR3	DMA 通道 3 外设地址寄存器
0x50	DMA_CMAR3	DMA 通道 3 储存地址寄存器
0x54		
0x58	DMA_CCR4	DMA 通道 4 配置寄存器
0x5c	DMA_CNDTR4	DMA 通道 4 传输数量寄存器
0x60	DMA_CPAR4	DMA 通道 4 外设地址寄存器
0x64	DMA_CMAR4	DMA 通道 4 储存地址寄存器
0x68		
0x6c	DMA_CCR5	DMA 通道 5 配置寄存器
0x70	DMA_CNDTR5	DMA 通道 5 传输数量寄存器
0x74	DMA_CPAR5	DMA 通道 5 外设地址寄存器
0x78	DMA_CMAR5	DMA 通道 5 储存地址寄存器
0x7c		
0x80	DMA_CCR6	DMA 通道 6 配置寄存器
0x84	DMA_CNDTR6	DMA 通道 6 传输数量寄存器
0x88	DMA_CPAR6	DMA 通道 6 外设地址寄存器
0x8c	DMA_CMAR6	DMA 通道 6 储存地址寄存器
0x90		
0x94	DMA_CCR7	DMA 通道 7 配置寄存器
0x98	DMA_CNDTR7	DMA 通道 7 传输数量寄存器
0x9c	DMA_CPAR7	DMA 通道 7 外设地址寄存器
0xa0	DMA_CMAR7	DMA 通道 7 储存地址寄存器

21.3.1 DMA 中断状态寄存器 (DMA_ISR)

偏移量: 0x00
复位值: 0x00000000

表 21-3 DMA 中断状态寄存器

位域	名称	访问	描述
$4*x + 3$	TEIF _x	R	通道 x 传输错误标志 0: 通道 x 无传输错误事件; 1: 通道 x 有传输错误事件。
$4*x + 2$	HTIF _x	R	通道 x 传输过半标志 0: 通道 x 无传输过半事件; 1: 通道 x 有传输过半事件。 注: 该标志位仅在传输个数为偶数时有效。
$4*x + 1$	TCIF _x	R	通道 x 传输完成标志 0: 通道 x 无传输完成事件; 1: 通道 x 有传输完成事件。
$4*x$	GIF _x	R	通道 x 全局中断标志 0: 通道 x 无传输错误/过半/完成事件; 1: 通道 x 有传输错误/过半/完成事件。

21.3.2 DMA 中断标志清除寄存器 (DMA_IFCR)

偏移量: 0x04

复位值: 0x00000000

表 21-4 DMA 中断标志清除寄存器

位域	名称	访问	描述
$4*x + 3$	CTEIF _x	RW	清除通道 x 传输错误标志 0: 无效; 1: 清除 DMA_ISR 寄存器中对应的传输错误事件标志。
$4*x + 2$	CHTIF _x	RW	清除通道 x 传输过半标志 0: 无效; 1: 清除 DMA_ISR 寄存器中对应的传输过半事件标志。
$4*x + 1$	CTCIF _x	RW	清除通道 x 传输完成标志 0: 无效; 1: 清除 DMA_ISR 寄存器中对应的传输完成事件标志。
$4*x$	CGIF _x	RW	清除通道 x 全局中断标志 0: 无效; 1: 清除 DMA_ISR 寄存器中对应的传输错误/过半/完成事件标志。

21.3.3 DMA 通道 x 配置寄存器 (DMA_CCR_x)

偏移量: $0x08 + 0x14 * x$

复位值: 0x00000000

表 21-5 DMA 通道 x 配置寄存器

位域	名称	访问	描述
31:15	Reserved	—	—

位域	名称	访问	描述
14	MEM2MEM	RW	存储器到存储器模式，该位由软件配置和清除。 0：非存储器（设备）到存储器（内存）模式； 1：启动存储器（内存）到存储器（内存）模式。
13:12	PL	RW	通道优先级，该位域由软件配置和清除。 00：低；01：中；10：高；11：最高。
11:10	MSIZE	RW	存储器数据宽度，该位域由软件配置和清除。 00：8 位；01：16 位；10：32 位；11：保留。
9:8	PSIZE	RW	外设数据宽度，该位域由软件配置和清除。 00：8 位；01：16 位；10：32 位；11：保留。
7	MINC	RW	存储器地址增量模式，该位域由软件配置和清除。 0：无效；1：有效。
6	PINC	RW	外设地址增量模式，该位域由软件配置和清除。 0：无效；1：有效。
5	CIRC	RW	循环模式，该位域由软件配置和清除。 0：无效；1：有效。
4	DIR	RW	数据传输方向，该位域由软件配置和清除。 0：从外设读；1：从存储器读。
3	TEIE	RW	传输错误中断使能，该位域由软件配置和清除。 0：无效；1：有效。
2	HTIE	RW	传输过半中断使能，该位域由软件配置和清除。 0：无效；1：有效。
1	TCIE	RW	传输完成中断使能，该位域由软件配置和清除。 0：无效；1：有效。
0	EN	RW	通道开启，该位域由软件配置和清除。 0：无效；1：有效。

21.3.4 DMA 通道 x 传输数量寄存器（DMA_CNDTRx）

偏移量：0x0c + 0x14 * x

复位值：0x00000000

表 21-6 DMA 通道 x 传输数量寄存器

位域	名称	访问	描述
31:0	NDT	RW	数据传输个数， 这个寄存器只能通道停用写入。通道开启后变为只读，此时读数为待传输个数。 注：DMA 单次传输最大支持范围为 4294967295，当 msize/psize 均为 0 时，NDT 最大可配置为 4294967295；当 msize/psize 任一为 1 时，NDT 最大可配置为 2147483647；当 msize/psize 任一为 1 时，NDT 最大可配置为 1073741823。

21.3.5 DMA 通道 x 外设地址寄存器 (DMA_CPARx)

偏移量: $0x10 + 0x14 * x$

复位值: $0x00000000$

表 21-7 DMA 通道 x 外设地址寄存器

位域	名称	访问	描述
31:0	PADDR	RW	外设地址， 外设数据起始地址。 通道开启后变为只读。

21.3.6 DMA 通道 x 存储器地址寄存器 (DMA_CMARx)

偏移量: $0x14 + 0x14 * x$

复位值: $0x00000000$

表 21-8 DMA 通道 x 存储器地址寄存器

位域	名称	访问	描述
31:0	MADDR	RW	存储器地址， 存储器数据起始地址。 通道开启后变为只读。

21.4 功能描述

21.4.1 配置流程

假设我们要启动一次 DMA 传输，我们需要执行下列初始化操作：

- 1) 在 DMA_CPARx 设置外设寄存器地址；
- 2) 在 DMA_CMARx 设置存储器寄存器地址；
- 3) 在 DMA_CNDTRx 设置要传输的数据个数；
- 4) 在 DMA_CCRx 设置要通道优先级；
- 5) 在 DMA_CCRx 设置要数据传输方向，循环模式，外设和存储器增量模式，外设和存储器数据宽度，传输中断使能；
- 6) 在 DMA_CCRx 设置 ENABLE, 启动该通道；

21.4.2 宽度和对齐方式

以 $PINC=MINC=1$ 为例，当源宽度与目的宽度不一致时，若源宽度大于目的宽度，则写入目的地址的数据为源数据自低位开始截取目的宽度的结果；若源宽度小于目的宽度，则写入目的地址的数据为源数据高位补 0 至目的宽度的结果。

21.4.3 通道映射

外设各请求类型与各通道的映射见下表。

表 21-9 MAIN-APB-DMA 外设请求的通道映射

外设 DMA 通道	CH0	CH1	CH2	CH3	CH4	CH5	CH6	CH7
MAIN-UART0	RX	TX						
MAIN-UART1			RX	TX				
MAIN-I2C					RX	TX		
MAIN-SPI-IO							RX	TX

ADC 通道请求与各通道的映射可在芯片通用配置寄存器，可选的通道映射见下表，其中“*”符号代表默认映射配置。

表 21-10 PRT-APB-DMA 外设请求的通道映射

外设 DMA 通道	CH0	CH1	CH2	CH3	CH4	CH5	CH6	CH7
PRT-UART0	RX	TX						
PRT-UART1			RX	TX				
PRT-I2C					RX	TX		
PRT-ADC	RX*	RX	RX	RX	RX	RX	RX	RX

22 ADC 控制器

22.1 概述

龙芯 2P0300 集成 12 比特最高采样率 2MSPS 的数模转换控制器（ADC），可用于测量 8 路模拟电压输入，并支持 DMA 传输。各通道的 A/D 转换可以按照预设的顺序进行单次、连续、扫描或间断执行，转换结果会以左对齐或右对齐的方式储存。

22.2 工作模式

表 22-1 工作模式配置

		连续	扫描	间断	
		CONT	SCAN	DISCEN	JDISCEN
单次转换模式		0	0	0	0
连续转换模式		1	0	x	x
扫描模式	单次	0	1	0	0
	连续	1	1	0	0
间断模式	规则组	0	1	1	0
	注入组	0	1	0	1

- 单次转换模式

单次转换模式下，ADC 在触发后只执行一次转换。

- 连续转换模式

在连续转换模式中，ADC 在前一组任务完成后马上就重启转换。

- 扫描模式

此模式用来顺序转换一组预设的模拟通道。

- 间断模式

一次触发，顺序转换 DISCNUM 个模拟通道后停止转换；再次触发，从上次停止的通道顺序转换 DISCNUM 个通道，直至达到预设的通道数。

22.3 其他功能

22.3.1 注入通道管理

触发注入：JAUTO=0，SCAN=1

1. 利用外部触发或通过设置 ADC_CR2 寄存器的 ADON 位，启动一组规则通道的转换。
2. 如果在规则通道转换期间产生一外部注入触发，根据 jtrigmod 配置复位当前转换，以单次扫描方式转换注入通道序列。
3. 然后，恢复上次被中断的规则组通道转换。如果在注入转换期间产生一规则事件，注入转换不会被中断，但是规则序列将在注入序列结束后被执行。

自动注入：JAUTO=1，SCAN=1

1. 如果设置了 JAUTO 位，在规则组通道之后，注入组通道被自动转换。

22.3.2 数据对齐

ADC_CR2 寄存器中的 ALIGN 位选择转换后数据储存的对齐方式。数据可以左对齐或右对齐，如下图所示。

数据右对齐

注入组

SEXT	SEXT	SEXT	SEXT	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
------	------	------	------	-----	-----	----	----	----	----	----	----	----	----	----	----

规则组

0	0	0	0	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
---	---	---	---	-----	-----	----	----	----	----	----	----	----	----	----	----

数据左对齐

注入组

SEXT	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0
------	-----	-----	----	----	----	----	----	----	----	----	----	----	---	---	---

规则组

D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0
-----	-----	----	----	----	----	----	----	----	----	----	----	---	---	---	---

图22-1 数据对齐方式

注入组通道转换的数据值已经减去了在 ADC_JOFRx 寄存器中定义的偏移量，因此结果可以是一个负值。SEXT 位是扩展的符号值。

对于规则组通道，不需减去偏移值，因此只有 12 个位有效。

22.3.3 可编程的通道采样时间

ADC 使用若干个 ADC_CLK 周期对输入电压采样，采样周期数目可以通过 ADC_SMPR1 和 ADC_SMPR2 寄存器中的 SMP[2:0] 位更改。每个通道可以分别用不同的时间采样。

总转换时间如下计算：

$$TCONV = \text{采样时间} + 13 \text{ 个周期}$$

22.3.4 模拟看门狗

如果被 ADC 转换的模拟电压低于低阈值或高于高阈值，AWD 模拟看门狗状态位被设置。阈值位于 ADC_HTR 和 ADC_LTR 寄存器的最低 12 个有效位中。通过设置 ADC_CR1 寄存器的 AWDIE 位以允许产生相应中断。

阈值独立于由 ADC_CR2 寄存器上的 ALIGN 位选择的数据对齐模式。比较是在对齐之前完成的。

22.3.5 DMA 请求

因为规则通道转换的值储存在一个仅有的数据寄存器中，所以当转换多个规则通道时需要使用 DMA，这可以避免丢失已经存储在 ADC_DR 寄存器中的数据。

只有在规则通道的转换结束时才产生 DMA 请求，并将转换的数据从 ADC_DR 寄存器传输到用户指定的目的地址。

22.3.6 外部触发

转换可以由外部事件触发(例如定时器捕获, EXTI 线)。如果设置了 EXTTRIG 控制位, 则外部事件就能够触发转换。EXTSEL[2:0]和 JEXTSEL[2:0]控制位允许应用程序选择 6 个可能的事件中的某一个, 可以触发规则和注入组的采样。

22.4 寄存器描述

ADC 的寄存器空间的基地址为 0x1510_5000, 包含 20 个 32 位寄存器。

22.4.1 寄存器地址列表

表 22-2 ADC 寄存器列表

名称	偏移地址	位宽	描述
ADC_SR	0x00	32	ADC 状态寄存器
ADC_CR1	0x04	32	ADC 控制寄存器 1
ADC_CR2	0x08	32	ADC 控制寄存器 2
ADC_SMPR1	0x0C	32	ADC 采样时间寄存器 1
ADC_SMPR2	0x10	32	ADC 采样时间寄存器 2
ADC_JOFR1	0x14	32	ADC 注入通道偏移寄存器 1
ADC_JOFR2	0x18	32	ADC 注入通道偏移寄存器 2
ADC_JOFR3	0x1C	32	ADC 注入通道偏移寄存器 3
ADC_JOFR4	0x20	32	ADC 注入通道偏移寄存器 4
ADC_HTR	0x24	32	ADC 看门狗高阈值寄存器
ADC_LTR	0x28	32	ADC 看门狗低阈值寄存器
ADC_SQR1	0x2C	32	ADC 规则序列寄存器 1
ADC_SQR2	0x30	32	ADC 规则序列寄存器 2
ADC_SQR3	0x34	32	ADC 规则序列寄存器 3
ADC_JSQR	0x38	32	ADC 注入序列寄存器
ADC_JDR1	0x3C	32	ADC 注入数据寄存器 1
ADC_JDR2	0x40	32	ADC 注入数据寄存器 2
ADC_JDR3	0x44	32	ADC 注入数据寄存器 3
ADC_JDR4	0x48	32	ADC 注入数据寄存器 4
ADC_DR	0x4C	32	ADC 规则数据寄存器

22.4.2 ADC_SR

中文名: ADC 状态寄存器

寄存器位宽: [31:0]

偏移量: 0x00

复位值: 0x00000000

表 22-3 ADC 状态寄存器

位域	位域名称	访问	描述
31:5	Reserved		保留
4	STRT	RW	规则通道开始标志位 0: 规则通道转换未开始 1: 规则通道转换已开始
3	JSTRT	RW	注入通道开始标志位 0: 注入通道组转换未开始 1: 注入通道组转换已开始

位域	位域名称	访问	描述
31:5	Reserved		保留
2	JEOC	RW	注入通道转换结束标志位 0: 转换未完成 1: 转换完成
1	EOC	RW	转换结束标志位 0: 转换未完成 1: 转换完成
0	AWD	RW	模拟看门狗标志位 0: 没有发生模拟看门狗事件 1: 发生模拟看门狗事件

22.4.3 ADC_CR1

中文名: ADC 控制寄存器 1

寄存器位宽: [31: 0]

偏移量: 0x04

复位值: 0x00000000

表 22-4 ADC 控制寄存器 1

位域	位域名称	访问	描述
31:30	OPS	RW	ADC 控制信号相位调节 0: 与 ADCCLK 上升沿同时刻 1: 较 ADCCLK 上升沿前一 pclk 2: 较 ADCCLK 上升沿后一 pclk
29:24	CLKDIV[5:0]	RW	ADCCLK 分频系数 [5:0] 位
23	AWDEN	RW	规则通道启用模拟看门狗 0: 在规则通道上禁用模拟看门狗 1: 在规则通道上启用模拟看门狗
22	JAWDEN	RW	注入通道启用模拟看门狗 0: 在注入通道上禁用模拟看门狗 1: 在注入通道上启用模拟看门狗
21	Reserved		保留
20	DIFFMOD	RW	差分模式使能 启用时仅可对低四对模拟输入进行比较 0: 不启用差分模式 1: 启用差分模式
19:16	Reserved		保留
15:13	DISCNUM	RW	间断模式通道计数 在间断模式下, 收到外部触发后转换规则通道的数目 000: 1 个通道 001: 2 个通道 111: 8 个通道
12	JDISCEN	RW	在注入通道上的间断模式 用于开启或关闭注入通道组上的间断模式 0: 注入通道停用间断模式 1: 注入通道启用间断模式
11	DISCEN	RW	在规则通道上的间断模式 用于开启或关闭规则通道组上的间断模式 0: 规则通道停用间断模式 1: 规则通道启用间断模式

位域	位域名称	访问	描述
10	JAUTO	RW	自动注入通道组开转换 用于开启或关闭规则通道组转换结束后自动的注入通道组转换 0: 关闭自动的注入通道组转换 1: 开启自动的注入通道组转换
9	AWDSGL	RW	扫描模式中在一个单一的通道上使用看门狗 用于开启或关闭由 AWDCH[4:0]位指定的通道上的模拟看门狗功能 0: 在所有的通道上使用模拟看门狗 1: 在单一通道上使用模拟看门狗
8	SCAN	RW	扫描模式 用于开启或关闭扫描模式。在扫描模式中，转换由 ADC_SQRx 或 ADC_JSQRx 寄存器选中的通道。 只有在最后一个通道转换完毕后，才会根据 EOCIE 或 JEOCIE 位产生 EOC 或 JEOC 中断 0: 关闭扫描模式 1: 使用扫描模式
7	JEOCIE	RW	注入通道中断使能 用于禁止或允许所有注入通道转换结束后产生中断。 0: 禁止 JEOC 中断 1: 允许 JEOC 中断。当硬件设置 JEOC 位时产生中断
6	AWDIE	RW	模拟看门狗中断使能 用于禁止或允许模拟看门狗产生中断。在扫描模式下，如果看门狗检测到超范围的数值时，只有在设置了该位时扫描才会中止。 0: 禁止模拟看门狗中断 1: 允许模拟看门狗中断
5	EOCIE	RW	EOC 中断使能 用于禁止或允许转换结束后产生中断。 0: 禁止 EOC 中断 1: 允许 EOC 中断。当硬件设置 EOC 位时产生中断
4:0	AWDCH	RW	模拟看门狗通道选择 用于选择模拟看门狗保护的输入通道。 00000: ADC 模拟输入通道 0 00001: ADC 模拟输入通道 1 01111: ADC 模拟输入通道 15 10000: ADC 模拟输入通道 16 10001: ADC 模拟输入通道 17 保留所有其他数值

22.4.4 ADC_CR2

中文名: ADC 控制寄存器 2

寄存器位宽: [31:0]

偏移量: 0x08

复位值: 0x00000000

表 22-5 ADC 控制寄存器 1

位域	位域名称	访问	描述
31	Reserved		保留
30	ADCEGE	RW	ADC 时钟触发沿选择 0: 上升沿触发 1: 下降沿触发

位域	位域名称	访问	描述
29:26	CLKDIV[9:6]	RW	ADC 时钟分频系数[9:6]
25:24	JTRIGMOD	RW	注入触发模式选择 0: 结束当前规则组转换并立即开始注入通道转换 1: 结束当前规则组转换并在插入一拍 ADC 复位控制信号后开始注入通道转换 2: 在当前规则组转换结束后开始注入通道转换
23	Reserved		保留
22	SWSTART	RW	开始转换规则通道 由软件设置该位以启动转换, 转换开始后硬件马上清除此位。如果在 extsel[2:0]位中选择了 swstart 为触发事件, 该位用于启动一组规则通道的转换 0: 复位状态 1: 开始转换规则通道
21	JSWSTART	RW	开始转换注入通道 由软件设置该位以启动转换, 软件可清除此位或在转换开始后硬件马上清除此位。如果在 jextsel[2:0]位中选择了 jswstart 为触发事件, 该位用于启动一组注入通道的转换 0: 复位状态 1: 开始转换注入通道
20	EXTTRIG	RW	规则通道的外部触发转换模式 该位由软件设置和清除, 用于开启或禁止可以启动规则通道组转换的外部触发事件 0: 不用外部事件启动转换 1: 使用外部事件启动转换
19:17	EXTSEL	RW	选择启动规则通道组转换的外部事件 这些位选择用于启动规则通道组转换的外部事件触发配置如下 3'b000: ATIM_CC1 事件 3'b001: ATIM_CC2 事件 3'b010: ATIM_CC3 事件 3'b011: GTIM_CC2 事件 3'b110: EXTI 线 11 3'b111: swstart
16	Reserved		保留
15	JEXTTRIG	RW	注入通道的外部触发转换模式 该位由软件设置和清除, 用于开启或禁止可以启动注入通道组转换的外部触发事件 0: 不用外部事件启动转换 1: 使用外部事件启动转换
14:12	JEXTSEL	RW	选择启动注入通道组转换的外部事件 这些位选择用于启动规则通道组转换的外部事件触发配置如下 3'b000: ATIM_TRGO 事件 3'b001: ATIM_CC4 事件 3'b010: GTIM_TRGO 事件 3'b011: GTIM_CC1 事件 3'b110: EXTI 线 15 3'b111: JSWSTART
11	ALIGN	RW	数据对齐 0: 右对齐 1: 左对齐
10:9	Reserved		保留
8	DMA	RW	直接存储器访问模式 0: 不使用 DMA 模式 1: 使用 DMA 模式

位域	位域名称	访问	描述
7:3	RSTCAL	RW	该位由软件设置并由硬件清除。在校准寄存器被初始化后该位将被清除 0: 校准寄存器已初始化 1: 初始化校准寄存器
2	CAL	RW	AD 校准 该位由软件设置以开始校准, 并在校准结束时由硬件清除 0: 校准完成 1: 开始校准
1	CONT	RW	连续转换 该位由软件设置和清除。如果设置了此位, 则转换将连续进行直到该位被清除 0: 单次转换模式 1: 连续转换模式
0	ADON	RW	开/关 AD 转换器 该位由软件设置和清除。当该位为 '0' 时, 写入 '1' 将把 ADC 从断电模式下唤醒 该位为 '1' 时, 写入 '1' 将启动转换 0: 关闭 ADC 转换/校准, 并进入断电模式 1: 开启 ADC 并启动转换

22.4.5 ADC_SMPR1

中文名: ADC 采样时间寄存器 1

寄存器位宽: [31:0]

偏移量: 0x0c

复位值: 0x00000000

表 22-6 ADC 采样时间寄存器 1

位域	位域名称	访问	描述
31:24	Reserved		保留
23:21	SMP18	RW	通道 18 的采样时间 建议至少配置为 2 个周期 000: 1 个周期 001: 2 个周期 010: 4 个周期 011: 8 个周期 100: 16 个周期 101: 32 个周期 110: 64 个周期 111: 128 个周期
20:18	SMP17	RW	通道 17 的采样时间
17:15	SMP16	RW	通道 16 的采样时间
14:12	SMP15	RW	通道 15 的采样时间
11:9	SMP14	RW	通道 14 的采样时间
8:6	SMP13	RW	通道 13 的采样时间
5:3	SMP12	RW	通道 12 的采样时间
2:0	SMP11	RW	通道 11 的采样时间

22.4.6 ADC_SMPR2

中文名: ADC 采样时间寄存器 2

寄存器位宽： [31:0]

偏移量： 0x10

复位值： 0x00000000

表 22-7 ADC 采样时间寄存器 2

位域	位域名称	访问	描述
29:27	SMP10	RW	通道 10 的采样时间
26:24	SMP9	RW	通道 9 的采样时间
23:21	SMP8	RW	通道 8 的采样时间
20:18	SMP7	RW	通道 7 的采样时间
17:15	SMP6	RW	通道 6 的采样时间
14:12	SMP5	RW	通道 5 的采样时间
11:9	SMP4	RW	通道 4 的采样时间
8:6	SMP3	RW	通道 3 的采样时间
5:3	SMP2	RW	通道 2 的采样时间
2:0	SMP1	RW	通道 1 的采样时间

22.4.7 ADC_JOFR_x (1-4)

中文名： ADC 输入通道数据偏移寄存器 x (1-4)

寄存器位宽： [31:0]

偏移量： 0x14, 0x18, 0x1c, 0x20

复位值： 0x00000000

表 22-8 ADC 控制寄存器 1

位域	位域名称	访问	描述
31:12	Reserved		保留
11:0	JOFFSET	RW	注入通道 x 的数据偏移 当转换注入通道时，这些位定义了用于从原始转换数据中减去的数值。转换的结果可以在 ADC_JDR _x 寄存器中读出。

22.4.8 ADC_HTR

中文名： ADC 看门狗高阈值寄存器

寄存器位宽： [31:0]

偏移量： 0x24

复位值： 0x00000000

表 22-9 ADC 看门狗高阈值寄存器

位域	位域名称	访问	描述
31:12	Reserved		保留
11:0	HT	RW	模拟看门狗高阈值

22.4.9 ADC_LTR

中文名：ADC 看门狗低阈值寄存器

寄存器位宽：[31:0]

偏移量：0x28

复位值：0x00000000

表 22-10 ADC 看门狗低阈值寄存器

位域	位域名称	访问	描述
31:12	Reserved		保留
11:0	LT	RW	模拟看门狗低阈值

22.4.10 ADC_SQR1

中文名：ADC 规则序列寄存器 1

寄存器位宽：[31:0]

偏移量：0x2c

复位值：0x00000000

表 22-11 ADC 看门狗低阈值寄存器

位域	位域名称	访问	描述
31:24	Reserved		保留
23:20	L	RW	规则通道序列长度 0000: 1 个转换 0001: 2 个转换 1111: 16 个转换
19:15	SQ16	RW	注入序列第 16 个转换的输入源 0: PA13 1: PA12 2: PA11 3: PA10 4~7: Reserved 8: PA09 9: PA08 10: PA07 11: PA06
14:10	SQ15	RW	规则序列第 15 个转换的输入源
9:5	SQ14	RW	规则序列第 14 个转换的输入源
4:0	SQ13	RW	规则序列第 13 个转换的输入源

22.4.11 ADC_SQR2

中文名：ADC 规则序列寄存器 1

寄存器位宽：[31:0]

偏移量：0x30

复位值：0x00000000

表 22-12 ADC 看门狗低阈值寄存器

位域	位域名称	访问	描述
31:30	Reserved		保留
29:25	SQ12	RW	规则序列第 12 个转换的输入源
24:20	SQ11	RW	规则序列第 11 个转换的输入源
19:15	SQ10	RW	规则序列第 10 个转换的输入源
14:10	SQ9	RW	规则序列第 9 个转换的输入源
9:5	SQ8	RW	规则序列第 8 个转换的输入源
4:0	SQ7	RW	规则序列第 7 个转换的输入源

22.4.12 ADC_SQR3

中文名：ADC 规则序列寄存器 3

寄存器位宽：[31:0]

偏移量：0x34

复位值：0x00000000

表 22-13 ADC 看门狗低阈值寄存器

位域	位域名称	访问	描述
31:30	Reserved		保留
29:25	SQ6	RW	规则序列第 6 个转换的输入源
24:20	SQ5	RW	规则序列第 5 个转换的输入源
19:15	SQ4	RW	规则序列第 4 个转换的输入源
14:10	SQ3	RW	规则序列第 3 个转换的输入源
9:5	SQ2	RW	规则序列第 2 个转换的输入源
4:0	SQ1	RW	规则序列第 1 个转换的输入源

22.4.13 ADC_JSQR

中文名：ADC 注入序列寄存器

寄存器位宽：[31:0]

偏移量：0x38

复位值：0x00000000

表 22-14 ADC 注入序列寄存器

位域	位域名称	访问	描述
31:22	Reserved		保留
21:20	JL	RW	注入通道序列长度 00: 1 个转换 01: 2 个转换 10: 3 个转换 11: 4 个转换
19:15	JSQ4	RW	注入序列第 4 个转换的输入源
14:10	JSQ3	RW	注入序列第 3 个转换的输入源
9:5	JSQ2	RW	注入序列第 2 个转换的输入源
4:0	JSQ1	RW	注入序列第 1 个转换的输入源

22.4.14 ADC_JDRx (1-4)

中文名: ADC 注入数据寄存器

寄存器位宽: [31:0]

偏移量: 0x3c, 0x40, 0x44, 0x48

复位值: 0x00000000

表 22-15 ADC 注入数据寄存器

位域	位域名称	访问	描述
31:16	Reserved		保留
15:0	JDATA	RO	ADC 注入转换结果

22.4.15 ADC_DR

中文名: ADC 规则数据寄存器

寄存器位宽: [31:0]

偏移量: 0x4c

复位值: 0x00000000

表 22- 16 ADC 规则数据寄存器

位域	位域名称	访问	描述
31:16	Reserved		保留
15:0	DATA	RO	ADC 规则转换结果

23 功耗管理模块

23.1 概述

龙芯 2P0300 功耗管理模块提供系统功耗管理实现机制。

- 系统待机、休眠与唤醒，支持多种唤醒方式（网络，OTG 等）
- 支持 Dynamic Frequency Scaling（DFS），处理器核 DFS 控制，由片内打印 LA132 处理器核独立控制。
- 系统时钟控制，模块时钟门控，多种方式调节频率。

23.2 动态功耗管理

龙芯 2P0300 支持芯片内部各功能模块动态时钟门控管理功能，主要包括动态频率调节 (DFS) 和时钟门控管理技术。

其中，具体功耗管理功能划分如下表：

表 23-1 芯片各功能模块功耗管理功能划分

功能模块	DFS	动态时钟门控
NODE (LA264+SCACHE)	支持	支持
PRT-LA132	支持	支持
SCA-LA132	支持	支持
DDR	-	支持
IMAGE	-	支持
打印设备	-	支持
扫描设备	-	支持
主互联	-	支持
GMAC	-	支持
OTG	-	支持
USB	-	支持
主系统设备	-	支持

23.2.1 DFS 功能描述

2P0300 芯片支持主系统处理器核 (LA264+SCACHE)、打印处理器核 (LA132)、扫描处理器核 (LA132) 动态频率调节 (DFS) 控制，由片内打印核 LA132 处理器核控制实现，控制结构框图如下图所示：

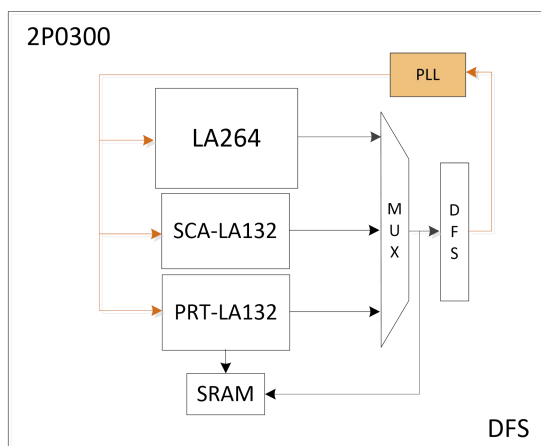


图23-1 处理器核 DFS 控制结构框图

芯片内部各处理器核动态调频均由打印核 LA132 控制实现，芯片内部配有 128KB 大小 SRAM 供打印核 LA132 启动运行，启动入口地址可由打印启动入口配置寄存器（地址：0x15103100）配置选择。

2P0300 各处理器核动态调频控制具体实现：

（1）LA264 处理器核**频率调节**，主要通过打印核 LA132 配置 LA264-DFS 控制寄存器（LA264_DFS_CTRL）实现，打印核 LA132 通过访问 LA264_DFS 控制寄存器，首先，配置 DFS 使能位有效，此时可通过配置 CLKBYP 位先将 LA264 处理器核时钟 BYPASS 为系统参考时钟（100MHz），再配置 DFS_CLK_ODIV、DFS_FREQSCALE 分频参数，对应处理器核时钟分频参数，进行时钟分频配置，待选择合适分频参数后，解除 CLKBYP 配置，处理器核恢复正常时钟运行。

（2）LA132 处理器核**频率调节**，主要通过打印核 LA132 配置 LA132-DFS 控制寄存器（LA132_DFS_CTRL）实现，打印核 LA132 通过访问 LA132_DFS 控制寄存器，首先，配置 DFS 使能位有效，此时可通过配置 CLKBYP 位先将各 LA132 处理器核时钟 BYPASS 为系统参考时钟（100MHz），再配置 DFS_CLK_ODIV、DFS_FREQSCALE 分频参数，对应处理器核时钟分频参数，进行时钟分频配置，待选择合适分频参数后，解除 CLKBYP 配置，处理器核恢复正常时钟运行。

2P0300 芯片主处理器核（LA264+SCACHE）、打印核、扫描核动态频率调节（DFS）具体操作流程如下图所示：

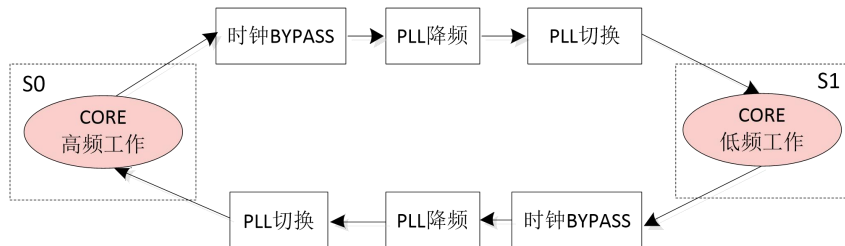


图23-2 处理器核 DFS 操作流程流程图

软件操作：

首先，2P0300 处理器核进入低压工作模式，首先需将处理器核进行降频处理，具体操作为：1) 使能 DFS，配置 dfs_en 有效；2) 将处理器核时钟 bypass 为系统参考时钟，配置 dfs_clkbyp 有效；3) 配置 dfs_clkodiv 时钟输出分频参数(最高位需为 1)；4) 切换 PLL 时钟，配置 dfs_clkbyp 无效；完成 2P0300 处理器核低频工作模式切换。

与此类似，2P0300 处理器核进入高频高性能工作模式，需对其工作频率进行升频处理，并进行相应时钟切换，完成 2P0300 处理器核高频模式切换。

23.2.2 设备唤醒功能描述

2P0300 芯片支持多种工作模式：正常工作、待机、休眠等模式，其中待机、休眠工作模式下均支持多种设备唤醒功能，包括 OTG 唤醒、网络唤醒、WIFI 唤醒及按键唤醒等。工作模式切换、各个设备唤醒操作均由打印系统统一控制维护。

其中，芯片待机唤醒操作控制流程，如下图所示：

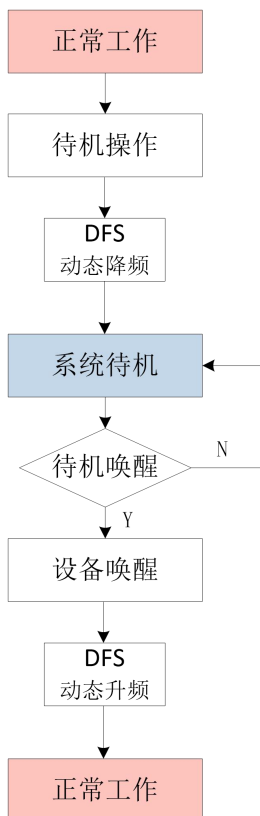


图23-3 芯片待机唤醒操作流程

芯片正常工作状态下，在没有打印等任务需求时，可进行待机操作控制，该操作任务由打印系统统一完成，主要包括：DFS 动态降频，将芯片内各处理器核进行动态降频操作，操作完成后芯片进入系统待机模式。系统待机模式下，若芯片设备需进行唤醒操作，主要包括 OTG 唤醒、网络唤醒、WIFI 及按键等唤醒，打印系统接收并维护各个设备唤醒事件，检测到以上设备有效唤醒事件后，将进行待机唤醒操作，主要包括：DFS 动态升频，将芯

片内各处理器核进行动态升频恢复操作，完成后芯片进入正常工作模式，可响应打印等任务需求。

其中，芯片休眠唤醒操作控制流程，如下图所示：

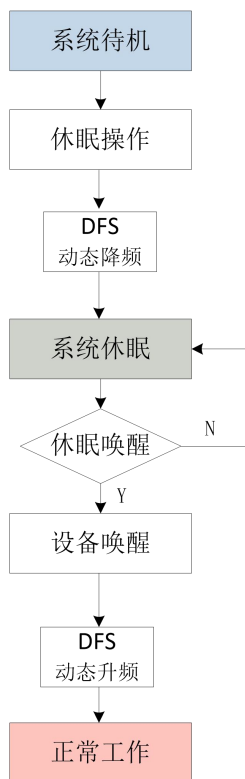


图23-4 芯片休眠唤醒操作流程

芯片系统待机状态下，一段时间内没有打印等任务需求时，可进一步进行休眠操作控制，该操作任务仍由打印系统统一完成，主要包括：DFS 动态降频，将芯片内各处理器核进一步进行动态降频操作，操作完成后芯片进入系统休眠模式。系统休眠模式下，若芯片设备需进行唤醒操作，主要包括 OTG 唤醒、网络唤醒、WIFI 及按键等唤醒，打印系统接收并维护各个设备唤醒事件，检测到以上设备有效唤醒事件后，将进行休眠唤醒操作，主要包括：DFS 动态升频，将芯片内各处理器核进行动态升频恢复操作，操作完成后芯片进入正常工作模式，可响应打印等任务需求。

23.3 寄存器描述

本节介绍功耗管理控制器相关寄存器。

动态频率管理(DFS)的物理基地址为：0x1510_3038；

寄存器电压域表示寄存器的该位所属电压域。

寄存器属性简写包括：

R/W（可读可写），RO（只读），

R/WC（可读，写清除），WO（只写，读无效）

23.3.1 DFS 寄存器描述

DFS 配置寄存器基址为：0x1510_3038。

23.3.1.1 LA264_DVFS_CTRL : LA264 DVFS Control Register

表 23-2 LA264 DVFS Control 寄存器描述

地址偏移		电压域	属性
0x0		SOC	R/W
位域	描述		电压域
31:20	保留		
19:16	DVFS_FREQSCALE - R/W NODE 模块 PLL 时钟输出 Freqscale 分频参数配置： 对应 PLL 时钟分频配置参数 freqscale。 freqscale [3]:NODE模块时钟分频模式freq_mode; freqscale [2:0]:NODE模块时钟输出分频系数：0~7		SOC
15:8	DVFS_CLK_ODIV - R/W NODE 模块 PLL 时钟输出分频参数配置，最高位(odiv[7])为软件参数配置使能位(高有效)： 对应 PLL 时钟配置参数 ODIV[6:0]：0~127		SOC
7:3	保留		-
2	DVFS_CLK_OFF - R/W NODE 模块时钟关断配置位，高电平时钟关断，默认值为 0。		SOC
1	DVFS_CLKBYP - R/W NODE 模块时钟 BYPASS 配置位，默认值为 0： 1: NODE 时钟 BYPASS 为系统参考时钟；0: NODE 时钟为 PLL 输出时钟。		SOC
0	DVFS_EN - R/W NODE 模块动态调频调压(DFS)使能位，高电平有效，默认值为 0。		SOC

注：分频模式分为两种分频配置：

- (1) freq_mode=0，设备时钟分频计算公式为： $f_{out}=f_{in}*(freqscale[2:0]+1)/8$ ；
- (2) freq_mode=1，设备时钟分频计算公式为： $f_{out}=f_{in}/(freqscale[2:0]+1)$ 。

23.3.1.2 LA132_DVFS_CTRL : LA132 DVFS Control Register

表 23-3 LA132 DVFS Control 寄存器描述

地址偏移		电压域	属性
0x4		SOC	R/W
位域	描述		电压域
31:24	保留		-
23:20	SCALA132_DVFS_FREQSCALE - R/W LA132(扫描核)模块 PLL 时钟输出 Freqscale 分频参数配置： 对应 PLL 时钟分频配置参数 freqscale。 scala132_freqscale [3]:扫描核模块时钟分频模式 freq_mode; scala132_freqscale [2:0]:扫描核模块时钟输出分频系数：0~7		SOC
19:16	PRTLA132_DVFS_FREQSCALE - R/W LA132(打印核)模块 PLL 时钟输出 Freqscale 分频参数配置： 对应 PLL 时钟分频配置参数 freqscale。 prtla132_freqscale [3]:打印核模块时钟分频模式 freq_mode; prtla132_freqscale [2:0]:打印核模块时钟输出分频系数：0~7		SOC
15:8	LA132_DVFS_CLK_ODIV - R/W LA132(打印、扫描核)模块 PLL 时钟输出分频参数配置，最高位(odiv[7])为软件参数配置使能位(高有效)： 对应 PLL 时钟配置参数 ODIV[6:0]：0~127		SOC

7	SCALA132_RSTN_EN - R/W LA132(扫描核)复位门控关断配置位，默认值为 0： 0-复位有效，1-复位解除。	SOC
6	SCALA132_CLK_EN - R/W LA132(扫描核)时钟门控关断配置位，默认值为 1： 0-时钟关闭，1-时钟开启。	SOC
5	PRTLA132_RSTN_EN - R/W LA132(打印核)复位门控关断配置位，默认值为 0： 0-复位有效，1-复位解除。	SOC
4	PRTLA132_CLK_EN - R/W LA132(打印核)时钟门控关断配置位，默认值为 1： 0-时钟关闭，1-时钟开启。	SOC
3	保留	-
2	LA132_DVFS_CLK_OFF - R/W LA132(打印、扫描核)时钟关断配置位，高电平时钟关断，默认值为 0。	SOC
1	LA132_DVFS_CLKBYP - R/W LA132(打印、扫描核)时钟 BYPASS 配置位，默认值为 0： 1: LA132 时钟 BYPASS 为系统参考时钟；0: LA132 时钟为 PLL 输出时钟。	SOC
0	LA132_DVFS_EN - R/W LA132(打印、扫描核)动态调频调压(DFS)使能位，高电平有效，默认值为 0。	SOC

23.3.2 WDT 寄存器描述

芯片内部看门狗(WDT)共三组，分别置于主系统、打印、扫描系统中，每组看门狗功能均一致，各个看门狗寄存器基址如下表：

表 23-4 WDT 寄存器地址

各系统 WDT 基地址	设备	备注
0x1420_5000	MAIN_WDT	每组 WDT 独立控制
0x1510_3900	PRT_WDT	每组 WDT 独立控制
0x1520_2900	SCA_WDT	每组 WDT 独立控制

23.3.2.1 WD_EN : Watch Dog Enable Register

表 23-5 WD_EN 寄存器描述

地址偏移	电压域	属性
0x00	SOC	R/W
位域	描述	
31:1	保留	
1	WD_EN - R/W Watch dog 功能使能，高电平有效	
0	OS_RST - R/W 系统软复位，该位软件写‘1’使系统复位	

23.3.2.2 WD_SET : Watch Dog Set Register

表 23-6 WD_SET 寄存器描述

地址偏移	电压域	属性
0x04	SOC	WO
位域	描述	
31:1	保留	
0	写该位将重填 watch dog 计数器，配置该位需首先配置 WDT 使能位。	

23.3.2.3WD_TimerL : Watch Dog Timer Low Register

表 23-7 WD_TIMERL 寄存器描述

地址偏移	电压域	属性
0x08	SOC	R/W
位域	描述	
31:0	该寄存器的值为 watch dog 重填值的低 32 位，复位后为全 1。	

23.3.2.4WD_TimerH : Watch Dog Timer High Register

表 23-8 WD_TIMER 寄存器描述

地址偏移	电压域	属性
0x0c	SOC	R/W
位域	描述	
31:0	该寄存器的值为 watch dog 重填值的高 32 位，复位后为全 1。	

23.3.2.5PLTRST_SET : Plantrst Set Register

表 23-9 PLTRST_SET 寄存器描述

地址偏移	电压域	属性
0x10	SOC	R/W
位域	描述	
31:1	保留	
0	Platform Reset 信号输出选择位（仅 PRT_WDT 存在该位域）： 0：内部看门狗、系统软复位不输出，仅内部复位有效； 1：选择将内部看门狗、系统软复位由 platform reset 引脚同步输出（默认配置，复位引脚输出）。	

24 GPIO

24.1 概述

龙芯 2P0300 共有 109 个 GPIO 引脚，全部与其他功能引脚复用。其中，主系统 GPIO 引脚资源共 48 个，打印系统 GPIO 引脚资源共 32 个，扫描系统 GPIO 引脚资源共 29 个。

24.2 寄存器描述

主系统、打印系统、扫描系统下 GPIO 寄存器基地址分别为：

表 24-1 GPIO 寄存器地址

各系统基地址	设备	备注
0x1420_1000	MAIN_GPIO0~47	每个 GPIO 引脚均由一组寄存器控制
0x1510_1000	PRT_GPIO0~31	每个 GPIO 引脚均由一组寄存器控制
0x1520_1000	SCA_GPIO0~28	每个 GPIO 引脚均由一组寄存器控制

每个系统下，GPIO 引脚均由一组寄存器控制，包括：GPIO 方向控制（GPIO_OEN）、GPIO 输出值（GPIO_O）、GPIO 输入值（GPIO_I）、GPIO 输入中断使能控制（GPIO_INT_EN）、GPIO 输入中断极性控制（GPIO_INT_POL）、GPIO 输入中断边沿性控制（GPIO_INT_EDGE）、GPIO 输入中断双沿控制（GPIO_INT_DUAL）、GPIO 输入中断清除（GPIO_INT_CLR）、GPIO 输入中断状态（GPIO_INT_STS）。

表 24-2 GPIO 控制寄存器

寄存器	大小（按位）	描述		
GPIO_OEN	1	GPIO 输出使能，低有效。		
GPIO_O	1	GPIO 输出值。		
GPIO_I	1	GPIO 输入值。		
GPIO_INT_EN	1	GPIO 中断使能。		
GPIO_INT_POL	1	GPIO 中断极性。		
GPIO_INT_EDGE	1	GPIO 中断边沿性。与 GPIO 中断极性配合控制 GPIO 中断状态的产生。		
		POL	EDGE	描述
		0	0	低电平触发中断
		1	0	高电平触发中断
		0	1	下降沿触发中断
		1	1	上升沿触发中断
GPIO_INT_CLR	1	GPIO 中断状态清除		
GPIO_INT_STS	1	GPIO 中断状态		
GPIO_INT_DUAL	1	GPIO 中断双沿控制，仅在边沿中断模式下有效		

24.3 访问地址

GPIO 的寄存器访问地址等于各系统下的基地址加寄存器偏移，各寄存器访问地址均需以字节形式访问。

芯片提供了两种方式来控制 GPIO 引脚：一种是按位控制每个 GPIO 引脚，一种是按字节控制每个 GPIO 引脚。芯片通过提供两个地址空间来映射 GPIO 控制寄存器实现该功能。一种是按位映射，一种是按字节来索引控制寄存器的每个比特位。对应的 GPIO 内部的地址空间也分为两部分。以上两种地址读写均需按照字节形式进行访问，推荐使用后一种方式来控制器 GPIO 引脚。

GPIO 模块的内部寄存器物理地址构成如下：

表 24-3 GPIO 模块内部寄存器物理地址

地址空间	说明
0x0 - 0x80	按位控制寄存器地址（需按字节形式访问，按位读写）
0x800-0xF80	按字节控制寄存器地址（需按字节形式访问，以字节为单位读写）

表 24-4 按位控制 GPIO 配置寄存器地址

地址偏移	寄存器	大小（位）	描述
0x00	GPIO_OEN	各系统区分	GPIO 输出使能，低有效。每位控制一个 GPIO 引脚。
0x10	GPIO_O	各系统区分	GPIO 输出值。每位控制一个 GPIO 引脚。
0x20	GPIO_I	各系统区分	GPIO 输入值。每位控制一个 GPIO 引脚。
0x30	GPIO_INT_EN	各系统区分	GPIO 中断使能。每位控制一个 GPIO 引脚。
0x40	GPIO_INT_POL	各系统区分	GPIO 中断极性。每位控制一个 GPIO 引脚。
0x50	GPIO_INT_EDGE	各系统区分	GPIO 中断边沿性。每位控制一个 GPIO 引脚。
0x60	GPIO_INT_CLR	各系统区分	GPIO 中断清除。每位控制一个 GPIO 引脚。
0x70	GPIO_INT_STS	各系统区分	GPIO 中断状态。每位控制一个 GPIO 引脚。
0x80	GPIO_INT_DUAL	各系统区分	GPIO 中断双沿模式。每位控制一个 GPIO 引脚。

表 24-5 按字节控制 GPIO 配置寄存器地址

地址偏移	寄存器	大小（字节）	描述
0x800	GPIO_OEN	各系统区分	GPIO 输出使能，低有效。每个字节控制一个 GPIO 引脚。
0x900	GPIO_O	各系统区分	GPIO 输出值。每个字节控制一个 GPIO 引脚。
0xA00	GPIO_I	各系统区分	GPIO 输入值。每个字节控制一个 GPIO 引脚。
0xB00	GPIO_INT_EN	各系统区分	GPIO 中断使能。每个字节控制一个 GPIO 引脚。
0xC00	GPIO_INT_POL	各系统区分	GPIO 中断极性。每个字节控制一个 GPIO 引脚。
0xD00	GPIO_INT_EDGE	各系统区分	GPIO 中断边沿性。每个字节控制一个 GPIO 引脚。
0xE00	GPIO_INT_CLR	各系统区分	GPIO 中断清除。每个字节控制一个 GPIO 引脚。
0xF00	GPIO_INT_STS	各系统区分	GPIO 中断状态。每个字节控制一个 GPIO 引脚。
0xF80	GPIO_INT_DUAL	各系统区分	GPIO 中断双沿模式。每个字节控制一个 GPIO 引脚。

24.4 控制寄存器

24.4.1 GPIO 方向控制

地址偏移：00-07h 属性：R/W

默认值：FFFFFFFFh 大小：8

表 24-6 GPIO 方向按位控制寄存器

位域	名称	访问	描述
63:0	GPIO_OEN	R/W	每位对应于 GPIO[63:0] (依各系统 GPIO 资源) 的方向控制。 0: 输出 1: 输入

地址偏移：800-83fh 属性：R/W

默认值：FFFFFFFFh 大小：64

表 24-7 GPIO 方向按字节控制寄存器

位域	名称	访问	描述
7:0	GPIO_OEN	R/W	每字节对应于每一位 GPIO[63:0] (依各系统 GPIO 资源) 的方向控制。 0: 输出 1: 输入

24.4.2 GPIO 输出值

地址偏移：10-17h 属性：R/W

默认值：00000000h 大小：8

表 24-8 GPIO 按位输出寄存器

位域	名称	访问	描述
63:0	GPIO_O	R/W	每位对应于 GPIO[63:0] (依各系统 GPIO 资源) 的输出值。

地址偏移：900-93fh 属性：R/W

默认值：00000000h 大小：64

表 24-9 GPIO 按字节输出寄存器

位域	名称	访问	描述
7:0	GPIO_O	R/W	每字节对应于每一位 GPIO[63:0] (依各系统 GPIO 资源) 的输出值。

24.4.3 GPIO 输入值

地址偏移：20-27h 属性：RO

默认值：N/A 大小：8

表 24-10 GPIO 按位输入寄存器

位域	名称	访问	描述
63:0	GPIO_I	RO	每位对应于 GPIO[63:0] (依各系统 GPIO 资源) 的输入值。

地址偏移：a00-a3fh 属性：R0
默认值：00000000h 大小：64

表 24-11 GPIO 按字节输入寄存器

位域	名称	访问	描述
7:0	GPIO_I	R0	每字节对应于每一位 GPIO[63:0]（依各系统 GPIO 资源）的输入值。

24.4.4 GPIO 中断使能

地址偏移：30-37h 属性：R/W
默认值：00000000h 大小：8

表 24-12 GPIO 按位中断使能寄存器

位域	名称	访问	描述
63:0	GPIO_INT_EN	R/W	每位对应于 GPIO[63:0]的输入中断使能。 0：关闭中断 1：使能中断

地址偏移：b00-b3fh 属性：R/W
默认值：00000000h 大小：64

表 24-13 GPIO 按字节中断使能寄存器

位域	名称	访问	描述
7:0	GPIO_INT_EN	R/W	每字节对应于每一位 GPIO[63:0]（依各系统 GPIO 资源）的输入中断使能。 0：关闭中断 1：使能中断

24.4.5 GPIO 中断极性

地址偏移：40-47h 属性：R/W
默认值：00000000h 大小：8

表 24-14 GPIO 按位中断极性寄存器

位域	名称	访问	描述
63:0	GPIO_INT_POL	R/W	每位对应于 GPIO[63:0]的中断极性。 与中断边沿性配合，组成四种中断触发方式，见下文 GPIO 中断边沿。

地址偏移：c00-c3fh 属性：R/W
默认值：00000000h 大小：64

表 24-15 GPIO 按字节中断极性寄存器

位域	名称	访问	描述
7:0	GPIO_INT_POL	R/W	每字节对应于每一位 GPIO[63:0]（依各系统 GPIO 资源）的中断极性。 与中断边沿性配合，组成四种中断触发方式，见下文 GPIO 中断边沿。

24.4.6 GPIO 中断边沿

地址偏移：50-57h 属性：R/W

默认值：00000000h 大小：8

表 24-16 GPIO 按位中断边沿寄存器

位域	名称	访问	描述		
63:0	GPIO_INT_EDGE	R/W	每位对应于 GPIO[63:0]的中断边沿。 与中断极性配合，组成四种中断触发方式。		
			POL	EDGE	描述
			0	0	低电平触发中断
			1	0	高电平触发中断
			0	1	下降沿触发中断
			1	1	上升沿触发中断

地址偏移：d00-d3fh 属性：R/W

默认值：00000000h 大小：64

表 24-17 GPIO 按字节中断边沿寄存器

位域	名称	访问	描述		
7:0	GPIO_INT_EDGE	R/W	每字节对应于每一位 GPIO[63:0] (依各系统 GPIO 资源)的中断边沿。 与中断极性配合，组成四种中断触发方式。		
			POL	EDGE	描述
			0	0	低电平触发中断
			1	0	高电平触发中断
			0	1	下降沿触发中断
			1	1	上升沿触发中断

24.4.7 GPIO 中断清除

地址偏移：60-67h 属性：R/W

默认值：00000000h 大小：8

表 24-18 GPIO 按位中断清除寄存器

位域	名称	访问	描述
63:0	GPIO_INT_CLR	R/W	对应于 GPIO[63:0]的中断清除。 写 1 清除相应 GPIO 位上的中断，随后硬件会自动置 0 中断清除寄存器相应位，不需软件再作处理。

地址偏移：e00-e3fh 属性：R/W

默认值：00000000h 大小：64

表 24-19 GPIO 按字节中断清除寄存器

位域	名称	访问	描述
7:0	GPIO_INT_CLR	R/W	每字节对应于每一位 GPIO[63:0] (依各系统 GPIO 资源) 的中断极性。 写 1 清除相应 GPIO 位上的中断，随后硬件会自动置 0 中断清除寄存器相应位，不需软件再作处理。

24.4.8 GPIO 中断状态

地址偏移：70-77h 属性：R/W

默认值：00000000h 大小：8

表 24-20 GPIO 按位中断状态寄存器

位域	名称	访问	描述
63:0	GPIO_INT_STS	R/W	对应于 GPIO[63:0] 的中断状态。 1：有中断 0：无中断

地址偏移：f00-f3fh 属性：R/W

默认值：00000000h 大小：64

表 24-21 GPIO 按字节中断状态寄存器

位域	名称	访问	描述
7:0	GPIO_INT_STS	R/W	每字节对应于每一位 GPIO[63:0] (依各系统 GPIO 资源) 的中断状态。 1：有中断 0：无中断

24.4.9 GPIO 中断双沿模式

地址偏移：80-87h 属性：R/W

默认值：00000000h 大小：8

表 24-22 GPIO 按位中断双沿寄存器

位域	名称	访问	描述
63:0	GPIO_INT_DUAL	R/W	每位对应于 GPIO[63:0] 的中断双沿模式。 1：双沿中断触发 0：单沿中断触发

地址偏移：f80-fbfh 属性：R/W

默认值：00000000h 大小：64

表 24-23 GPIO 按字节中断双沿寄存器

位域	名称	访问	描述
7:0	GPIO_INT_DUAL	R/W	每字节对应于每一位 GPIO[63:0] (依各系统 GPIO 资源) 的中断双沿模式。 1：双沿中断触发 0：单沿中断触发

修订记录

版本号	更新内容
V1.0	发布版本

技术支持

可通过邮箱向我司提交芯片手册和产品使用的问题，并获取技术支持。

服务邮箱：service@loongson.cn

声明

本文档版权归龙芯中科技术股份有限公司所有，未经许可不得擅自实施传播等侵害版权人合法权益的行为。

本文档仅提供阶段性信息，可根据实际情况进行更新，恕不另行通知。如因文档使用不当造成的直接或间接损失，本公司不承担任何责任。

龙芯中科技术股份有限公司

Loongson Technology Corporation Limited

地址：北京市海淀区中关村环保科技示范园龙芯产业园 2 号楼

Building No.2, Loongson Industrial Park,

Zhongguancun Environmental Protection Park, Haidian District, Beijing

电话(Tel): 010-62546668

传真(Fax): 010-62600826