

LOONGSON

龙芯 2K0300 处理器

用户手册

V1.01

2025 年05 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 2K0300 处理器用户手册》主要介绍龙芯 2K0300 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

目 录	I
图目录	IX
表目录	X
1 概述	1
1.1 体系结构框图	2
1.2 芯片主要功能	2
1.2.1 处理器核	2
1.2.2 内存接口	2
1.2.3 显示接口	3
1.2.4 USB 控制器	3
1.2.5 GMAC 控制器	3
1.2.6 I2S	3
1.2.7 SPI	3
1.2.8 UART	4
1.2.9 I2C	4
1.2.10 AD	4
1.2.11 CAN	4
1.2.12 TIMER	4
1.2.13 PWM	4
1.2.14 SDIO	5
1.2.15 LocalIO	5
1.2.16 HPET	5
1.2.17 RTC	5
1.2.18 GPIO	5
1.2.19 Watchdog	5
1.2.20 温度传感器	5
1.2.21 中断控制器	5
1.3 芯片引脚复用关系	6
2 时钟结构	10
2.1 NODE PLL	10
2.2 DDR PLL	11
2.3 PIX PLL	12
2.4 内部 PLL 配置方法	12
2.4.1 硬件配置	12
2.4.2 软件配置	13
2.5 USB 参考时钟	13
2.6 时钟信号说明	13
3 芯片配置与控制	15
3.1 芯片初始化信号	15
3.2 地址空间分配	16
3.3 时钟与复位控制	19
3.3.1 时钟配置概要	19
3.3.2 展频 PLL 配置	19
3.3.3 复位控制	20
3.4 芯片配置寄存器	21
3.4.1 芯片通用配置寄存器 00	24
3.4.2 芯片通用配置寄存器 01	25
3.4.3 芯片通用配置寄存器 02	26
3.4.4 芯片通用配置寄存器 03	26



3.4.5 芯片通用配置寄存器 04	28
3.4.6 芯片通用配置寄存器 05	29
3.4.7 芯片通用配置寄存器 06	31
3.4.8 芯片通用配置寄存器 07	32
3.4.9 芯片通用配置寄存器 08	33
3.4.10 芯片通用配置寄存器 09	34
3.4.11 芯片通用配置寄存器 10	34
3.4.12 芯片通用配置寄存器 11	34
3.4.13 芯片通用配置寄存器 12	34
3.4.14 芯片通用配置寄存器 13	34
3.4.15 芯片通用配置寄存器 14	36
3.4.16 芯片通用配置寄存器 15	37
3.4.17 芯片采样参数寄存器 0	38
3.4.18 芯片采样参数寄存器 1	38
3.4.19 芯片采样参数寄存器 2	39
3.4.20 芯片采样参数寄存器 3	39
3.4.21 芯片计数寄存器 0	39
3.4.22 芯片计数寄存器 1	39
3.4.23 NODE PLL 时钟配置寄存器 0	39
3.4.24 NODE PLL 时钟配置寄存器 1	40
3.4.25 DDR PLL 时钟配置寄存器 0	40
3.4.26 DDR PLL 时钟配置寄存器 1	41
3.4.27 PIX PLL 时钟配置寄存器 0	41
3.4.28 PIX PLL 时钟配置寄存器 1	42
3.4.29 设备时钟分频配置寄存器	42
3.4.30 设备时钟输出使能配置寄存器	43
3.4.31 GPIO 复用配置寄存器 0	44
3.4.32 GPIO 复用配置寄存器 1	45
3.4.33 GPIO 复用配置寄存器 2	46
3.4.34 GPIO 复用配置寄存器 3	47
3.4.35 GPIO 复用配置寄存器 4	48
3.4.36 GPIO 复用配置寄存器 5	49
3.4.37 GPIO 复用配置寄存器 6	50
3.4.38 USB PHY 配置寄存器 0	51
3.4.39 USB PHY 配置寄存器 1	51
3.4.40 USB PHY 配置寄存器 2	51
3.5 中断配置及路由	53
3.5.1 中断触发类型	56
3.5.2 中断相关寄存器描述	56
3.5.3 中断路由寄存器描述	61
4 DDR4 控制器	64
4.1 访问地址	64
4.2 DDR4 控制器寄存器	64
5 GMAC 控制器	75
5.1 寄存器描述	75
5.2 软件编程向导:	75
6 USB 控制器	77
6.1 总体概述	77
6.2 控制器寄存器	77
7 OTG 控制器	78
7.1 概述	78



7.2 寄存器描述	78
8 显示控制器	79
8.1 概述	79
8.2 寄存器访问地址和引脚说明	79
9 SPI-FLASH 控制器	80
9.1 SPI 控制器结构	80
9.2 配置寄存器	81
9.2.1 控制寄存器 (SPCR)	81
9.2.2 状态寄存器 (SPSR)	81
9.2.3 数据寄存器 (TxFIFO/RxFIFO)	81
9.2.4 外部寄存器 (SPER)	82
9.2.5 参数控制寄存器 (SFC_PARAM)	82
9.2.6 片选控制寄存器 (SFC_SOFTCS)	82
9.2.7 时序控制寄存器 (SFC_TIMING)	82
9.3 接口时序	83
9.3.1 SPI 主控制器接口时序	83
9.3.2 SPI Flash 访问时序	83
9.4 使用指南	84
9.4.1 SPI 主控制器的读写操作	84
9.4.2 硬件 SPI Flash 读	85
9.4.3 混合访问 SPI Flash 和 SPI 主控制器	85
10 SPI-I/O 控制器	86
10.1 SPI 控制器结构	86
10.2 配置寄存器	86
10.2.1 控制寄存器 1 (CR1)	86
10.2.2 控制寄存器 2 (CR2)	87
10.2.3 控制寄存器 3 (CR3)	87
10.2.4 控制寄存器 4 (CR4)	88
10.2.5 中断寄存器 (IER)	88
10.2.6 状态寄存器 1 (SR1)	88
10.2.7 状态寄存器 2 (SR2)	89
10.2.8 配置寄存器 1 (CFG1)	90
10.2.9 配置寄存器 2 (CFG2)	90
10.2.10 配置寄存器 3 (CFG3)	90
10.2.11 CRC 寄存器 1 (CRC1)	91
10.2.12 CRC 寄存器 2 (CRC2)	91
10.2.13 数据寄存器 (DR)	92
10.3 功能描述	92
10.3.1 PIN 脚配置	92
10.3.2 SS 模式	93
10.3.3 波特率	93
10.3.4 传输格式	94
10.3.5 FIFO 阈值	94
10.3.6 FIFO 中数据储存格式和访问格式	94
10.3.7 单次传输帧数	95
10.3.8 开始与挂起	95
11 I2S 控制器	96
11.1 I2S 控制器概述	96
11.2 I2S 配置寄存器	96
12 I2C 控制器	99
12.1 概述	99



12.2 主要特性	99
12.3 功能描述	100
12.3.1 模式选择	100
12.3.2 从模式	100
12.3.2.1 从发送模式	100
12.3.2.2 从接收模式	101
12.3.3 主模式	102
12.3.3.1 主模式时钟生成	102
12.3.3.2 开始条件	102
12.3.3.3 从设备地址发送	102
12.3.3.4 主发送模式	103
12.3.3.5 主接收模式	103
12.3.4 错误条件	105
12.3.5 DMA 请求	106
12.4 中断	107
12.5 寄存器定义	107
12.5.1 I2C 控制寄存器 (I2C_CR1)	107
12.5.2 I2C 控制寄存器 2 (I2C_CR2)	109
12.5.3 I2C 从地址寄存器 (I2C_OAR)	109
12.5.4 I2C 数据寄存器 (I2C_DR)	109
12.5.5 I2C 状态寄存器 (I2C_SR1)	110
12.5.6 I2C 状态寄存器 2 (I2C_SR2)	111
12.5.7 I2C 时钟控制寄存器 (I2C_CCR)	112
12.5.8 I2C 上升时间寄存器 (I2C_TRISE)	112
13 LocalIO 控制器	114
13.1 访问地址及引脚复用	114
13.2 LOCALIO 控制器功能概述	114
14 UART 控制器	116
14.1 概述	116
14.2 控制器结构	116
14.3 寄存器描述	117
14.3.1 数据寄存器 (DAT)	117
14.3.2 中断使能寄存器 (IER)	118
14.3.3 中断标识寄存器 (IIR)	118
14.3.4 FIFO 控制寄存器 (FCR)	119
14.3.5 线路控制寄存器 (LCR)	119
14.3.6 MODEM 控制寄存器 (MCR)	120
14.3.7 线路状态寄存器 (LSR)	121
14.3.8 MODEM 状态寄存器 (MSR)	121
14.3.9 分频锁存器	122
15 SDIO 控制器	124
15.1 功能概述	124
15.2 访问地址及引脚复用	124
15.3 寄存器描述	124
15.4 软件编程指南	132
15.4.1 SD Memory 卡软件编程说明	132
15.4.2 SDIO 卡软件编程说明	133
15.4.3 DDR 模式设置	134
15.5 支持 SDIO 型号	134
16 eMMC 控制器	135
16.1 功能概述	135



16.2 访问地址及引脚复用	135
16.3 寄存器描述	135
16.4 专用 DMA 控制器	143
16.4.1 结构描述	143
16.4.2 DMA 描述符	144
16.5 软件配置流程	147
16.5.1 eMMC 正常读写流程	147
16.5.2 eMMC 初始化流程	148
16.5.3 DDR 模式设置	148
17 CANFD 控制器	149
17.1 概述	149
17.2 CANFD 控制器特性	149
18 ATIM 控制器	150
18.1 概述	150
18.2 功能描述	150
18.2.1 计数模式	150
18.2.2 输入模式	151
18.2.2.1 编码器接口模式	151
18.2.2.2 霍尔传感器模式	152
18.2.3 输出模式	152
18.2.3.1 电平输出模式	152
18.2.3.2 PWM 模式	152
18.2.3.3 互补输出和死区插入	153
18.2.3.4 刹车输入	153
18.2.4 定时器外部控制	153
18.3 寄存器描述	153
18.3.1 寄存器地址列表	153
18.3.2 ATIM_CR1	154
18.3.3 ATIM_CR2	155
18.3.4 ATIM_SMCR	157
18.3.5 ATIM_DIER	158
18.3.6 ATIM_SR	160
18.3.7 ATIM_EGR	161
18.3.8 ATIM_CCMR1 (OUT)	162
18.3.9 ATIM_CCMR1 (IN)	163
18.3.10 ATIM_CCMR2 (OUT)	165
18.3.11 ATIM_CCMR2 (IN)	166
18.3.12 ATIM_CCER	167
18.3.13 ATIM_CNT	168
18.3.14 ATIM_PSC	169
18.3.15 ATIM_ARR	169
18.3.16 ATIM_RCR	169
18.3.17 ATIM_CCR1	170
18.3.18 ATIM_CCR2	170
18.3.19 ATIM_CCR3	170
18.3.20 ATIM_CCR4	171
18.3.21 ATIM_BDTR	171
18.3.22 ATIM_INSTA	172
19 GTIM 控制器	174
19.1 概述	174
19.2 功能描述	174



19.2.1 计数模式	174
19.2.2 输入模式	175
19.2.2.1 编码器接口模式	175
19.2.2.2 霍尔传感器模式	176
19.2.3 输出模式	176
19.2.3.1 电平输出模式	176
19.2.3.2 PWM 模式	176
19.2.4 定时器外部控制	177
19.3 寄存器描述	177
19.3.1 寄存器地址列表	177
19.3.2 GTIM_CR1	177
19.3.3 GTIM_CR2	179
19.3.4 GTIM_SMCR	179
19.3.5 GTIM_DIER	181
19.3.6 GTIM_SR	182
19.3.7 GTIM_EGR	183
19.3.8 GTIM_CCMR1 (OUT)	184
19.3.9 GTIM_CCMR1 (IN)	186
19.3.10 GTIM_CCMR2 (OUT)	187
19.3.11 GTIM_CCMR2 (IN)	188
19.3.12 GTIM_CCER	190
19.3.13 GTIM_CNT	190
19.3.14 GTIM_PSC	191
19.3.15 GTIM_ARR	191
19.3.16 GTIM_CCR1	191
19.3.17 GTIM_CCR2	192
19.3.18 GTIM_CCR3	192
19.3.19 GTIM_CCR4	192
19.3.20 GTIM_INSTA	193
20 BTIM 控制器	194
20.1 概述	194
20.2 功能描述	194
20.3 寄存器描述	194
20.3.1 寄存器地址列表	194
20.3.2 BTIM_CR1	195
20.3.3 BTIM_CR2	195
20.3.4 BTIM_DIER	196
20.3.5 BTIM_SR	196
20.3.6 BTIM_EGR	197
20.3.7 BTIM_CNT	197
20.3.8 BTIM_PSC	197
20.3.9 BTIM_ARR	198
21 PWM 控制器	199
21.1 概述	199
21.2 访问地址及引脚复用	199
21.3 寄存器描述	199
21.4 功能说明	200
21.4.1 脉宽调制功能	200
21.4.2 脉冲测量功能	201
21.4.3 防死区功能	201
22 ADC 控制器	203



22.1 概述	203
22.2 工作模式	203
22.3 其他功能	203
22.3.1 注入通道管理	203
22.3.2 数据对齐	204
22.3.3 可编程的通道采样时间	204
22.3.4 模拟看门狗	204
22.3.5 DMA 请求	204
22.3.6 外部触发	205
22.4 寄存器描述	205
22.4.1 寄存器地址列表	205
22.4.2 ADC_SR	205
22.4.3 ADC_CR1	206
22.4.4 ADC_CR2	208
22.4.5 ADC_SMPR1	210
22.4.6 ADC_SMPR2	211
22.4.7 ADC_JOFRx (1-4)	211
22.4.8 ADC_HTR	212
22.4.9 ADC_LTR	212
22.4.10 ADC_SQR1	212
22.4.11 ADC_SQR2	213
22.4.12 ADC_SQR3	213
22.4.13 ADC_JSQR	214
22.4.14 ADC_JDRx (1-4)	214
22.4.15 ADC_DR	215
23 DMA 控制器	216
23.1 概述	216
23.2 寄存器定义	216
23.2.1 DMA 中断状态寄存器 (DMA_ISR)	217
23.2.2 DMA 中断标志清除寄存器 (DMA_IFCR)	218
23.2.3 DMA 通道 x 配置寄存器 (DMA_CCRx)	218
23.2.4 DMA 通道 x 传输数量寄存器 (DMA_CNDTRx)	219
23.2.5 DMA 通道 x 外设地址寄存器 (DMA_CPARx)	219
23.2.6 DMA 通道 x 存储器地址寄存器 (DMA_CMARx)	220
23.3 功能描述	220
23.3.1 配置流程	220
23.3.2 宽度和对齐方式	220
23.3.3 通道映射	220
24 HPET 控制器	222
24.1 概述	222
24.2 访问地址	222
24.3 寄存器描述	222
24.3.1 General Capabilities and ID Register	223
24.3.2 General Configuration Register	223
24.3.3 General Interrupt Status Register	224
24.3.4 Main Counter Value Register	224
24.3.5 Timer N Configuration and Capabilities Register	224
24.3.6 Timer N Comparator Value Register	225
25 RTC	227
25.1 概述	227
25.2 寄存器描述	227



25.2.1 寄存器地址列表	227
25.2.2 SYS_TOYWRITE0	228
25.2.3 SYS_TOYWRITE1	228
25.2.4 SYS_TOYREAD0	228
25.2.5 SYS_TOYREAD1	229
25.2.6 SYS_TOYMATCH0/1/2	229
25.2.7 SYS_RTCCTRL	229
25.2.8 SYS_RTCWRITE	230
25.2.9 SYS_RTCREAD	230
25.2.10 SYS_RTCMATCH0/1/2	231
26 功耗管理模块	232
26.1 概述	232
26.2 动态功耗管理	232
26.2.1 DFS 功能描述	232
26.3 寄存器描述	233
26.3.1 DFS 寄存器描述	233
26.3.2 WDT 寄存器描述	233
27 GPIO	235
27.1 概述	235
27.2 寄存器描述	235
27.3 访问地址	236
27.4 控制寄存器	237
27.4.1 GPIO 方向控制	237
27.4.2 GPIO 输出值	237
27.4.3 GPIO 输入值	238
27.4.4 GPIO 中断使能	238
27.4.5 GPIO 中断极性	239
27.4.6 GPIO 中断边沿	239
27.4.7 GPIO 中断清除	240
27.4.8 GPIO 中断状态	240
27.4.9 GPIO 中断双沿模式	241
修订记录	242



图目录

图1- 1 龙芯 2K0300 芯片结构图	2
图2- 1 时钟结构	10
图2- 2 NODE PLL 结构图	11
图2- 3 DDR PLL 时钟结构	12
图2- 4 PIX PLL 时钟结构	12
图3- 1 展频 PLL 概念性结构	19
图3- 2 龙芯 2K0300 传统中断路由示意图	53
图3- 3 龙芯 2K0300 扩展 IO 中断路由示意图	55
图9- 1 SPI 控制器结构	80
图9- 2 SPI 主控制器接口时序	83
图9- 3 SPI Flash 标准读时序	83
图9- 4 SPI Flash 快速读时序	84
图9- 5 SPI Flash 双向 I/O 读时序	84
图12- 1 从发送模式示意图	101
图12- 2 从接收模式示意图	101
图12- 3 主发送模式示意图	103
图12- 4 主接收模式（及时）示意图	104
图12- 5 主接收模式（非及时、3 字节）示意图	105
图12- 6 主接收模式（非及时、2 字节）示意图	105
图13- 1 LocalIO 读时序	114
图13- 2 LocalIO 写时序	115
图14- 1 UART 控制器结构	116
图16- 1 eMMC 初始化流程图	148
图21- 1 防死区功能	202
图22- 1 数据对齐方式	204
图26- 1 处理器核 DFS 操作流程	232



表目录

表 3- 1 初始化配置信号	15
表 3- 2 地址空间分配之 LA264 视角	16
表 3- 3 地址空间分配之 DMA 视角	18
表 3- 4 PLL 相关配置信号说明表	20
表 3- 5 系统芯片配置寄存器列表	21
表 3- 6 芯片通用配置寄存器 00	24
表 3- 7 芯片通用配置寄存器 01	25
表 3- 8 芯片通用配置寄存器 02	26
表 3- 9 芯片通用配置寄存器 03	26
表 3- 10 芯片通用配置寄存器 04	28
表 3- 11 芯片通用配置寄存器 05	29
表 3- 12 芯片通用配置寄存器 06	31
表 3- 13 芯片通用配置寄存器 07	32
表 3- 14 芯片通用配置寄存器 08	33
表 3- 15 芯片通用配置寄存器 09	34
表 3- 16 芯片通用配置寄存器 10	34
表 3- 17 芯片通用配置寄存器 11	34
表 3- 18 芯片通用配置寄存器 12	34
表 3- 19 芯片通用配置寄存器 13	35
表 3- 20 芯片通用配置寄存器 14	36
表 3- 21 芯片通用配置寄存器 15	37
表 3- 22 芯片采样参数寄存器 0	38
表 3- 23 芯片采样参数寄存器 1	38
表 3- 24 芯片采样参数寄存器 2	39
表 3- 25 芯片采样参数寄存器 3	39
表 3- 26 芯片计数寄存器 0	39
表 3- 27 芯片计数寄存器 1	39
表 3- 28 NODE PLL 时钟配置寄存器 0	39
表 3- 29 NODE PLL 时钟配置寄存器 1	40
表 3- 30 DDR PLL 时钟配置寄存器 0	40
表 3- 31 DDR PLL 时钟配置寄存器 1	41
表 3- 32 PIX PLL 时钟配置寄存器 0	41
表 3- 33 PIX PLL 时钟配置寄存器 1	42



表 3- 34	设备时钟分频配置寄存器	42
表 3- 35	设备时钟输出使能配置寄存器	43
表 3- 36	GPIO 复用配置寄存器 0	44
表 3- 37	GPIO 复用配置寄存器 1	45
表 3- 38	GPIO 复用配置寄存器 2	46
表 3- 39	GPIO 复用配置寄存器 3	47
表 3- 40	GPIO 复用配置寄存器 4	48
表 3- 41	GPIO 复用配置寄存器 5	49
表 3- 42	GPIO 复用配置寄存器 6	50
表 3- 43	USB PHY 配置寄存器 2	51
表 3- 44	I0 传统中断寄存器列表	54
表 3- 45	系统 I0 扩展中断寄存器列表	56
表 3- 46	传统中断控制寄存器属性	56
表 3- 47	传统中断控制寄存器地址	58
表 3- 48	扩展中断控制寄存器属性	59
表 3- 49	传统中断路由寄存器的说明	61
表 3- 50	传统中断路由寄存器地址	61
表 3- 51	扩展中断路由寄存器的说明	62
表 4- 1	内存控制器地址空间分配	64
表 4- 2	DDR4 SDRAM 配置参数寄存器	65
表 6- 1	USB 控制器地址空间分布	77
表 7- 1	OTG 控制器地址空间分布	78
表 9- 1	SPI 控制器地址空间分布	80
表 9- 2	SPI0 配置寄存器列表	81
表 9- 3	SPI 控制寄存器 (SPCR)	81
表 9- 4	SPI 状态寄存器 (SPSR)	81
表 9- 5	SPI 数据寄存器 (TxFIFO/RxFIFO)	81
表 9- 6	SPI 外部寄存器 (SPER)	82
表 9- 7	SPI 分频系数	82
表 9- 8	SPI 参数控制寄存器 (SFC_PARAM)	82
表 9- 9	SPI 片选控制寄存器 (SFC_SOFTCS)	82
表 9- 10	SPI 时序控制寄存器 (SFC_TIMING)	83
表 10- 1	SPI-I0 控制器地址空间分布	86
表 10- 2	SPI-I0 寄存器列表	86



表 10- 3	SPI 控制寄存器 1 (CR1)	86
表 10- 4	SPI 控制寄存器 2 (CR2)	87
表 10- 5	SPI 控制寄存器 3 (CR3)	87
表 10- 6	SPI 控制寄存器 4 (CR4)	88
表 10- 7	SPI 中断寄存器 (IER)	88
表 10- 8	SPI 状态寄存器 1 (SR1)	88
表 10- 9	SPI 状态寄存器 2 (SR2)	89
表 10- 10	SPI 配置寄存器 1 (CFG1)	90
表 10- 11	SPI 配置寄存器 2 (CFG2)	90
表 10- 12	SPI 配置寄存器 3 (CFG3)	90
表 10- 13	SPI CRC 寄存器 1 (CRC1)	91
表 10- 14	SPI CRC 寄存器 2 (CRC2)	91
表 10- 15	SPI 数据寄存器 (DR)	92
表 10- 16	PIN 脚配置位和流动方向关系	92
表 10- 17	PIN 脚常见配置	93
表 10- 18	PIN 脚配置位和流动方向关系	93
表 10- 19	FIFO 访问格式	95
表 11- 1	I2S 控制器地址空间分布	96
表 11- 2	I2S 控制器地址空间分布	96
表 11- 3	I2S 标识寄存器	96
表 11- 4	I2S 配置寄存器 0	97
表 11- 5	I2S 控制寄存器	97
表 11- 6	I2S 配置寄存器 1	98
表 12- 1	I2C 控制器地址空间分布	99
表 12- 2	I2C 中断请求	107
表 12- 3	I2C 控制器寄存器列表	107
表 12- 4	I2C 控制寄存器	107
表 12- 5	I2C 控制寄存器 2	109
表 12- 6	I2C 从地址寄存器	109
表 12- 7	I2C 数据寄存器	110
表 12- 8	I2C 状态寄存器	110
表 12- 9	I2C 状态寄存器 2	111
表 12- 10	I2C 时钟控制寄存器	112
表 12- 11	I2C 上升时间寄存器	112



表 13- 1	LocalIO 地址空间分布	114
表 14- 1	UART0~9 控制器地址空间分布	117
表 14- 2	UART 配置寄存器列表	117
表 14- 3	数据传输寄存器	118
表 14- 4	中断使能寄存器	118
表 14- 5	中断源寄存器	118
表 14- 6	中断控制功能表	119
表 14- 7	FIFO 控制寄存器	119
表 14- 8	线路控制寄存器	119
表 14- 9	Modem 控制寄存器	120
表 14- 10	线路状态寄存器	121
表 14- 11	Modem 状态寄存器	122
表 14- 12	分频锁存器低 8 位寄存器	122
表 14- 13	分频锁存器高 8 位寄存器	122
表 14- 14	分频锁存器小数位寄存器	122
表 15- 1	SDIO 内部寄存器物理地址构成	124
表 15- 2	SDI_CON 寄存器	124
表 15- 3	SDI_CON 寄存器位域描述	124
表 15- 4	SDI_PRE 寄存器	125
表 15- 5	SDI_PRE 寄存器位域描述	125
表 15- 6	SDI_CMD_ARG 寄存器	125
表 15- 7	SDI_CMD_ARG 寄存器位域描述	125
表 15- 8	SDI_CMD_CON 寄存器	125
表 15- 9	SDI_CMD_CON 寄存器位域描述	125
表 15- 10	SDI_CMD_STA 寄存器	126
表 15- 11	SDI_CMD_STA 寄存器位域描述	126
表 15- 12	SDI_RSP0 寄存器	126
表 15- 13	SDI_RSP0 寄存器位域描述	126
表 15- 14	SDI_RSP1 寄存器	126
表 15- 15	SDI_RSP1 寄存器位域描述	126
表 15- 16	SDI_RSP2 寄存器	127
表 15- 17	SDI_RSP2 寄存器位域描述	127
表 15- 18	SDI_RSP3 寄存器	127
表 15- 19	SDI_RSP3 寄存器位域描述	127



表 15- 20	SDI_DTIMER 寄存器	127
表 15- 21	SDI_DTIMER 寄存器位域描述	127
表 15- 22	SDI_BSIZE 寄存器	127
表 15- 23	SDI_BSIZE 寄存器位域描述	127
表 15- 24	SDI_DAT_CON 寄存器	127
表 15- 25	SDI_DAT_CON 寄存器位域描述	127
表 15- 26	SDI_DAT_CNT 寄存器	128
表 15- 27	SDI_DAT_CNT 寄存器位域描述	128
表 15- 28	SDI_DAT_STA 寄存器	128
表 15- 29	SDI_DAT_STA 寄存器位域描述	128
表 15- 30	SDI_FIFO_STA 寄存器	129
表 15- 31	SDI_FIFO_STA 寄存器位域描述	129
表 15- 32	SDI_INT_MASK 寄存器	129
表 15- 33	SDI_INT_MASK 寄存器位域描述	130
表 15- 34	SDI_DAT 寄存器	130
表 15- 35	SDI_DAT 寄存器位域描述	130
表 15- 36	SDI_INT_EN 寄存器	130
表 15- 37	SDI_INT_EN 寄存器位域描述	130
表 15- 38	dll_master_val 寄存器	131
表 15- 39	dll_master_val 寄存器位域描述	131
表 15- 40	dll_con 寄存器	131
表 15- 41	dll_con 寄存器位域描述	131
表 15- 42	param_delay 寄存器	131
表 15- 43	param_delay 寄存器位域描述	131
表 15- 44	sdio_emmc_sel 寄存器	132
表 15- 45	sdio_emmc_sel 寄存器位域描述	132
表 16- 1	eMMC 内部寄存器物理地址构成	135
表 16- 2	EMMC_CON 寄存器	135
表 16- 3	EMMC_CON 寄存器位域描述	135
表 16- 4	EMMC_PRE 寄存器	135
表 16- 5	EMMC_PRE 寄存器位域描述	136
表 16- 6	EMMC_CMD_ARG 寄存器	136
表 16- 7	EMMC_CMD_ARG 寄存器位域描述	136
表 16- 8	EMMC_CMD_CON 寄存器	136



表 16- 9	EMMC_CMD_CON 寄存器位域描述	136
表 16- 10	EMMC_CMD_STA 寄存器	137
表 16- 11	EMMC_CMD_STA 寄存器位域描述	137
表 16- 12	EMMC_RSP0 寄存器	137
表 16- 13	EMMC_RSP0 寄存器位域描述	137
表 16- 14	EMMC_RSP1 寄存器	137
表 16- 15	EMMC_RESP1 寄存器位域描述	137
表 16- 16	EMMC_RSP2 寄存器	138
表 16- 17	EMMC_RSP2 寄存器位域描述	138
表 16- 18	EMMC_RSP3 寄存器	138
表 16- 19	EMMC_RSP3 寄存器位域描述	138
表 16- 20	EMMC_DTIMER 寄存器	138
表 16- 21	EMMC_DTIMER 寄存器位域描述	138
表 16- 22	EMMC_BSIZE 寄存器	138
表 16- 23	EMMC_BSIZE 寄存器位域描述	138
表 16- 24	EMMC_DAT_CON 寄存器	138
表 16- 25	EMMC_DAT_CON 寄存器位域描述	139
表 16- 26	EMMC_DAT_CNT 寄存器	139
表 16- 27	EMMC_DAT_CNT 寄存器位域描述	139
表 16- 28	EMMC_DAT_STA 寄存器	139
表 16- 29	EMMC_DAT_STA 寄存器位域描述	139
表 16- 30	EMMC_FIFO_STA 寄存器	140
表 16- 31	EMMC_FIFO_STA 寄存器位域描述	140
表 16- 32	EMMC_INT_MASK 寄存器	141
表 16- 33	EMMC_INT_MASK 寄存器位域描述	141
表 16- 34	EMMC_DAT 寄存器	141
表 16- 35	EMMC_DAT 寄存器位域描述	141
表 16- 36	EMMC_INT_EN 寄存器	141
表 16- 37	EMMC_INT_EN 寄存器位域描述	141
表 16- 38	DLL_MASTER_VAL 寄存器	142
表 16- 39	DLL_MASTER_VAL 寄存器位域描述	142
表 16- 40	DLL_CON 寄存器	142
表 16- 41	DLL_CON 寄存器位域描述	142
表 16- 42	PARAM_DELAY 寄存器	142



表 16- 43	PARAM_DELAY 寄存器位域描述	143
表 16- 44	SDIO_EMMC_SEL 寄存器	143
表 16- 45	SDIO_EMMC_SEL 寄存器位域描述	143
表 16- 46	DMA_ORDER_ADDR_LOW 寄存器	144
表 16- 47	DMA_SADDR 寄存器	144
表 16- 48	DMA_DADDR 寄存器	144
表 16- 49	DMA_LENGTH 寄存器	144
表 16- 50	DMA_STEP_LENGTH 寄存器	145
表 16- 51	DMA_STEP_TIMERS 寄存器	145
表 16- 52	DMA_CMD 寄存器	145
表 16- 53	DMA 写状态描述	146
表 16- 54	DMA 读状态描述	146
表 16- 55	DMA_ADDR_HIGH 寄存器	146
表 16- 56	DMA_SADDR_HIGH 寄存器	147
表 17- 1	CANFD 控制器地址空间分布	149
表 18- 1	工作模式配置	151
表 18- 2	ATIM 寄存器列表	153
表 18- 3	控制寄存器 1	154
表 18- 4	控制寄存器 2	155
表 18- 5	从模式控制寄存器	157
表 18- 6	DMA/中断使能寄存器	159
表 18- 7	状态寄存器	160
表 18- 8	事件产生寄存器	161
表 18- 9	捕获/比较模式寄存器 1(输出)	162
表 18- 10	捕获/比较模式寄存器 1(输入)	163
表 18- 11	捕获/比较模式寄存器 2(输出)	165
表 18- 12	捕获/比较模式寄存器 2(输入)	166
表 18- 13	捕获/比较使能寄存器	167
表 18- 14	计数器	169
表 18- 15	预分频器	169
表 18- 16	自动重装载寄存器	169
表 18- 17	重复计数寄存器	169
表 18- 18	捕获/比较寄存器 1	170
表 18- 19	捕获/比较寄存器 2	170



表 18- 20	捕获/比较寄存器 3	170
表 18- 21	捕获/比较寄存器 4	171
表 18- 22	刹车和死区寄存器	171
表 18- 23	输入通道状态寄存器	172
表 19- 1	工作模式配置	175
表 19- 2	GTIM 寄存器列表	177
表 19- 3	控制寄存器 1	178
表 19- 4	控制寄存器 2	179
表 19- 5	从模式控制寄存器	180
表 19- 6	DMA/中断使能寄存器	181
表 19- 7	状态寄存器	182
表 19- 8	事件产生寄存器	184
表 19- 9	捕获/比较模式寄存器 1(输出)	184
表 19- 10	捕获/比较模式寄存器 1(输入)	186
表 19- 11	捕获/比较模式寄存器 2(输出)	187
表 19- 12	捕获/比较模式寄存器 2(输入)	188
表 19- 13	捕获/比较使能寄存器	190
表 19- 14	计数器	191
表 19- 15	预分频器	191
表 19- 16	自动重装载寄存器	191
表 19- 17	捕获/比较寄存器 1	191
表 19- 18	捕获/比较寄存器 2	192
表 19- 19	捕获/比较寄存器 3	192
表 19- 20	捕获/比较寄存器 4	193
表 19- 21	输入通道状态寄存器	193
表 20- 1	GTIM 寄存器列表	194
表 20- 2	控制寄存器 1	195
表 20- 3	控制寄存器 2	196
表 20- 4	DMA/中断使能寄存器	196
表 20- 5	状态寄存器	197
表 20- 6	事件产生寄存器	197
表 20- 7	计数器	197
表 20- 8	预分频器	197
表 20- 9	自动重装载寄存器	198



表 21- 1	PWM 寄存器地址	199
表 21- 2	PWM 寄存器列表	199
表 21- 3	PWM 控制寄存器设置	199
表 22- 1	工作模式配置	203
表 22- 2	ADC 寄存器列表	205
表 22- 3	ADC 状态寄存器	206
表 22- 4	ADC 控制寄存器 1	206
表 22- 5	ADC 控制寄存器 1	208
表 22- 6	ADC 采样时间寄存器 1	210
表 22- 7	ADC 采样时间寄存器 2	211
表 22- 8	ADC 控制寄存器 1	212
表 22- 9	ADC 看门狗高阈值寄存器	212
表 22- 10	ADC 看门狗低阈值寄存器	212
表 22- 11	ADC 看门狗低阈值寄存器	212
表 22- 12	ADC 看门狗低阈值寄存器	213
表 22- 13	ADC 看门狗低阈值寄存器	214
表 22- 14	ADC 注入序列寄存器	214
表 22- 15	ADC 注入数据寄存器	214
表 22- 16	ADC 规则数据寄存器	215
表 23- 1	DMA 寄存器列表	216
表 23- 2	DMA 中断状态寄存器	217
表 23- 3	DMA 中断标志清除寄存器	218
表 23- 4	DMA 通道 x 配置寄存器	218
表 23- 5	DMA 通道 x 传输数量寄存器	219
表 23- 6	DMA 通道 x 外设地址寄存器	219
表 23- 7	DMA 通道 x 存储器地址寄存器	220
表 23- 8	外设请求的通道映射	220
表 24- 1	HPET 寄存器地址	222
表 24- 2	HPET 寄存器	222
表 24- 3	General Capabilities ID 寄存器	223
表 24- 4	General Configuration 寄存器	223
表 24- 5	General Interrupt Status 寄存器	224
表 24- 6	Main Counter Value 寄存器	224
表 24- 7	Timer N Configuration Capabilities 寄存器	224



表 24- 8	Timer N Comparator Value 寄存器	225
表 25- 1	RTC 寄存器列表	227
表 25- 2	TOY 计数器低 32 位写入寄存器	228
表 25- 3	TOY 计数器高 32 位写入寄存器	228
表 25- 4	TOY 计数器低 32 位读出寄存器	228
表 25- 5	TOY 计数器高 32 位读出寄存器	229
表 25- 6	TOY 计数器中断寄存器 0/1/2	229
表 25- 7	RTC 定时器中断寄存器 0/1/2	229
表 25- 8	RTC 计数器写入寄存器	230
表 25- 9	RTC 计数器读出寄存器	230
表 25- 10	RTC 定时器中断寄存器 0/1/2	231
表 26- 1	LA264 DFS Control 寄存器描述	233
表 26- 2	WDT 寄存器地址	233
表 26- 3	WD_EN 寄存器描述	234
表 26- 4	WD_SET 寄存器描述	234
表 26- 5	WD_TIMER 寄存器描述	234
表 27- 1	GPIO 寄存器地址	235
表 27- 2	GPIO 控制寄存器	235
表 27- 3	GPIO 模块内部寄存器物理地址	236
表 27- 4	按位控制 GPIO 配置寄存器地址	236
表 27- 5	按字节控制 GPIO 配置寄存器地址	236
表 27- 6	GPIO 方向按位控制寄存器	237
表 27- 7	GPIO 方向按字节控制寄存器	237
表 27- 8	GPIO 按位输出寄存器	237
表 27- 9	GPIO 按字节输出寄存器	238
表 27- 10	GPIO 按位输入寄存器	238
表 27- 11	GPIO 按字节输入寄存器	238
表 27- 12	GPIO 按位中断使能寄存器	238
表 27- 13	GPIO 按字节中断使能寄存器	238
表 27- 14	GPIO 按位中断极性寄存器	239
表 27- 15	GPIO 按字节中断极性寄存器	239
表 27- 16	GPIO 按位中断边沿寄存器	239
表 27- 17	GPIO 按字节中断边沿寄存器	239
表 27- 18	GPIO 按位中断清除寄存器	240



表 27- 19	GPIO 按字节中断清除寄存器	240
表 27- 20	GPIO 按位中断状态寄存器	240
表 27- 21	GPIO 按字节中断状态寄存器	241
表 27- 22	GPIO 按位中断双沿寄存器	241
表 27- 23	GPIO 按字节中断双沿寄存器	241



1 概述

龙芯 2K0300 芯片是基于 LA264 处理器核的多功能 SOC 芯片，可广泛适用于工业控制、通信设备、信息家电和物联网等应用领域。龙芯 2K0300 采用高集成度设计，提供丰富的功能接口，可满足多场景应用需求，同时支持低功耗技术，能够在低能耗表现下提供高效处理性能。

龙芯 2K0300 芯片片内集成 16 位 DDR4 内存控制器，并集成了丰富的外设接口：USB2.0 接口，GMAC 接口，DVO 显示接口，I2S 音频接口，SPI/QSPI，ADC，SDIO，eMMC 和其他工控领域常用接口。

龙芯 2K0300 具有以下关键特性：

- 集成一个 64 位双发射 LA264 处理器核，L1 Cache (I/D) 各 32KB，L2 Cache 512KB，最高主频 1.0GHz
- 集成 1 个 16 位 DDR4 控制器，典型速率 1600Mbps
- 集成 1 路显示控制器，最大分辨率可支持 1920*1080@60Hz/24bit
- 集成 2 个 10M/100M/1000M 自适应 GMAC，支持 RGMII/MII
- 集成 1 个 USB2.0 HOST 接口，1 个 OTG 接口支持 HOST/DEVICE 模式
- 集成 4 个 SPI 控制器，1 路支持系统启动 (SPI0)，2 路支持 QSPI 模式 (SPI0/1)
- 集成 4 路 I2C 控制器，支持主从模式
- 集成 1 路 I2S 控制器，支持单通道和多通道音频数据
- 集成 1 个 8 通道 12 位 AD 接口
- 集成 10 个 UART 控制器
- 集成 2 个 SDIO 控制器，均支持 SDIO/eMMC
- 集成 4 个 CAN 控制器，均支持 CAN-FD
- 集成 4 路 PWM 控制器，支持输入/输出
- 集成 3 组定时器 A/G/B-TIMER
- 集成 106 路复用 GPIO，支持位操作，支持输入/输出
- 集成 1 个温度传感器
- 集成 RTC/HPET
- 集成看门狗电路
- 集成动态功耗控制模块，支持 DFS
- 集成中断控制器，支持灵活的中断设置
- 支持 JTAG 调试

1.1 体系结构框图

龙芯 2K0300 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 I/O 子网络 (Cache 访问路径)。二级 Cache 及 IODMA、内存控制器、GMAC、USB、DC 等 I/O 设备共享高速系统互连网络。低速外设 (CAN/I2C/UART 等) 作为一个集合加在南桥总线上。

龙芯 2K0300 芯片结构图如图 1-1 所示：

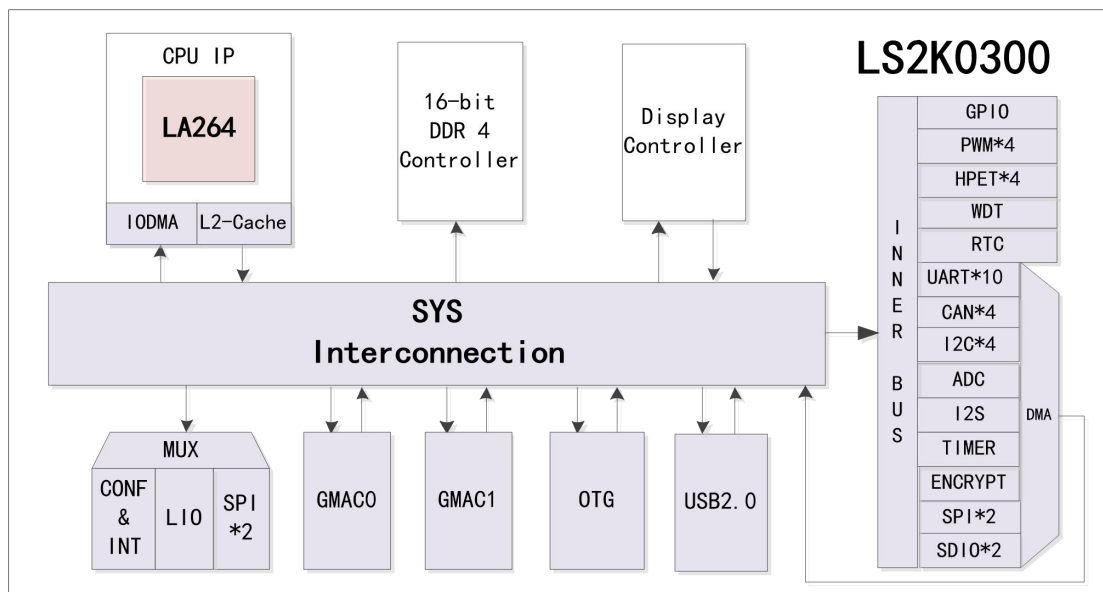


图1- 1 龙芯 2K0300 芯片结构图

1.2 芯片主要功能

1.2.1 处理器核

- 64 位双发射 LA264 核
- LoongArch 体系结构兼容
- 包括 1 个全流水的 64 位双精度浮点乘加部件
- 32KB 数据 Cache 和 32KB 的指令 Cache
- 512KB 共享二级 Cache
- 通过目录协议维护 I/O DMA 访问的 Cache 一致性
- JTAG 支持

1.2.2 内存接口

- 16 位 DDR4 控制器，支持 DDR4-1600
- 支持硬件 ECC
- 支持命令调度

1.2.3 显示接口

- 1 路 DVO 显示输出
- 分辨率可支持 320×240~1920×1080@60Hz/24bit

1.2.4 USB 控制器

- 1 个独立的 USB2.0 的 HOST 端口
- 1 个独立的 OTG 端口，支持 HOST/DEVICE 模式
- 兼容 USB1.1、USB2.0
- 内部 EHCI 控制和实现高速传输可达 480 Mbps
- 内部 OHCI 控制和实现全速和低速传输
- 低功耗管理

1.2.5 GMAC 控制器

- 两路 10/100/1000Mbps 自适应以太网 MAC
- 双网卡均兼容 IEEE 802.3
- 对外部 PHY 实现 RGMII/MII 接口
- 半双工/全双工自适应
- Timestamp 功能
- 半双工时，支持碰撞检测与重发（CSMA/CD）协议
- 支持 CRC 校验码的自动生成与校验，支持前置符生成与删除

1.2.6 I2S

- 1 路 I2S 接口
- 支持单声道和立体声道音频数据
- 支持 DMA 传输模式

1.2.7 SPI

- 双缓冲接收器
- 极性和相位可编程的串行时钟
- 主模式支持（SPI0/1）
- 主从模式支持（SPI2/3），支持 DMA 传输模式（SPI2/3）
- 支持到 4 个的变长字节传输
- 支持系统启动（仅 SPI0 支持）
- 支持 QSPI 模式（仅 SPI0/1 支持）
- 支持标准读、连续地址读、快速读、双路 I/O 等 SPI Flash 读模式

1.2.8 UART

- 2 个全功能 UART 和流控 TXD, RXD, CTS, RTS, DSR, DTR, DCD, RI
- 最多 10 个 UART 接口, 均支持 DMA 传输模式
- 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

1.2.9 I2C

- 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
- 支持 DMA 传输模式
- 总线的时钟频率可编程
- 可以产生开始/停止/应答等操作
- 能够对总线的状态进行探测
- 支持低速和快速模式
- 支持 7 位寻址和 10 位寻址
- 支持时钟延伸和等待状态

1.2.10 AD

- 支持 8 路 12 位 AD 电路采样
- 支持 DMA 传输模式

1.2.11 CAN

- 支持 4 路 CAN 接口, 支持 CAN-FD
- 支持 DMA 传输模式

1.2.12 TIMER

- 3 组定时器 A/G/B-TIMER
- ATIM/GTIM 支持输入捕获/PWM 输出, 支持 DMA 传输模式
- ATIM 支持 3 路互补 PWM 输出, GTIM 支持编码器/霍尔模式

1.2.13 PWM

- 32 位计数器

- 支持脉冲生成及捕获
- 4 路控制器

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 LocalIO

- 1 路 LocalIO 总线，支持最大 32MB 空间 MEM 访问
- 支持 8/16 位数据宽度，双片选扩展

1.2.16 HPET

- 4 个 32 位计数器
- 支持 1 个周期性中断
- 支持 2 个非周期性中断

1.2.17 RTC

- 计时精确到 0.1 秒
- 可产生 3 个计时中断

1.2.18 GPIO

- 106 位复用 GPIO 引脚
- 支持外部中断输入
- 与其他接口复用，使用各个接口电压域

1.2.19 Watchdog

- 32 比特计数器及初始化寄存器
- 低功耗模式暂停功能

1.2.20 温度传感器

- 温度观测
- 高低温中断

1.2.21 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
- 支持多种中断分发模式

1.3 芯片引脚复用关系

各信号引脚的功能复用关系如下表所示：

表 1- 1 芯片引脚复用关系

芯片引脚	GPI0 复用	主功能复用	第一复用	第二复用
LCD_CLK	GPI000	lcd_clk	-	-
LCD_VSYNC	GPI001	lcd_vsync	-	liao[0]
LCD_HSYNC	GPI002	lcd_hsync	-	liao[1]
LCD_EN	GPI003	lcd_en	-	liao[2]
LCD_D0	GPI004	lcd_dat_b[0]	-	liao[3]
LCD_D1	GPI005	lcd_dat_b[1]	-	liao[4]
LCD_D2	GPI006	lcd_dat_b[2]	-	liao[5]
LCD_D3	GPI007	lcd_dat_b[3]	-	liao[6]
LCD_D4	GPI008	lcd_dat_b[4]	-	liao[7]
LCD_D5	GPI009	lcd_dat_b[5]	-	liao[8]
LCD_D6	GPI010	lcd_dat_b[6]	-	liao[9]
LCD_D7	GPI011	lcd_dat_b[7]	-	liao[10]
LCD_D8	GPI012	lcd_dat_g[0]	-	liao[11]
LCD_D9	GPI013	lcd_dat_g[1]	-	liao[12]
LCD_D10	GPI014	lcd_dat_g[2]	-	liao[13]
LCD_D11	GPI015	lcd_dat_g[3]	-	liao[14]
LCD_D12	GPI016	lcd_dat_g[4]	-	liao[15]
LCD_D13	GPI017	lcd_dat_g[5]	-	liao[16]
LCD_D14	GPI018	lcd_dat_g[6]	-	liao[17]
LCD_D15	GPI019	lcd_dat_g[7]	-	liao[18]
LCD_D16	GPI020	lcd_dat_r[0]	-	liao[19]
LCD_D17	GPI021	lcd_dat_r[1]	-	liao[20]
LCD_D18	GPI022	lcd_dat_r[2]	-	liao[21]
LCD_D19	GPI023	lcd_dat_r[3]	-	liao[22]
LCD_D20	GPI024	lcd_dat_r[4]	-	liocsn[0]
LCD_D21	GPI025	lcd_dat_r[5]	-	liocsn[1]
LCD_D22	GPI026	lcd_dat_r[6]	-	liowrn
LCD_D23	GPI027	lcd_dat_r[7]	-	liordn

芯片引脚	GPIO 复用	主功能复用	第一复用	第二复用
GMACO_RX_CTL	GPIO28	gmac0_rx_ctl	-	tim1_ch1
GMACO_RX0	GPIO29	gmac0_rx[0]	-	tim1_ch2
GMACO_RX1	GPIO30	gmac0_rx[1]	-	tim1_ch3
GMACO_RX2	GPIO31	gmac0_rx[2]	-	tim1_ch1n
GMACO_RX3	GPIO32	gmac0_rx[3]	-	tim1_ch2n
GMACO_TX_CTL	GPIO33	gmac0_tx_ctl	-	tim1_ch3n
GMACO_TX0	GPIO34	gmac0_tx[0]	-	tim2_ch1
GMACO_TX1	GPIO35	gmac0_tx[1]	-	tim2_ch2
GMACO_TX2	GPIO36	gmac0_tx[2]	can_rx[0]	tim2_ch3
GMACO_TX3	GPIO37	gmac0_tx[3]	can_tx[0]	-
GMACO_MDCK	GPIO38	gmac0_mdck	can_rx[1]	-
GMACO_MDIO	GPIO39	gmac0_mdio	can_tx[1]	-
UART0_RX	GPIO40	uart0_rx	gmac0_ptp_trig	lio_data[0]
UART0_TX	GPIO41	uart0_tx	gmac0_ptp_pps	lio_data[1]
UART1_RX	GPIO42	uart1_rx	gmac1_ptp_trig	lio_data[2]
UART1_TX	GPIO43	uart1_tx	gmac1_ptp_pps	lio_data[3]
UART2_TX	GPIO44	uart2_tx	gmac1_rx_ctl	lio_data[4]
UART2_RX	GPIO45	uart2_rx	gmac1_rx[0]	lio_data[5]
UART3_TX	GPIO46	uart3_tx	gmac1_rx[1]	lio_data[6]
UART3_RX	GPIO47	uart3_rx	gmac1_rx[2]	lio_data[7]
I2C0_SCL	GPIO48	i2c_scl[0]	gmac1_rx[3]	lio_data[8]
I2C0_SDA	GPIO49	i2c_sda[0]	gmac1_tx_ctl	lio_data[9]
I2C1_SCL	GPIO50	i2c_scl[1]	gmac1_tx[0]	lio_data[10]
I2C1_SDA	GPIO51	i2c_sda[1]	gmac1_tx[1]	lio_data[11]
I2C2_SCL	GPIO52	i2c_scl[2]	gmac1_tx[2]	lio_data[12]
I2C2_SDA	GPIO53	i2c_sda[2]	gmac1_tx[3]	lio_data[13]
I2C3_SCL	GPIO54	i2c_scl[3]	gmac1_mdck	lio_data[14]
I2C3_SDA	GPIO55	i2c_sda[3]	gmac1_mdio	lio_data[15]
SPI0_CLK	GPIO56	spi0_clk	-	can_rx[2]
SPI0_MISO	GPIO57	spi0_miso	-	can_tx[2]
SPI0_MOSI	GPIO58	spi0_mosi	-	can_rx[3]
SPI0_CS	GPIO59	spi0_cs[0]	-	can_tx[3]
SPI1_CLK	GPIO60	spi1_clk	i2c_scl[0]	uart0_rts

芯片引脚	GPIO 复用	主功能复用	第一复用	第二复用
SPI1_MISO	GPIO61	spi1_miso	i2c_sda[0]	uart0_cts
SPI1_MOSI	GPIO62	spi1_mosi	i2c_scl[1]	uart0_dsr
SPI1_CS	GPIO63	spi1_cs	i2c_sda[1]	uart0_dtr
SPI2_CLK	GPIO64	spi2_clk	pwm[0]	uart0_dcd
SPI2_MISO	GPIO65	spi2_miso	pwm[1]	uart0_ri
SPI2_MOSI	GPIO66	spi2_mosi	pwm[2]	uart1_rts
SPI2_CS	GPIO67	spi2_cs	pwm[3]	uart1_cts
CAN0_RX	GPIO68	can_rx[0]	spi0_cs[1]	uart1_dsr
CAN0_TX	GPIO69	can_tx[0]	spi0_cs[2]	uart1_dtr
CAN1_RX	GPIO70	can_rx[1]	spi0_cs[3]	uart1_dcd
CAN1_TX	GPIO71	can_tx[1]	-	uart1_ri
CAN2_RX	GPIO72	can_rx[2]	sdio1_d[4]	gmac0_col
CAN2_TX	GPIO73	can_tx[2]	sdio1_d[5]	gmac0_crs
CAN3_RX	GPIO74	can_rx[3]	sdio1_d[6]	gmac1_col
CAN3_TX	GPIO75	can_tx[3]	sdio1_d[7]	gmac1_crs
I2S_MCLK	GPIO76	i2s_mclk	tim1_ch4	-
I2S_BCLK	GPIO77	i2s_bclk	tim2_ch4	-
I2S_LR	GPIO78	i2s_lr	atim_etr	spi1_cs[1]
I2S_DI	GPIO79	i2s_datai	gtim_etr	spi1_cs[2]
I2S_DO	GPIO80	i2s_datao	tim1_breakin	spi1_cs[3]
TIM1_CH1	GPIO81	tim1_ch1	-	-
TIM1_CH2	GPIO82	tim1_ch2	spi3_clk	i2c_scl[2]
TIM1_CH3	GPIO83	tim1_ch3	spi3_miso	i2c_sda[2]
TIM1_CH1N	GPIO84	tim1_ch1n	spi3_mosi	i2c_scl[3]
TIM1_CH2N	GPIO85	tim1_ch2n	spi3_cs	i2c_sda[3]
TIM1_CH3N	GPIO86	tim1_ch3n	sdio1_d[4]	pwm[0]
TIM2_CH1	GPIO87	tim2_ch1	sdio1_d[5]	pwm[1]
TIM2_CH2	GPIO88	tim2_ch2	sdio1_d[6]	pwm[2]
TIM2_CH3	GPIO89	tim2_ch3	sdio1_d[7]	pwm[3]
SDIO0_CLK	GPIO90	sdio0_clk	-	-
SDIO0_CMD	GPIO91	sdio0_cmd	-	-
SDIO0_D0	GPIO92	sdio0_d[0]	can_rx[0]	uart0_rx
SDIO0_D1	GPIO93	sdio0_d[1]	can_tx[0]	uart0_tx

芯片引脚	GPIO 复用	主功能复用	第一复用	第二复用
SDIO0_D2	GPI094	sdio0_d[2]	can_rx[1]	uart1_rx
SDIO0_D3	GPI095	sdio0_d[3]	can_tx[1]	uart1_tx
SDIO0_D4	GPI096	sdio0_d[4]	can_rx[2]	uart2_tx
SDIO0_D5	GPI097	sdio0_d[5]	can_tx[2]	uart2_rx
SDIO0_D6	GPI098	sdio0_d[6]	can_rx[3]	uart3_tx
SDIO0_D7	GPI099	sdio0_d[7]	can_tx[3]	uart3_rx
SDIO1_CLK	GPI0100	sdio1_clk	-	-
SDIO1_CMD	GPI0101	sdio1_cmd	tim1_ch4	-
SDIO1_D0	GPI0102	sdio1_d[0]	tim2_ch4	pwm[0]
SDIO1_D1	GPI0103	sdio1_d[1]	atim_etr	pwm[1]
SDIO1_D2	GPI0104	sdio1_d[2]	gtim_etr	pwm[2]
SDIO1_D3	GPI0105	sdio1_d[3]	tim1_breakin	pwm[3]

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，其中 GPI00~63 默认为输入状态，GPI064~105 默认为输出低电平状态。

2 时钟结构

龙芯 2K0300 的时钟结构如图2- 1 所示，片内主要由一路外部时钟输入，120MHz 固定频率时钟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 组频率上相互依赖的时钟输出。这 3 个 PLL 的用途分别为：

一个 NODE PLL 用于产生 NODE、GMAC、I2S 模块时钟，该时钟为芯片的主要时钟，分为三组频率时钟，一路供 CPU 核、二级 Cache、一二级交叉开关以及 IO 子网络使用，一路供 GMAC 模块使用，一路供 I2S 设备控制器使用；

一个 DDR PLL 同时产生 DDR、NETWORK 以及 USB、APB、BOOT、SDIO 等模块的时钟；

一个 PIX PLL 同时产生 DC 显示、GMAC 控制器备份的内部时钟。

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。

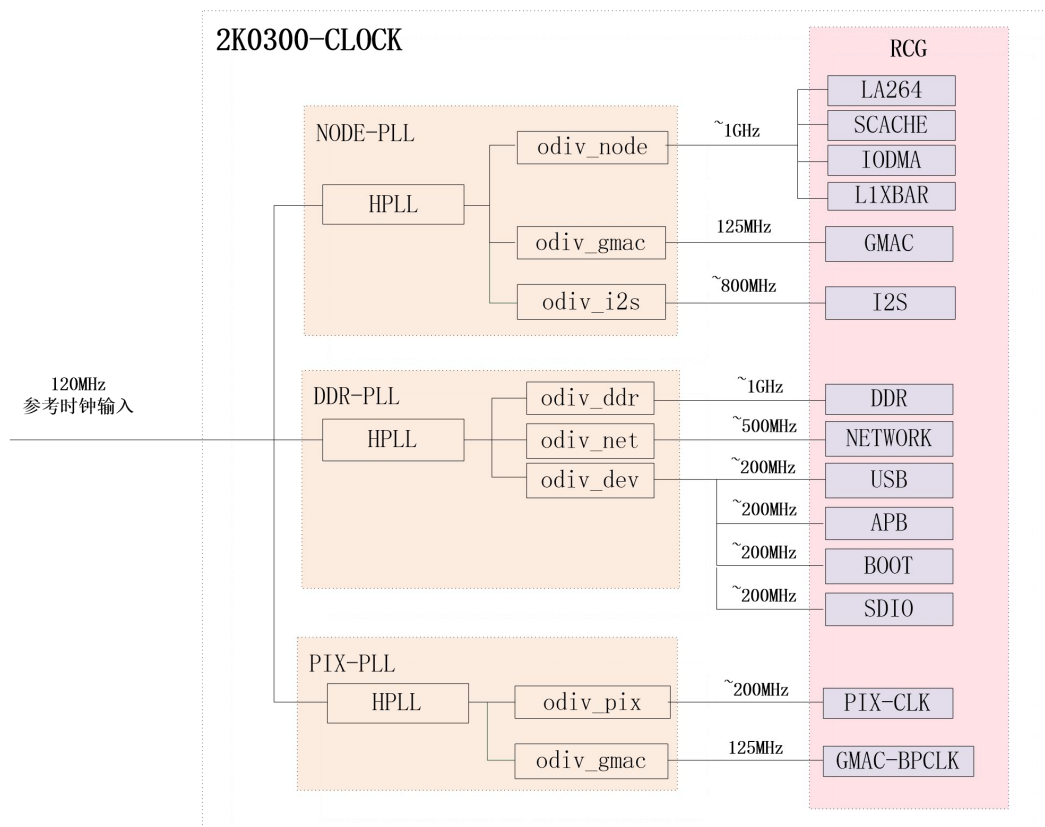


图2- 1 时钟结构

2.1 NODE PLL

NODE PLL 的产生结构图如下图所示，

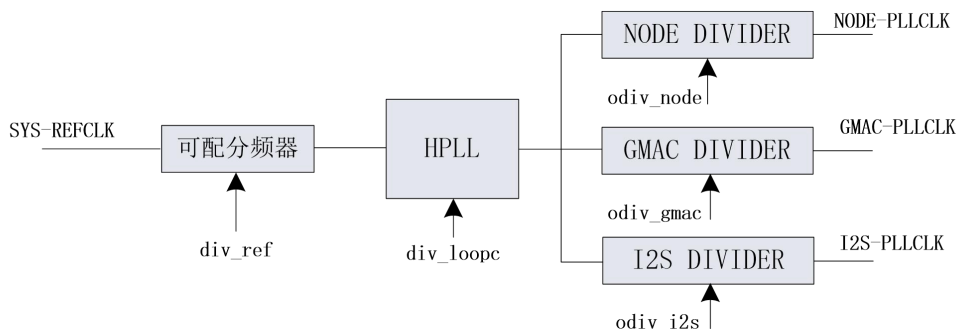


图2- 2 NODE PLL 结构图

NODE PLL 会输出三个时钟，分别为：

- 1) node_pllclk 用于 NODE 控制器（LA264/SCache），频率范围：~1GHz；
- 2) gmac_pllclk 用于 GMAC 控制器（GMAC0/GMAC1），频率值：125MHz；
- 3) i2s_pllclk 用于 I2S 控制器，频率范围：~800MHz。

输出时钟频率的计算方式如下：

- 1) $\text{node_pllclk} = \text{refclk} / \text{div_ref} * \text{div_loopc} / \text{odiv_node}$;
- 2) $\text{gmac_pllclk} = \text{refclk} / \text{div_ref} * \text{div_loopc} / \text{odiv_gmac}$;
- 3) $\text{i2s_pllclk} = \text{refclk} / \text{div_ref} * \text{div_loopc} / \text{odiv_i2s}$ 。

node_pllclk 的工作频率在~1GHz 左右，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 1GHz~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。

输出的时钟还可以经由 freq_scale 模块进行细粒度分频控制。具体分频方法请参考设备分频章节。

2.2 DDR PLL

DDR PLL 结构与 NODE 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：~1.0GHz；
- 2) network_pllclk 用于 NETWORK 模块，频率范围：~500MHz；
- 3) dev_pllclk 用于 USB、APB（除 SDIO 外其他 APB 低速接口控制器，包括：UART/I2C/PWM/CAN/ADC/I2S/SPI2/SPI3/HPET/GPIO/TIMER/ENCRYPT/RTC/WDT）、BOOT（Confbus/SPI0/SPI1/LocalIO 控制器）、SDIO 等模块，频率范围：~200MHz。

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。

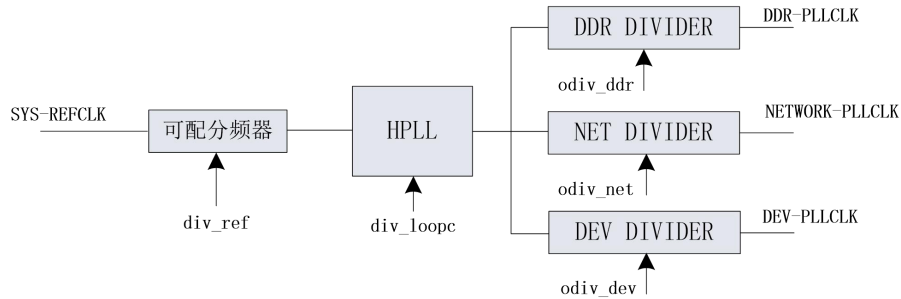


图2- 3 DDR PLL 时钟结构

2.3 PIX PLL

PIX PLL 结构与其他 PLL 结构类似，输出两个时钟，分别为：

- 1) pix_pllclk 用于显示接口，频率范围：~200MHz；
- 2) gmacbp_pllclk 用于网络接口备份时钟，频率值 125MHz。

两个时钟共用一个 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。

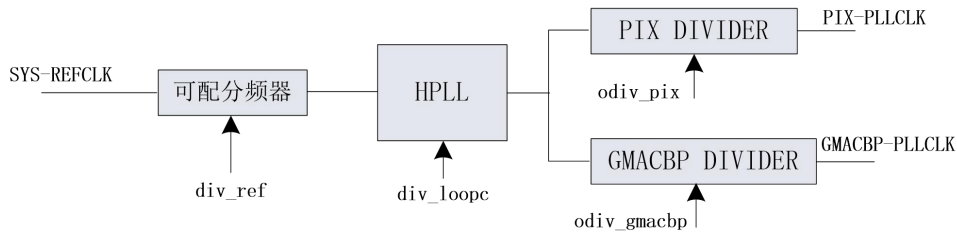


图2- 4 PIX PLL 时钟结构

2.4 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。

2.4.1 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	750M	1G	软件配置	硬件 bypass 所有

DDR	800M	1.2G		PLL，所有时钟频率与参考时钟相同（120MHz）
NETWORK	266M	400M		
GMAC	125M	125M		
I2S	750M	1G		
USB	100M	150M		
APB	100M	150M		
BOOT	100M	150M		
SDIO	100M	150M		
PIX	100M	200M		

2.4.2 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 3 章。

2.5 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：

- 1) 使用 1 个 24MHz 晶振输入；
- 2) 不使用单独的参考时钟输入，都使用 24MHz 的内部参考时钟（120MHz 的 SYSCLK 经五分频后）。

2.6 时钟信号说明

表 2-2 统一介绍了龙芯 2K0300 的所有时钟引脚。

表 2- 2 2K0300 时钟信号说明

时钟信号名称		推荐工作频率 (MHz)	类型	描述
SYSCLK		120M	I	外接系统参考时钟晶振
TESTCLK		100M	I	外接系统测试时钟
RTC_XI		32.768K	I/O	RTC 参考时钟晶体
RTC_X0				
EJTAG_TCK		~50M	I	JTAG 输入时钟
GMAC0/1_RX_CLK		125M	I	GMAC 接收输入时钟
GMAC0/1_TX_CLK_0		125M	O	GMAC 发送输出时钟
LCD_CLK		~200M	O	LCD 显示输出时钟
SPI0/1_CLK		~100M	O	SPI0/1 输出时钟
SPI2/3_CLK		~100M	I/O	SPI2/3 输出（主）/输入（从）时钟
SDIO0/1_CLK		~200M	O	SDIO0/1 输出时钟
I2S_BCLK		12.5M	I/O	I2S 音频时钟
I2S_MCLK		200M	O	I2S 输出主时钟
DDR_CKp[1:0]		~800M	DIFF OUT	DDR4 SDRAM 差分时钟输出
DDR_CKn[1:0]				
内部时钟	NODE_CLOCK	~1.0G	—	NODE 模块时钟，供 LA264、SCACHE、IODMA、L1-XBAR 模块使用
	STABLE_CLOCK	~120M	—	LA264 核内 STABLE 计数时钟, 由系统参考时钟提供，内部可分频
	DDR_CLOCK	~1.0G	—	DDR 控制器时钟，供 DDR4 控制器使用
	NETWORK_CLOCK	~500M	—	供 NETWORK 互联结构使用
	USB_CLOCK	~200M	—	USB/OTG 模块时钟
	GMAC_CLOCK	125M	—	GMAC0/1 模块时钟
	BOOT_CLOCK	~200M	—	BOOT 设备控制器时钟 (CONFBUS/SPI0/SPI1/LocalIO 控制器等)
	APB_CLOCK	~200M	—	APB 设备时钟（除 SDIO 外其他 APB 低速控制器）
	SDIO_CLOCK	~200M	—	SDIO/EMMC 模块时钟
	I2S_CLOCK	~800M	—	I2S 模块时钟
	THSENS_CLOCK	~1.2M	—	供 THSENS 设备使用，由系统参考时钟 100 分频后提供

3 芯片配置与控制

3.1 芯片初始化信号

龙芯 2K0300 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状态值得到芯片配置信息(复位结束后配置状态保持不变，芯片不再采样，对应引脚可做功能引脚使用)。相关配置信息编码成 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
LCD_D[0]	I	启动选择输入 0=SPIO 1=SDIO0/eMMC0
LCD_D [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
LCD_D [3]	I	SDIO0 模式配置输入 0=SDIO 模式 1=eMMC 模式
LCD_D [4]	I	SDIO1 模式配置输入 0=SDIO 模式 1=eMMC 模式
LCD_D[5]	I	USB REFCLOCK 模式输入配置 0=内部时钟输入 1=外部引脚输入
LCD_D [6]	I	eMMC0 引脚电平模式输入 0=3.3V-I/O 电平（对应 IO_3V3_1V8 供电需选择 3.3V 供电） 1=1.8V-I/O 电平（对应 IO_3V3_1V8 供电需选择 1.8V 供电）

3.2 地址空间分配

龙芯 2K0300 的地址空间根据系统应用不同，分为 LA264 处理器核、DMA 两个视角地址空间：所有 LA264 处理器核可访问的设备地址在系统地址空间上，为 LA264 视角；可通过 DMA 直接访问系统内存的设备所见到的空间为 DMA 视角。以下表格分别给出了这两个视角的具体定义。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 LA264 视角

地址空间 (unmapped)	大小	功能	拓扑
0x0000_0000 - 0x0fff_ffff	256MB	L-DDR	DDR
0x1000_0000 - 0x11ff_ffff	32MB	SPI0 MEM	MEM
0x1200_0000 - 0x13ff_ffff	32MB	SPI1 MEM	MEM
0x1400_0000 - 0x15ff_ffff	32MB	LIO MEM	MEM
0x1600_0000 - 0x1600_ffff	64KB	SYS-CONFBUS	CHIP-CONFIG、INT 等
0x1601_0000 - 0x1601_7fff	32KB	SPI0-I/O	
0x1601_8000 - 0x1601_ffff	32KB	SPI1-I/O	
0x1602_0000 - 0x1602_ffff	64KB	GMAC0	
0x1603_0000 - 0x1603_ffff	64KB	GMAC1	
0x1604_0000 - 0x1604_ffff	256KB	OTG	
0x1608_0000 - 0x1608_ffff	64KB	USB	
0x1609_0000 - 0x1609_ffff	64KB	DC	
		Reserved	
0x1610_0000 - 0x161f_ffff	1MB	DEVs (APB)	
		Reserved	
0x1c00_0000 - 0x1c0f_ffff	1MB	BOOT	
		Reserved	
地址空间 (APB-DEVs) [19:0] Base 0x1610_0000	大小 1MB	功能	拓扑
0x0_0000	16KB	UARTx10	每路 UART 占用 1KB 配置空间
0x0_4000	16KB	GPIOx106	可按照位、字节地址两种方式进行控制访问
0x0_8000	16KB	I2Cx4	每路 I2C 占用 4KB 配置空间
0x0_c000	16KB	SPIx2	每路 SPI 占用 8KB 配置空间
0x1_0000	16KB	CANx4	每路 CAN 占用 1KB 配置空间
0x1_4000	16KB	I2S	I2S 配置空间

0x1_8000	4KB	ATIMER	ATIM 配置空间
0x1_9000	4KB	GTIMER	GTIM 配置空间
0x1_a000	4KB	BTIMER	BTIM 配置空间
0x1_b000	4KB	PWM	4 路 PWM 每路各占 16B 配置空间
0x1_c000	16KB	ADC	ADC 配置空间
0x2_0000	16KB	HPET*4	每路 HPET 占用 4KB 配置空间
0x2_4000	16KB	WDT	
0x2_8000	16KB	RTC	
0x2_c000	16KB	DMA (共享)	8 通道共享 DMA，可供 UART、I2C、I2S、SPI、CAN、TIMER、ADC 共享使用
0x3_0000	512B	ENC-DMA0	加解密共享 DMA，可供 AES、DES、SM3、SM4 读写共享，可通过配置模式选择
0x3_0200	512B	ENC-DMA1	加解密共享 DMA，可供 AES、DES、SM3、SM4 读写共享，可通过配置模式选择
0x3_0400	1KB-16B	SM2x1	
0x3_07f0	16B	RNGx1	
0x3_0800	512B	AESx1	
0x3_0a00	512B	DESx1	
0x3_0c00	512B	SM3x1	
0x3_0e00	512B	SM4x1	
0x3_4000	16KB	OTP	FUSE 配置访问
0x4_0000	32KB	SDIO0	SDIO0/eMMC0 配置访问： 0x0~0x3ff:SDIO0/eMMC0 模块功能寄存器； 0x400~0x7ff:DMA 写配置寄存器； 0x800~0xffff:DMA 读配置寄存器。
0x4_8000	32KB	SDIO1	SDIO1/eMMC1 配置访问： 0x0~0x3ff:SDIO1/eMMC1 模块功能寄存器； 0x400~0x7ff:DMA 写配置寄存器； 0x800~0xffff:DMA 读配置寄存器。

龙芯 2K0300 内部可发起 DMA 的主设备包括 GMAC、USB、DC、SDIO、共享 DMA 等，主设备的请求可以访问路由到表 3-3 中的所有空间。

表 3- 3 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	

3.3 时钟与复位控制

3.3.1 时钟配置概要

龙芯 2K0300 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。

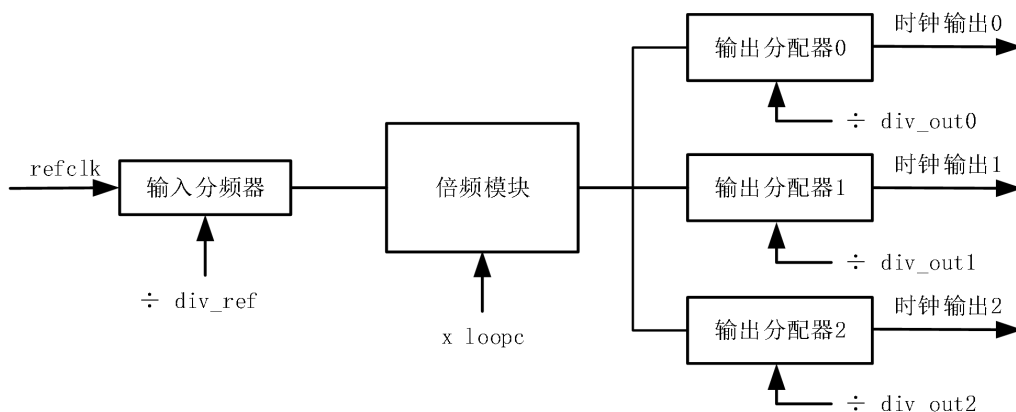


图3- 1 展频 PLL 概念性结构

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，龙芯 2K0300 的 refclk 固定为 120MHz，此外需要保证输入分频器的输出 (refclk / div_ref) 在 20 ~ 40MHz 的范围内，倍频模块倍频后的频率 (refclk / div_ref * loopc) 在 1.0GHz ~ 3.2GHz 的范围内。

PLL 相关的配置信号及说明见表 5-5。

表 3- 4 PLL 相关配置信号说明表

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

3.3.3 复位控制

龙芯 2K0300 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 2K0300 有大量的配置寄存器，多数分布于各个功能模块中，本节介绍芯片级的配置寄存器。

表 3- 5 系统芯片配置寄存器列表

地址	名称	描述
0x16000100	CHIP_CTRL00	芯片通用配置寄存器 00
0x16000104	CHIP_CTRL01	芯片通用配置寄存器 01
0x16000108	CHIP_CTRL02	芯片通用配置寄存器 02
0x1600010c	CHIP_CTRL03	芯片通用配置寄存器 03
0x16000110	CHIP_CTRL04	芯片通用配置寄存器 04
0x16000114	CHIP_CTRL05	芯片通用配置寄存器 05
0x16000118	CHIP_CTRL06	芯片通用配置寄存器 06
0x1600011c	CHIP_CTRL07	芯片通用配置寄存器 07
0x16000120	CHIP_CTRL08	芯片通用配置寄存器 08
0x16000124	CHIP_CTRL09	芯片通用配置寄存器 09
0x16000128	CHIP_CTRL10	芯片通用配置寄存器 10
0x1600012c	CHIP_CTRL11	芯片通用配置寄存器 11
0x16000130	CHIP_CTRL12	芯片通用配置寄存器 12
0x16000134	CHIP_CTRL13	芯片通用配置寄存器 13
0x16000138	CHIP_CTRL14	芯片通用配置寄存器 14
0x1600013c	CHIP_CTRL15	芯片通用配置寄存器 15
0x16000140	CHIP_SAMP0	芯片采样参数寄存器 0
0x16000144	CHIP_SAMP1	芯片采样参数寄存器 1
0x16000148	CHIP_SAMP2	芯片采样参数寄存器 2
0x1600014c	CHIP_SAMP3	芯片采样参数寄存器 3
0x16000150	CHIP_HPT0	芯片计数寄存器 0
0x16000154	CHIP_HPT1	芯片计数寄存器 1
0x16000400	PLL_NODE0	NODE PLL 时钟配置寄存器 0
0x16000404	PLL_NODE1	NODE PLL 时钟配置寄存器 1
0x16000408	PLL_DDRO	DDR PLL 时钟配置寄存器 0
0x1600040c	PLL_DDR1	DDR PLL 时钟配置寄存器 1
0x16000410	PLL_PIX0	PIX PLL 时钟配置寄存器 0
0x16000414	PLL_PIX1	PIX PLL 时钟配置寄存器 1
0x16000420	PLL_FREQSCALE	设备时钟分频配置寄存器
0x16000424	PLL_CLKEN	设备时钟输出使能配置寄存器
0x16000490	GPIO_CFG0	GPIO 复用配置寄存器 0
0x16000494	GPIO_CFG1	GPIO 复用配置寄存器 1
0x16000498	GPIO_CFG2	GPIO 复用配置寄存器 2

地址	名称	描述
0x1600049c	GPIO_CFG3	GPIO 复用配置寄存器 3
0x160004a0	GPIO_CFG4	GPIO 复用配置寄存器 4
0x160004a4	GPIO_CFG5	GPIO 复用配置寄存器 5
0x160004a8	GPIO_CFG6	GPIO 复用配置寄存器 6
0x16000500	USB_PHY0	USB PHY 配置寄存器 0
0x16000504	USB_PHY1	USB PHY 配置寄存器 1
0x16000508	USB_PHY2	USB PHY 配置寄存器 2
0x1600050c	USB_PHY3	USB PHY 配置寄存器 3
0x16001040	CORE_INTISR0	路由给 CORE 的低 32 位中断状态
0x16001044	INTISR0	低 32 位中断状态寄存器
0x16001048	CORE_INTISR1	路由给 CORE 的高 32 位中断状态
0x1600104c	INTISR1	高 32 位中断状态寄存器
0x16001148	EXTIOI_ACK	扩展中断反馈寄存器
0x16001400	ENTRY0_0	8 位中断路由寄存器[0--7]
0x16001408	ENTRY8_0	8 位中断路由寄存器[8--15]
0x16001410	ENTRY16_0	8 位中断路由寄存器[16--23]
0x16001418	ENTRY24_0	8 位中断路由寄存器[24--31]
0x16001420	INTISR_0	低 32 位中断状态寄存器
0x16001424	INTIEN_0	低 32 位中断使能状态寄存器
0x16001428	INTSET_0	低 32 位设置使能寄存器
0x1600142c	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
0x16001430	INTPOL_0	低 32 位极性设置寄存器(电平中断)
0x16001434	INTEDGE_0	低 32 位触发方式寄存器(1: 脉冲触发; 0: 电平触发)
0x16001440	ENTRY0_1	8 位中断路由寄存器[32--39]
0x16001448	ENTRY8_1	8 位中断路由寄存器[40--47]
0x16001450	ENTRY16_1	8 位中断路由寄存器[48--55]
0x16001458	ENTRY24_1	8 位中断路由寄存器[56--63]
0x16001460	INTISR_1	高 32 位中断状态寄存器
0x16001464	INTIEN_1	高 32 位中断使能状态寄存器
0x16001468	INTSET_1	高 32 位设置使能寄存器
0x1600146c	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
0x16001470	INTPOL_1	高 32 位极性设置寄存器(电平中断)

地址	名称	描述
0x16001474	INTEDGE_1	高 32 位触发方式寄存器 (1: 脉冲触发; 0: 电平触发)
0x160014c0	EXTIOI_MAP	扩展中断路由寄存器
0x16001500	Thsens_int_ctrl_Hi0	温度传感器高温中断控制寄存器 0
0x16001504	Thsens_int_ctrl_Hi1	温度传感器高温中断控制寄存器 1
0x16001508	Thsens_int_ctrl_Lo0	温度传感器低温中断控制寄存器 0
0x1600150c	Thsens_int_ctrl_Lo1	温度传感器低温中断控制寄存器 1
0x16001510	Thsens_int_status/clr	温度传感器中断状态寄存器
0x16001514	Thsens_value	温度传感器测量值 (仅低 11 位有效位), 计算公式为: $Tval = Thsens_val[10:0] * 0.57 - 394.7$
0x16001514	Thsens_cfg	温度传感器配置寄存器
0x16001520	Thsens_scale_hi0	温度传感器高阈值配置寄存器 0
0x16001524	Thsens_scale_hi1	温度传感器高阈值配置寄存器 1
0x16001600	EXTIOI_EN0	扩展中断使能寄存器 0
0x16001604	EXTIOI_EN1	扩展中断使能寄存器 1
0x16001608	EXTIOI_EN2	扩展中断使能寄存器 2
0x1600160c	EXTIOI_EN3	扩展中断使能寄存器 3
0x16001700	EXTIOI_ISR0	扩展中断状态寄存器 0
0x16001704	EXTIOI_ISR1	扩展中断状态寄存器 1
0x16001708	EXTIOI_ISR2	扩展中断状态寄存器 2
0x1600170c	EXTIOI_ISR3	扩展中断状态寄存器 3
0x16001800	EXTIOI_CORE_ISR0	CORE 扩展中断状态寄存器 0
0x16001804	EXTIOI_CORE_ISR1	CORE 扩展中断状态寄存器 1
0x16001808	EXTIOI_CORE_ISR2	CORE 扩展中断状态寄存器 2
0x1600180c	EXTIOI_CORE_ISR3	CORE 扩展中断状态寄存器 3
0x16003fe0	CHIP_ID4	芯片识别号 4
0x16003fe4	CHIP_ID5	芯片识别号 5
0x16003fe8	CHIP_ID6	芯片识别号 6
0x1600fec	CHIP_ID7	芯片识别号 7
0x16003ff0	CHIP_ID0	芯片识别号 0
0x16003ff4	CHIP_ID1	芯片识别号 1
0x16003ff8	CHIP_ID2	芯片识别号 2
0x16003ffc	CHIP_ID3	芯片识别号 3

3.4.1 芯片通用配置寄存器 00

芯片通用配置寄存器0，包括 GMAC、内存控制器的配置、UART 引脚复用模式、USB 等。
寄存器地址：0x16000100。

表 3- 6 芯片通用配置寄存器 00

位域	名称	访问	缺省值	描述
31	conf_usb_prefetch	RW	0x0	USB接口总线使能读预取
30	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
29	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
28	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
27:24	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uart7复用有效(若bit2为0则uart7复用为4线模式, 若bit2为1则复用为2线模式), 为0对应uart1的8线模式, uart7复用无效; Bit2: 为1对应uart8复用为2线模式, 为0对应uart1的8线模式, uart8复用无效; Bit3: 为1对应uart9复用为2线模式, 为0对应uart1的8线模式, uart9复用无效。 注: uart9复用为2线模式时, 请保证uart6也为2线模式。
23:20	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uart4复用有效(若bit2为0则uart4复用为4线模式, 若bit2为1则复用为2线模式), 为0对应uart0的8线模式, uart4复用无效; Bit2: 为1对应uart5复用为2线模式, 为0对应uart0的8线模式, uart5复用无效; Bit3: 为1对应uart6复用为2线模式, 为0对应uart0的8线模式, uart6复用无效。
19	extioi_en	RW	0x0	扩展中断使能控制位: 1: 打开扩展中断(传统/扩展中断同时有效); 0: 关闭扩展中断(仅传统中断有效)
18	reserved	RO	0x0	-

17	conf_rtc_timer_hspeed	RW	0x1	RTC计数器值快速访问使能配置位： 1：开启快速访问；0：关闭快速访问。
16	otp_ctrl_en	RW	0x0	OTP模块使能位
15:12	hpet_int_ctrl	RW	0x0	hpet0~3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	0x0	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 @0x0ff0,0000)，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	gmac1_test_lpbk	RW	0x0	GMAC1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。
2	gmac0_test_lpbk	RW	0x0	GMAC0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。
1	gmac1_mii_sel	RW	0x0	GMAC1接口MII模式选择： 1：MII接口模式；0：RGMII接口模式
0	gmac0_mii_sel	RW	0x0	GMAC0接口MII模式选择： 1：MII接口模式；0：RGMII接口模式

3.4.2 芯片通用配置寄存器 01

芯片通用配置寄存器01。

寄存器地址：0x16000104。

表 3- 7 芯片通用配置寄存器 01

位域	名称	访问	缺省值	描述
31:20	reserved	RO	0x0	—
19	lio_rom_width16	RW	0x0	ROM空间访问8/16位数据位宽配置位： 1：16位模式；0：8位模式。

18:14	lio_rom_count_init	RW	0x0	ROM空间访问延迟初始值：0~31，即：访问延迟计时以该配置值（内部按位取反）为起始开始计数，最高计数至32停止。
13:12	lio_clk_period	RW	0x0	LIO总线访问单位延迟的时钟周期数（即：每累加1个延迟计时占多少LIO控制器时钟周期数）： 00: 1; 01: 4; 10: 2; 11: 1。
11:4	conf_iodma_spare_rd	RW	0x0	iodma读操作最大数设置
3: 0	conf_usb_flush_idle	RW	0xf	设置清空write buffer前空闲周期数

3.4.3 芯片通用配置寄存器 02

芯片通用配置寄存器02。

寄存器地址：0x16000108。

表 3- 8 芯片通用配置寄存器 02

位域	名称	访问	缺省值	描述
31:12	reserved	RO	0x0	—
11:4	pmu_dfs_clkdiv	RW	0x0	PLL时钟输出分频参数配置，最高位CLKDIV[7]为软件参数配置使能位(高有效)： 对应PLL时钟配置参数CLKDIV[6:0]：0~127
2	pmu_stpclk_n	RW	0x0	时钟关断配置位，高电平时钟关断。
1	pmu_dfs_clkbyp	RW	0x0	时钟BYPASS配置位： 1：时钟BYPASS为系统参考时钟；0：时钟为PLL输出时钟。
0	pmu_dfs_en	RW	0x0	动态调频(DFS)使能位，高电平有效。

3.4.4 芯片通用配置寄存器 03

芯片通用配置寄存器03，包括对 USB、GMAC 等 Cache 一致性配置等。

寄存器地址：0x1600010c。

表 3- 9 芯片通用配置寄存器 03

位域	名称	访问	缺省值	描述
31:30	reserved	RO	0x0	—
29	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）
28	dc_read_upgrade	RW	0x0	dc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）
27	gmac1_read_upgrade	RW	0x0	GMAC1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）
26	gmac0_read_upgrade	RW	0x0	GMAC0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）

25	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）
24:22	reserved	RO	0x0	—
21	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）
20	dc_write_upgrade	RW	0x0	dc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）
19	gmac1_write_upgrade	RW	0x0	GMAC1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）
18	gmac0_write_upgrade	RW	0x0	GMAC0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）
17	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）
16:14	reserved	RO	0x0	—
13	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
12	dc_coherent	RW	0x0	dc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
11	gmac1_coherent	RW	0x0	GMAC1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
10	gmac0_coherent	RW	0x0	GMAC0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
9	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
8:6	reserved	RO	0x0	—
5	apb_coherent_enable	RW	0x0	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)

4	dc_coherent_enable	RW	0x0	dc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)
3	gmac1_coherent_enable	RW	0x0	GMAC1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)
2	gmac0_coherent_enable	RW	0x0	GMAC0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)
1	usb_coherent_enable	RW	0x0	USB/OTG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)
0	reserved	RO	0x0	—

3.4.5 芯片通用配置寄存器 04

芯片通用配置寄存器04。

寄存器地址：0x16000110。

表 3- 10 芯片通用配置寄存器 04

位域	名称	访问	缺省值	描述
31:29	reserved	RO	0x0	—
28:26	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。
25:24	pad_ctrl_usb	RW	0x1	USB OC引脚驱动类型参数配置：

				00: 2mA; 01: 4mA; 10: 8mA; 11: 12mA。
23:22	pad_ctrl_timer	RW	0x1	TIMER1引脚驱动类型参数配置: 00: 2mA; 01: 4mA; 10: 8mA; 11: 12mA。
21:20	pad_ctrl_i2s	RW	0x1	I2S引脚驱动类型参数配置: 00: 2mA; 01: 4mA; 10: 8mA; 11: 12mA。
19:18	pad_ctrl_spi	RW	0x1	SPI0~1引脚驱动类型参数配置: 00: 2mA; 01: 4mA; 10: 8mA; 11: 12mA。
17:16	pad_ctrl_sdio	RW	0x2	SDIO1引脚驱动类型参数配置: 00: 2mA; 01: 4mA; 10: 8mA; 11: 12mA。
15:14	pad_ctrl_gmac	RW	0x2	GMAC0~1引脚驱动类型参数配置: 00: 2mA; 01: 4mA; 10: 8mA; 11: 12mA。
13:12	pad_ctrl_uart	RW	0x0	UART0~1引脚驱动类型参数配置: 00: 2mA; 01: 4mA; 10: 8mA; 11: 12mA。
11:10	pad_ctrl_dvo	RW	0x2	DVO显示引脚驱动类型参数配置: 00: 2mA; 01: 4mA; 10: 8mA; 11: 12mA。
9:8	pad_ctrl_jtag	RW	0x0	JTAG引脚驱动类型参数配置: 00: 2mA; 01: 4mA; 10: 8mA; 11: 12mA。
7:6	reserved	RO	0x0	–
5	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
4	dc_order_en	RW	0x0	DC内部互联读写请求按序执行使能位, 高电平有效
3	gmac1_order_en	RW	0x0	GMAC1内部互联读写请求按序执行使能位, 高电平有效
2	gmac0_order_en	RW	0x0	GMAC0内部互联读写请求按序执行使能位, 高电平有效
1	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
0	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

3.4.6 芯片通用配置寄存器 05

芯片通用配置寄存器05, 包括各功能模块时钟门控的控制。

寄存器地址: 0x16000114。

表 3- 11 芯片通用配置寄存器 05

位域	名称	访问	缺省值	描述
31	atimer_clk_ctrl	RW	0x1	atimer模块时钟门控配置: 1: 时钟门控打开; 0: 时钟门控关闭。
30:27	hpet_clk_ctrl[3:0]	RW	0xf	hpet0~3模块时钟门控配置: 1: 时钟门控打开; 0: 时钟门控关闭。
26:23	i2c_clk_ctrl[3:0]	RW	0xf	i2c0~3模块时钟门控配置:

				1: 时钟门控打开; 0: 时钟门控关闭。
22:19	can_clk_ctrl[3:0]	RW	0xf	can0~3模块时钟门控配置: 1: 时钟门控打开; 0: 时钟门控关闭。
18:17	spi_clk_ctrl[3:2]	RW	0x3	spi2~3模块时钟门控配置: 1: 时钟门控打开; 0: 时钟门控关闭。
16	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。
15	otp_clk_ctrl	RW	0x1	otp模块时钟门控配置: 1: 时钟门控打开; 0: 时钟门控关闭。
14	rtc_clk_ctrl	RW	0x1	rtc模块时钟门控配置: 1: 时钟门控打开; 0: 时钟门控关闭。
13	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。
12	adc_clk_ctrl	RW	0x1	adc模块时钟门控配置: 1: 时钟门控打开; 0: 时钟门控关闭。
11	i2s_clk_ctrl	RW	0x1	i2s模块时钟门控配置: 1: 时钟门控打开; 0: 时钟门控关闭。
10	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。
9	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。
8	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。
7	usbm_clk_ctrl	RW	0x1	usb+otg模块时钟门控配置: 1: 时钟门控打开; 0: 时钟门控关闭。
6	gmac1_clk_ctrl	RW	0x1	gmac1模块时钟门控配置: 1: 时钟门控打开; 0: 时钟门控关闭。
5	gmac0_clk_ctrl	RW	0x1	gmac0模块时钟门控配置: 1: 时钟门控打开; 0: 时钟门控关闭。
4	dc_clk_ctrl	RW	0x1	dc模块时钟门控配置: 1: 时钟门控打开; 0: 时钟门控关闭。
3	lio_clk_ctrl	RW	0x1	lio模块时钟门控配置: 1: 时钟门控打开; 0: 时钟门控关闭。
2	spi1_clk_ctrl	RW	0x1	spi1模块时钟门控配置: 1: 时钟门控打开; 0: 时钟门控关闭。
1	spi0_clk_ctrl	RW	0x1	spi0模块时钟门控配置: 1: 时钟门控打开; 0: 时钟门控关闭。
0	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。

3.4.7 芯片通用配置寄存器 06

芯片通用配置寄存器06，包括各功能模块时钟门控的控制。

寄存器地址：0x16000118。

表 3- 12 芯片通用配置寄存器 06

位域	名称	访问	缺省值	描述
31:29	reserved	RO	0x0	—
28	apbm_clk_ctrl	RW	0x1	apb全部模块时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
27	sdio_clk_ctrl	RW	0x1	sdio全部模块时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
26	sm4_clk_ctrl	RW	0x1	sm4模块时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
25	sm3_clk_ctrl	RW	0x1	sm3模块时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
24	des_clk_ctrl	RW	0x1	des模块时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
23	aes_clk_ctrl	RW	0x1	aes模块时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
22	rng_clk_ctrl	RW	0x1	rng模块时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
21	sm2_clk_ctrl	RW	0x1	sm2模块时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
20	encdma_clk_ctrl	RW	0x1	encdma模块时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
19	canram_clk_ctrl	RW	0x1	canram模块时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
18:15	canbuf_clk_ctrl[3:0]	RW	0xf	canbuf0~3模块时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
14:13	sdio_clk_ctrl[1:0]	RW	0x3	sdio0~1模块时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
12:3	uart_clk_ctrl[9:0]	RW	0x3ff	uart0~9模块时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
2	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。
1	btimer_clk_ctrl	RW	0x1	btimer模块时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。
0	gtimer_clk_ctrl	RW	0x1	gtimer模块时钟门控配置： 1：时钟门控打开；0：时钟门控关闭。

3.4.8 芯片通用配置寄存器 07

芯片通用配置寄存器07，包括各功能模块软件复位控制。

寄存器地址：0x1600011c。

表 3- 13 芯片通用配置寄存器 07

位域	名称	访问	缺省值	描述
31	atimer_rst_ctrl	RW	0x1	atimer模块软件复位配置： 1：软件复位无效；0：软件复位有效。
30:27	hpet_rst_ctrl[3:0]	RW	0xf	hpet0~3模块软件复位配置： 1：软件复位无效；0：软件复位有效。
26:23	i2c_rst_ctrl[3:0]	RW	0xf	i2c0~3模块软件复位配置： 1：软件复位无效；0：软件复位有效。
22:19	can_rst_ctrl[3:0]	RW	0xf	can0~3模块软件复位配置： 1：软件复位无效；0：软件复位有效。
18:17	spi_rst_ctrl[3:2]	RW	0x3	spi2~3模块软件复位配置： 1：软件复位无效；0：软件复位有效。
16	dma_rst_ctrl	RW	0x1	dma模块软件复位配置： 1：软件复位无效；0：软件复位有效。
15	otp_rst_ctrl	RW	0x1	otp模块软件复位配置： 1：软件复位无效；0：软件复位有效。
14	rtc_rst_ctrl	RW	0x1	rtc模块软件复位配置： 1：软件复位无效；0：软件复位有效。
13	wdt_rst_ctrl	RW	0x1	wdt模块软件复位配置： 1：软件复位无效；0：软件复位有效。
12	adc_rst_ctrl	RW	0x1	adc模块软件复位配置： 1：软件复位无效；0：软件复位有效。
11	i2s_rst_ctrl	RW	0x1	i2s模块软件复位配置： 1：软件复位无效；0：软件复位有效。
10	gpio_rst_ctrl	RW	0x1	gpio模块软件复位配置： 1：软件复位无效；0：软件复位有效。
9	otg_rst_ctrl	RW	0x1	otg模块软件复位配置： 1：软件复位无效；0：软件复位有效。
8	usb_rst_ctrl	RW	0x1	usb模块软件复位配置： 1：软件复位无效；0：软件复位有效。
7	usbm_rst_ctrl	RW	0x1	usb+otg全部模块软件复位配置： 1：软件复位无效；0：软件复位有效。
6	gmac1_rst_ctrl	RW	0x1	gmac1模块软件复位配置： 1：软件复位无效；0：软件复位有效。
5	gmac0_rst_ctrl	RW	0x1	gmac0模块软件复位配置： 1：软件复位无效；0：软件复位有效。
4	dc_rst_ctrl	RW	0x1	dc模块软件复位配置： 1：软件复位无效；0：软件复位有效。

3	lio_rst_ctrl	RW	0x1	lio模块软件复位配置： 1：软件复位无效；0：软件复位有效。
2	spi1_rst_ctrl	RW	0x1	spi1模块软件复位配置： 1：软件复位无效；0：软件复位有效。
1	spi0_rst_ctrl	RW	0x1	spi0模块软件复位配置： 1：软件复位无效；0：软件复位有效。
0	ddr_rst_ctrl	RW	0x1	ddr模块软件复位配置： 1：软件复位无效；0：软件复位有效。

3.4.9 芯片通用配置寄存器 08

芯片通用配置寄存器08，包括各功能模块软件复位的控制。

寄存器地址：0x16000120。

表 3- 14 芯片通用配置寄存器 08

位域	名称	访问	缺省值	描述
31:29	reserved	R0	0x0	–
28	apbm_rst_ctrl	RW	0x1	apb全部模块软件复位配置： 1：软件复位无效；0：软件复位有效。
27	sdio_rst_ctrl	RW	0x1	sdio全部模块软件复位配置： 1：软件复位无效；0：软件复位有效。
26	sm4_rst_ctrl	RW	0x1	sm4模块软件复位配置： 1：软件复位无效；0：软件复位有效。
25	sm3_rst_ctrl	RW	0x1	sm3模块软件复位配置： 1：软件复位无效；0：软件复位有效。
24	des_rst_ctrl	RW	0x1	des模块软件复位配置： 1：软件复位无效；0：软件复位有效。
23	aes_rst_ctrl	RW	0x1	aes模块软件复位配置： 1：软件复位无效；0：软件复位有效。
22	rng_rst_ctrl	RW	0x1	rng模块软件复位配置： 1：软件复位无效；0：软件复位有效。
21	sm2_rst_ctrl	RW	0x1	sm2模块软件复位配置： 1：软件复位无效；0：软件复位有效。
20	encdma_rst_ctrl	RW	0x1	encdma模块软件复位配置： 1：软件复位无效；0：软件复位有效。
19	canram_rst_ctrl	RW	0x1	canram模块软件复位配置： 1：软件复位无效；0：软件复位有效。
18:15	canbuf_rst_ctrl[3:0]	RW	0xf	canbuf0~3模块软件复位配置： 1：软件复位无效；0：软件复位有效。
14:13	sdio_rst_ctrl[1:0]	RW	0x3	sdio0~1模块软件复位配置： 1：软件复位无效；0：软件复位有效。
12:3	uart_rst_ctrl[9:0]	RW	0x3ff	uart0~9模块软件复位配置：

				1: 软件复位无效; 0: 软件复位有效。
2	pwm_rst_ctrl	RW	0x1	pwm模块软件复位配置: 1: 软件复位无效; 0: 软件复位有效。
1	btimer_rst_ctrl	RW	0x1	btimer模块软件复位配置: 1: 软件复位无效; 0: 软件复位有效。
0	gtimer_rst_ctrl	RW	0x1	gtimer模块软件复位配置: 1: 软件复位无效; 0: 软件复位有效。

3.4.10 芯片通用配置寄存器 09

芯片通用配置寄存器09。

寄存器地址: 0x16000124。

表 3- 15 芯片通用配置寄存器 09

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

3.4.11 芯片通用配置寄存器 10

芯片通用配置寄存器10。

寄存器地址: 0x16000128。

表 3- 16 芯片通用配置寄存器 10

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

3.4.12 芯片通用配置寄存器 11

芯片通用配置寄存器11。

寄存器地址: 0x1600012c。

表 3- 17 芯片通用配置寄存器 11

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

3.4.13 芯片通用配置寄存器 12

芯片通用配置寄存器12。

寄存器地址: 0x16000130。

表 3- 18 芯片通用配置寄存器 12

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

3.4.14 芯片通用配置寄存器 13

芯片通用配置寄存器13, 包括 DMA 通道映射。

寄存器地址: 0x16000134。

表 3- 19 芯片通用配置寄存器 13

位域	名称	访问	缺省值	描述
31:30	uart7_dma_map	RW	0x3	UART7模块DMA通道映射： RX、TX对应的通用DMA通道分别为 00：通道0、1； 01：通道2、3； 10：通道4、5； 11：通道6、7。
29:28	uart6_dma_map	RW	0x2	UART6模块DMA通道映射： RX、TX对应的通用DMA通道分别为 00：通道0、1； 01：通道2、3； 10：通道4、5； 11：通道6、7。
27:26	uart5_dma_map	RW	0x1	UART5模块DMA通道映射： RX、TX对应的通用DMA通道分别为 00：通道0、1； 01：通道2、3； 10：通道4、5； 11：通道6、7。
25:24	uart4_dma_map	RW	0x0	UART4模块DMA通道映射： RX、TX对应的通用DMA通道分别为 00：通道0、1； 01：通道2、3； 10：通道4、5； 11：通道6、7。
23:22	uart3_dma_map	RW	0x3	UART3模块DMA通道映射： RX、TX对应的通用DMA通道分别为 00：通道0、1； 01：通道2、3； 10：通道4、5； 11：通道6、7。
21:20	uart2_dma_map	RW	0x2	UART2模块DMA通道映射： RX、TX对应的通用DMA通道分别为 00：通道0、1； 01：通道2、3； 10：通道4、5； 11：通道6、7。
19:18	uart1_dma_map	RW	0x1	UART1模块DMA通道映射： RX、TX对应的通用DMA通道分别为 00：通道0、1； 01：通道2、3； 10：通道4、5； 11：通道6、7。
17:16	uart0_dma_map	RW	0x0	UART0模块DMA通道映射： RX、TX对应的通用DMA通道分别为 00：通道0、1； 01：通道2、3； 10：通道4、5； 11：通道6、7。
15:14	reserved	RO	0x0	–
13:12	sm3_dma_map	RW	0x0	SM3模块数据DMA通道路由选择： 00：选择加解密DMA通道0； 01：选择加解密 DMA通道1；其他：无效
11:10	sm4_rdma_map	RW	0x1	SM4模块读数据DMA通道路由选择： 00：选择加解密DMA通道0； 01：选择加解密 DMA通道1；其他：无效

9:8	sm4_wdma_map	RW	0x0	SM4模块写数据DMA通道路由选择： 00：选择加解密DMA通道0；01：选择加解密DMA通道1；其他：无效
7:6	des_rdma_map	RW	0x1	DES模块读数据DMA通道路由选择： 00：选择加解密DMA通道0；01：选择加解密DMA通道1；其他：无效
5:4	des_wdma_map	RW	0x0	DES模块写数据DMA通道路由选择： 00：选择加解密DMA通道0；01：选择加解密DMA通道1；其他：无效
3:2	aes_rdma_map	RW	0x1	AES模块读数据DMA通道路由选择： 00：选择加解密DMA通道0；01：选择加解密DMA通道1；其他：无效
1:0	aes_wdma_map	RW	0x0	AES模块写数据DMA通道路由选择： 00：选择加解密DMA通道0；01：选择加解密DMA通道1；其他：无效

3.4.15 芯片通用配置寄存器 14

芯片通用配置寄存器14，包括 DMA 通道映射。

寄存器地址：0x16000138。

表 3- 20 芯片通用配置寄存器 14

位域	名称	访问	缺省值	描述
31:30	can3_dma_map[1:0]	RW	0x3	CAN3模块DMA通道映射，低2位： RX对应的DMA通道为 000-111：通用DMA通道0-7。
29:27	can2_dma_map	RW	0x2	CAN2模块DMA通道映射： RX对应的DMA通道为 000-111：通用DMA通道0-7。
26:24	can1_dma_map	RW	0x1	CAN1模块DMA通道映射： RX对应的DMA通道为 000-111：通用DMA通道0-7。
23:21	can0_dma_map	RW	0x0	CAN0模块DMA通道映射： RX对应的DMA通道为 000-111：通用DMA通道0-7。
20:18	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。
17:16	i2s_dma_map	RW	0x0	I2S模块DMA通道映射： RX、TX对应的通用DMA通道分别为 00：通道0、1； 01：通道2、3； 10：通道4、5； 11：通道6、7。

15:14	spi3_dma_map	RW	0x1	SPI3模块DMA通道映射： RX、TX对应的通用DMA通道分别为 00: 通道0、1； 01: 通道2、3； 10: 通道4、5； 11: 通道6、7。
13:12	spi2_dma_map	RW	0x0	SPI2模块DMA通道映射： RX、TX对应的通用DMA通道分别为 00: 通道0、1； 01: 通道2、3； 10: 通道4、5； 11: 通道6、7。
11:10	i2c3_dma_map	RW	0x3	I2C3模块DMA通道映射： RX、TX对应的通用DMA通道分别为 00: 通道0、1； 01: 通道2、3； 10: 通道4、5； 11: 通道6、7。
9:8	i2c2_dma_map	RW	0x2	I2C2模块DMA通道映射： RX、TX对应的通用DMA通道分别为 00: 通道0、1； 01: 通道2、3； 10: 通道4、5； 11: 通道6、7。
7:6	i2c1_dma_map	RW	0x1	I2C1模块DMA通道映射： RX、TX对应的通用DMA通道分别为 00: 通道0、1； 01: 通道2、3； 10: 通道4、5； 11: 通道6、7。
5:4	i2c0_dma_map	RW	0x0	I2C0模块DMA通道映射： RX、TX对应的通用DMA通道分别为 00: 通道0、1； 01: 通道2、3； 10: 通道4、5； 11: 通道6、7。
3:2	uart9_dma_map	RW	0x3	UART9模块DMA通道映射： RX、TX对应的通用DMA通道分别为 00: 通道0、1； 01: 通道2、3； 10: 通道4、5； 11: 通道6、7。
1:0	uart8_dma_map	RW	0x2	UART8模块DMA通道映射： RX、TX对应的通用DMA通道分别为 00: 通道0、1； 01: 通道2、3； 10: 通道4、5； 11: 通道6、7。

3.4.16 芯片通用配置寄存器 15

芯片通用配置寄存器15。

寄存器地址：0x1600013c。

表 3- 21 芯片通用配置寄存器 15

位域	名称	访问	缺省值	描述
31	id_read_disable	RW	0x0	ID读使能
30:1	reserved	RO	0x0	—
0	can3_dma_map[2]	RW	0x0	CAN3模块DMA通道映射，高1位：

				RX对应的DMA通道为 000-111：通用DMA通道0-7。
--	--	--	--	------------------------------------

3.4.17 芯片采样参数寄存器 0

芯片采样参数寄存器0，包括芯片启动配置相关信息。

寄存器地址：0x16000140。

表 3- 22 芯片采样参数寄存器 0

位域	名称	访问	缺省值	描述
31:7	reserved	RO	0x0	—
6	emmc_padtype	RO	0x0	EMMC0 PAD电平类型 0:3.3V-I/O类型 1:1.8V-I/O类型
5	usb_refclkmode	RO	0x0	usb/otg参考时钟模式输入 0:内部参考时钟输入（时钟频率24MHz） 1:外部PAD晶振输入(时钟频率24MHz)
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：SPI启动；1：SDIO启动。

3.4.18 芯片采样参数寄存器 1

芯片采样参数寄存器1。

寄存器地址：0x16000144。

表 3- 23 芯片采样参数寄存器 1

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

3.4.19 芯片采样参数寄存器 2

芯片采样参数寄存器2，DDR4 ECC 相关状态寄存器。

寄存器地址：0x16000148。

表 3- 24 芯片采样参数寄存器 2

位域	名称	访问	缺省值	描述
31:16	ddr4_ecc_addr[15:0]	R0		DDR ECC错误地址[15:0]
15:0	ddr4_ecc_count	R0		DDR ECC错误计数

3.4.20 芯片采样参数寄存器 3

芯片采样参数寄存器3，DDR4 ECC 相关状态寄存器。

寄存器地址：0x1600014c。

表 3- 25 芯片采样参数寄存器 3

位域	名称	访问	缺省值	描述
31:16	reserved	R0	0x0	—
15:0	ddr4_ecc_addr[31:16]	R0		DDR ECC错误地址[31:16]

3.4.21 芯片计数寄存器 0

芯片计数寄存器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：0x16000150。

表 3- 26 芯片计数寄存器 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

3.4.22 芯片计数寄存器 1

芯片计数寄存器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。

寄存器地址：0x16000154。

表 3- 27 芯片计数寄存器 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

3.4.23 NODE PLL 时钟配置寄存器 0

NODE PLL 时钟配置寄存器0，用于 NODE PLL 时钟参数配置。

寄存器地址：0x16000400。

表 3- 28 NODE PLL 时钟配置寄存器 0

位域	名称	访问	缺省值	描述
31	reserved	R0	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_i2s	RW	0x0	I2S选择PLL时钟输出配置，1代表选择PLL时钟输出
1	pll_sel_gmac	RW	0x0	GMAC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

3.4.24 NODE PLL 时钟配置寄存器 1

NODE PLL 时钟配置寄存器1，用于 NODE PLL 时钟参数配置。

寄存器地址：0x16000404。

表 3- 29 NODE 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_i2s	RW	0x0	I2S PLL分频系数配置：0~127
7	reserved	RO	0x0	—
6:0	odiv_gmac	RW	0x0	GMAC PLL分频系数配置：0~127

3.4.25 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：0x16000408。

表 3- 30 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_dev	RW	0x0	DEVICE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

3.4.26 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：0x1600040c。

表 3- 31 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
19:18	memdiv_mode	RW	0x1	DDR 输出时钟分频模式配置
17	soft_mc_rstn	RW	0x1	DDR控制器软复位控制
16	memdiv_rstn	RW	0x1	DDR分频复位控制
15	reserved	RO	0x0	—
14:8	odiv_dev	RW	0x0	DEVs 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

3.4.27 PIX PLL 时钟配置寄存器 0

PIX PLL 时钟配置寄存器0，用于 PIX PLL 时钟参数配置。

寄存器地址：0x16000410。

表 3- 32 PIX 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_pix	RW	0x0	PIX 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	reserved	RO	0x0	—
1	pll_sel_gmacbp	RW	0x0	GMAC-BACKUP选择PLL时钟输出配置，1代表选择PLL时钟输出
0	pll_sel_pix	RW	0x0	PIX显示选择PLL时钟输出配置，1代表选择PLL时钟输出

3.4.28 PIX PLL 时钟配置寄存器 1

PIX PLL 时钟配置寄存器1，用于 PIX PLL 时钟参数配置。

寄存器地址：0x16000414。

表 3- 33 PIX 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
19:7	reserved	RO	0x0	—
6:0	odiv_gmacbp	RW	0x0	GMAC-BACKIP PLL分频系数配置：0~127

3.4.29 设备时钟分频配置寄存器

设备时钟分频配置寄存器，分别对应两种分频模式（除 SDIO 分频配置外）：

(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： $f_{out}=f_{in}/(freqscale[2:0]+1)$ 。

寄存器地址：0x16000420。

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
31:28	reserved	RO	0xf	—
27:24	sdio_freqscale	RW	0x1	SDIO时钟分配系数（CLK/freqscale[27:24]）： 对应SDIO时钟输出分频系数：0~15（0/1均不分频）
23:20	i2s_freqscale	RW	0x7	i2s_freqscale[3]:I2S时钟分配模式 freq_mode; i2s_freqscale[2:0]:I2S时钟输出分频系数： 0~7
19:16	apb_freqscale	RW	0x7	apb_freqscale[3]:APB时钟分配模式 freq_mode; apb_freqscale[2:0]:APB时钟输出分频系数： 0~7

15:12	usb_freqscale	RW	0x7	usb_freqscale[3]:USB时钟分配模式 freq_mode; usb_freqscale[2:0]:USB时钟输出分频系数: 0~7
11:8	boot_freqscale	RW	0x7	boot_freqscale[3]:BOOT时钟分配模式 freq_mode; boot_freqscale[2:0]:BOOT时钟输出分频系 数: 0~7 注: BOOT时钟模块包括: Confbus、SPI0/1、 LocalIO控制模块时钟
7:4	pix_freqscale	RW	0x7	pix_freqscale[3]:显示时钟分配模式 freq_mode; pix_freqscale[2:0]:显示时钟输出分频系 数: 0~7
3:0	node_freqscale	RW	0x7	node_freqscale[3]:NODE时钟分配模式 freq_mode; node_freqscale[2:0]:NODE时钟输出分频系 数: 0~7

3.4.30 设备时钟输出使能配置寄存器

设备时钟输出使能配置寄存器,

寄存器地址: 0x16000424。

表 3- 35 设备时钟输出使能配置寄存器

位域	名称	访问	缺省值	描述
31:9	reserved	RO	0x7ffffff	—
8	gmac_clkssel	RW	0x1	gmac模块时钟源选择配置: 1: 选择SOC-PLL输出时钟; 0: 选择GMACBP-PLL备份输出时钟
7	Reserved	RO	0x1	—
6	pix_clken	RW	0x1	pixel显示模块时钟输出使能配置: 1: 时钟输出使能; 0: 时钟输出关闭
5	i2s_clken	RW	0x1	i2s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	sdio_clken	RW	0x1	sdio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	apb_clken	RW	0x1	apb设备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	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
1	boot_clken	RW	0x1	boot模块时钟输出使能配置 (Confbus/SPI0/SPI1/LocalIO模块时钟): 1: 时钟输出使能; 0: 时钟输出关闭

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

3.4.31 GPIO 复用配置寄存器 0

GPIO0~15复用配置寄存器。

寄存器地址：0x16000490。

表 3- 36 GPIO 复用配置寄存器 0

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。

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。

3.4.32 GPIO 复用配置寄存器 1

GPIO16~31复用配置寄存器。

寄存器地址：0x16000494。

表 3- 37 GPIO 复用配置寄存器 1

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。
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。

3.4.33 GPIO 复用配置寄存器 2

GPIO32~47复用配置寄存器。

寄存器地址：0x16000498。

表 3- 38 GPIO 复用配置寄存器 2

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GPIO38; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
11:10	GPIO37_MUX	RW	0x0	GPIO37引脚复用配置: 00: 复用为GPIO37; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
9:8	GPIO36_MUX	RW	0x0	GPIO36引脚复用配置: 00: 复用为GPIO36; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
7:6	GPIO35_MUX	RW	0x0	GPIO35引脚复用配置: 00: 复用为GPIO35; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
5:4	GPIO34_MUX	RW	0x0	GPIO34引脚复用配置: 00: 复用为GPIO34; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
3:2	GPIO33_MUX	RW	0x0	GPIO33引脚复用配置: 00: 复用为GPIO33; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
1:0	GPIO32_MUX	RW	0x0	GPIO32引脚复用配置: 00: 复用为GPIO32; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。

3.4.34 GPIO 复用配置寄存器 3

GPIO48~63复用配置寄存器。

寄存器地址: 0x1600049c。

表 3- 39 GPIO 复用配置寄存器 3

位域	名称	访问	缺省值	描述
31:30	GPIO63_MUX	RW	0x0	GPIO63引脚复用配置: 00: 复用为GPIO63; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
29:28	GPIO62_MUX	RW	0x0	GPIO62引脚复用配置: 00: 复用为GPIO62; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
27:26	GPIO61_MUX	RW	0x0	GPIO61引脚复用配置: 00: 复用为GPIO61; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
25:24	GPIO60_MUX	RW	0x0	GPIO60引脚复用配置: 00: 复用为GPIO60; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
23:22	GPIO59_MUX	RW	0x0	GPIO59引脚复用配置: 00: 复用为GPIO59; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
21:20	GPIO58_MUX	RW	0x0	GPIO58引脚复用配置: 00: 复用为GPIO58; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
19:18	GPIO57_MUX	RW	0x0	GPIO57引脚复用配置: 00: 复用为GPIO57; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。

17:16	GPI056_MUX	RW	0x0	GPI056引脚复用配置： 00: 复用为GPI056；01: 第一复用； 10: 第二复用； 11: 引脚主功能。
15:14	GPI055_MUX	RW	0x0	GPI055引脚复用配置： 00: 复用为GPI055；01: 第一复用； 10: 第二复用； 11: 引脚主功能。
13:12	GPI054_MUX	RW	0x0	GPI054引脚复用配置： 00: 复用为GPI054；01: 第一复用； 10: 第二复用； 11: 引脚主功能。
11:10	GPI053_MUX	RW	0x0	GPI053引脚复用配置： 00: 复用为GPI053；01: 第一复用； 10: 第二复用； 11: 引脚主功能。
9:8	GPI052_MUX	RW	0x0	GPI052引脚复用配置： 00: 复用为GPI052；01: 第一复用； 10: 第二复用； 11: 引脚主功能。
7:6	GPI051_MUX	RW	0x0	GPI051引脚复用配置： 00: 复用为GPI051；01: 第一复用； 10: 第二复用； 11: 引脚主功能。
5:4	GPI050_MUX	RW	0x0	GPI050引脚复用配置： 00: 复用为GPI050；01: 第一复用； 10: 第二复用； 11: 引脚主功能。
3:2	GPI049_MUX	RW	0x0	GPI049引脚复用配置： 00: 复用为GPI049；01: 第一复用； 10: 第二复用； 11: 引脚主功能。
1:0	GPI048_MUX	RW	0x0	GPI048引脚复用配置： 00: 复用为GPI048；01: 第一复用； 10: 第二复用； 11: 引脚主功能。

3.4.35 GPIO 复用配置寄存器 4

GPIO64~79复用配置寄存器。

寄存器地址：0x160004a0。

表 3- 40 GPIO 复用配置寄存器 4

位域	名称	访问	缺省值	描述
31:30	GPI079_MUX	RW	0x0	GPI079引脚复用配置： 00: 复用为GPI079；01: 第一复用； 10: 第二复用； 11: 引脚主功能。
29:28	GPI078_MUX	RW	0x0	GPI078引脚复用配置： 00: 复用为GPI078；01: 第一复用； 10: 第二复用； 11: 引脚主功能。
27:26	GPI077_MUX	RW	0x0	GPI077引脚复用配置： 00: 复用为GPI077；01: 第一复用； 10: 第二复用； 11: 引脚主功能。
25:24	GPI076_MUX	RW	0x0	GPI076引脚复用配置： 00: 复用为GPI076；01: 第一复用； 10: 第二复用； 11: 引脚主功能。
23:22	GPI075_MUX	RW	0x0	GPI075引脚复用配置： 00: 复用为GPI075；01: 第一复用； 10: 第二复用； 11: 引脚主功能。
21:20	GPI074_MUX	RW	0x0	GPI074引脚复用配置：

				00: 复用为GPIO74; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
19:18	GPIO73_MUX	RW	0x0	GPIO73引脚复用配置: 00: 复用为GPIO73; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
17:16	GPIO72_MUX	RW	0x0	GPIO72引脚复用配置: 00: 复用为GPIO72; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
15:14	GPIO71_MUX	RW	0x0	GPIO71引脚复用配置: 00: 复用为GPIO71; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
13:12	GPIO70_MUX	RW	0x0	GPIO70引脚复用配置: 00: 复用为GPIO70; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
11:10	GPIO69_MUX	RW	0x0	GPIO69引脚复用配置: 00: 复用为GPIO69; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
9:8	GPIO68_MUX	RW	0x0	GPIO68引脚复用配置: 00: 复用为GPIO68; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
7:6	GPIO67_MUX	RW	0x0	GPIO67引脚复用配置: 00: 复用为GPIO67; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
5:4	GPIO66_MUX	RW	0x0	GPIO66引脚复用配置: 00: 复用为GPIO66; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
3:2	GPIO65_MUX	RW	0x0	GPIO65引脚复用配置: 00: 复用为GPIO65; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
1:0	GPIO64_MUX	RW	0x0	GPIO64引脚复用配置: 00: 复用为GPIO64; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。

3.4.36 GPIO 复用配置寄存器 5

GPIO80~95复用配置寄存器。

寄存器地址: 0x160004a4。

表 3- 41 GPIO 复用配置寄存器 5

位域	名称	访问	缺省值	描述
31:30	GPIO95_MUX	RW	0x0	GPIO95引脚复用配置: 00: 复用为GPIO95; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
29:28	GPIO94_MUX	RW	0x0	GPIO94引脚复用配置: 00: 复用为GPIO94; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
27:26	GPIO93_MUX	RW	0x0	GPIO93引脚复用配置: 00: 复用为GPIO93; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
25:24	GPIO92_MUX	RW	0x0	GPIO92引脚复用配置: 00: 复用为GPIO92; 01: 第一复用;

				10: 第二复用; 11: 引脚主功能。
23:22	GPI091_MUX	RW	0x0	GPI091引脚复用配置: 00: 复用为GPI091; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
21:20	GPI090_MUX	RW	0x0	GPI090引脚复用配置: 00: 复用为GPI090; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
19:18	GPI089_MUX	RW	0x0	GPI089引脚复用配置: 00: 复用为GPI089; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
17:16	GPI088_MUX	RW	0x0	GPI088引脚复用配置: 00: 复用为GPI088; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
15:14	GPI087_MUX	RW	0x0	GPI087引脚复用配置: 00: 复用为GPI087; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
13:12	GPI086_MUX	RW	0x0	GPI086引脚复用配置: 00: 复用为GPI086; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
11:10	GPI085_MUX	RW	0x0	GPI085引脚复用配置: 00: 复用为GPI085; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
9:8	GPI084_MUX	RW	0x0	GPI084引脚复用配置: 00: 复用为GPI084; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
7:6	GPI083_MUX	RW	0x0	GPI083引脚复用配置: 00: 复用为GPI083; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
5:4	GPI082_MUX	RW	0x0	GPI082引脚复用配置: 00: 复用为GPI082; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
3:2	GPI081_MUX	RW	0x0	GPI081引脚复用配置: 00: 复用为GPI081; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。
1:0	GPI080_MUX	RW	0x0	GPI080引脚复用配置: 00: 复用为GPI080; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。

3.4.37 GPIO 复用配置寄存器 6

GPIO96~111复用配置寄存器。

寄存器地址: 0x160004a8。

表 3- 42 GPIO 复用配置寄存器 6

位域	名称	访问	缺省值	描述
31:20	reserved	RO	0x0	—
19:18	GPI0105_MUX	RW	0x0	GPI0105引脚复用配置: 00: 复用为GPI0105; 01: 第一复用; 10: 第二复用; 11: 引脚主功能。

17:16	GPI0104_MUX	RW	0x0	GPI0104引脚复用配置： 00：复用为GPI0104；01：第一复用； 10：第二复用； 11：引脚主功能。
15:14	GPI0103_MUX	RW	0x0	GPI0103引脚复用配置： 00：复用为GPI0103；01：第一复用； 10：第二复用； 11：引脚主功能。
13:12	GPI0102_MUX	RW	0x0	GPI0102引脚复用配置： 00：复用为GPI0102；01：第一复用； 10：第二复用； 11：引脚主功能。
11:10	GPI0101_MUX	RW	0x0	GPI0101引脚复用配置： 00：复用为GPI0101；01：第一复用； 10：第二复用； 11：引脚主功能。
9:8	GPI0100_MUX	RW	0x0	GPI0100引脚复用配置： 00：复用为GPI0100；01：第一复用； 10：第二复用； 11：引脚主功能。
7:6	GPI099_MUX	RW	0x0	GPI099引脚复用配置： 00：复用为GPI099；01：第一复用； 10：第二复用； 11：引脚主功能。
5:4	GPI098_MUX	RW	0x0	GPI098引脚复用配置： 00：复用为GPI098；01：第一复用； 10：第二复用； 11：引脚主功能。
3:2	GPI097_MUX	RW	0x0	GPI097引脚复用配置： 00：复用为GPI097；01：第一复用； 10：第二复用； 11：引脚主功能。
1:0	GPI096_MUX	RW	0x0	GPI096引脚复用配置： 00：复用为GPI096；01：第一复用； 10：第二复用； 11：引脚主功能。

3.4.38 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，可考 USB-PHY 手册寄存器描述。
寄存器地址：0x16000500。

3.4.39 USB PHY 配置寄存器 1

USB PHY 配置寄存器0，配置 USB 接口1的电气特性，可考 USB-PHY 手册寄存器描述。
寄存器地址：0x16000504。

3.4.40 USB PHY 配置寄存器 2

USB PHY 配置寄存器0，配置 USB 接口2的电气特性。
寄存器地址：0x16000508。

表 3- 43 USB PHY 配置寄存器 2

位域	名称	访问	缺省值	描述
31	otg_suspend_config	R/W	0	OTG端口挂起软件配置位，高有效
30	usb_suspend_config	R/W	0	USB端口挂起软件配置位，高有效

29	otg_suspend_soften	R/W	1	OTG端口挂起软件配置使能位，高有效
28	usb_suspend_soften	R/W	1	USB端口挂起软件配置使能位，高有效
27	usb_phy_por	R/W	0	USB端口复位控制位，高有效
26:19	Reserved	—	0	—
18	fs_data_mod	R/W	0	FS数据模式
17	otg_pulldown_soften	R/W	0	OTG端口下拉模式软件配置使能 0-关闭； 1-使能。
16	otg_opmode_soften	R/W	0	OTG端口操作模式软件配置使能 0-关闭； 1-使能。
15: 14	otg_opmode0	R/W	0	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	—	0	—
11	dppulldown0	R/W	0	USB端口dp端口下拉电阻使能 1: D+使能 0: D+关闭
10	dmpulldown0	R/W	0	USB端口dm端口下拉电阻使能 1: D-使能 0: D-关闭
9	dppulldown0	R/W	0	OTG端口dp端口下拉电阻使能 1: D+使能 0: D+关闭
8	dmpulldown0	R/W	0	OTG端口dm端口下拉电阻使能 1: D-使能 0: D-关闭
7:4	Reserved	—	0	—
3	usb_phy_clksel	R/W	1	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	—	0	—
1	usb_utmi_resetn	R/W	0	USB端口复位控制位： 0- 复位有效； 1- 复位撤销。
0	otg_utmi_resetn	R/W	0	OTG端口复位控制位： 0- 复位有效； 1- 复位撤销。

3.5 中断配置及路由

龙芯2K0300芯片中断分为传统、扩展中断方式。

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。

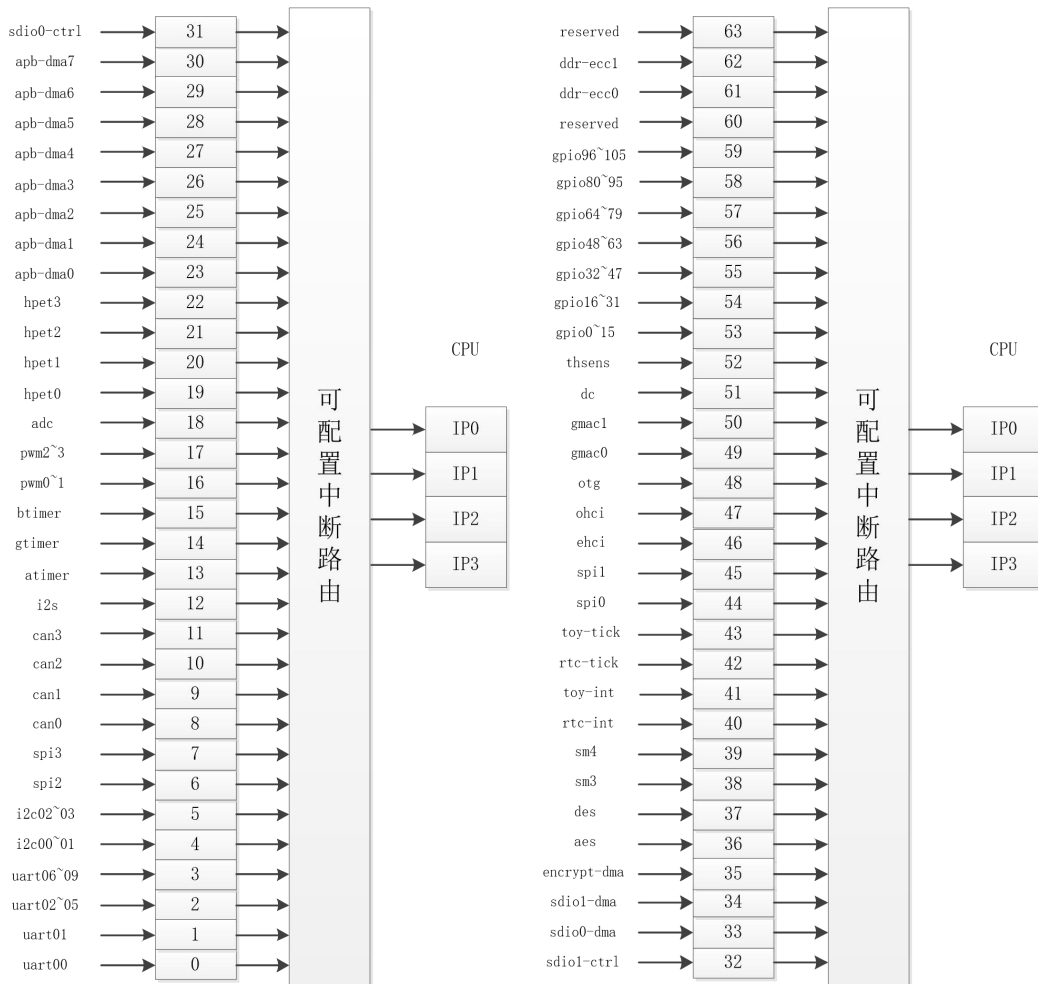


图3- 2 龙芯 2K0300 传统中断路由示意图

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 3-46。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- 44 IO 传统中断寄存器列表

地址	名称	描述
0x16001040	CORE_INTISR0	路由给CORE的低32位中断状态
0x16001044	INTISR0	低32位中断状态寄存器
0x16001048	CORE_INTISR1	路由给CORE的高32位中断状态
0x1600104c	INTISR1	高32位中断状态寄存器
0x16001400	ENTRY0_0	8位中断路由寄存器[0--7]
0x16001408	ENTRY8_0	8位中断路由寄存器[8--15]
0x16001410	ENTRY16_0	8位中断路由寄存器[16--23]
0x16001418	ENTRY24_0	8位中断路由寄存器[24--31]
0x16001420	INTISR_0	低32位中断状态寄存器
0x16001424	INTIEN_0	低32位中断使能状态寄存器
0x16001428	INTSET_0	低32位设置使能寄存器
0x1600142c	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
0x16001430	INTPOL_0	低32位极性设置寄存器(电平中断)
0x16001434	INTEDGE_0	低32位触发方式寄存器（1：脉冲触发；0：电平触发）
0x16001440	ENTRY0_1	8位中断路由寄存器[32--39]
0x16001448	ENTRY8_1	8位中断路由寄存器[40--47]
0x16001450	ENTRY16_1	8位中断路由寄存器[48--55]
0x16001458	ENTRY24_1	8位中断路由寄存器[56--63]
0x16001460	INTISR_1	高32位中断状态寄存器
0x16001464	INTIEN_1	高32位中断使能状态寄存器
0x16001468	INTSET_1	高32位设置使能寄存器
0x1600146c	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
0x16001470	INTPOL_1	高32位极性设置寄存器(电平中断)
0x16001474	INTEDGE_1	高32位触发方式寄存器（1：脉冲触发；0：电平触发）
0x16001500	Thsens_int_ctrl_Hi0	温度传感器高温中断控制寄存器0
0x16001504	Thsens_int_ctrl_Hi1	温度传感器高温中断控制寄存器1
0x16001508	Thsens_int_ctrl_Lo0	温度传感器低温中断控制寄存器0
0x1600150c	Thsens_int_ctrl_Lo1	温度传感器低温中断控制寄存器1
0x16001510	Thsens_int_status/c1r	温度传感器中断状态/清除寄存器
0x16001514	Thsens_value	温度传感器测量值(仅低11位有效位)，计算公式为： $Tval=Thsens_val[10:0]*0.57-394.7$
0x16001518	Thsens_cfg	温度传感器配置寄存器
0x16001520	Thsens_scale_hi0	温度传感器高阈值配置寄存器0

0x16001524	Thsens_scale_hil	温度传感器高阈值配置寄存器1
------------	------------------	----------------

龙芯2K0300芯片除了支持上述传统IO中断方式外，还增加了扩展IO中断方式，即：将2K0300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。

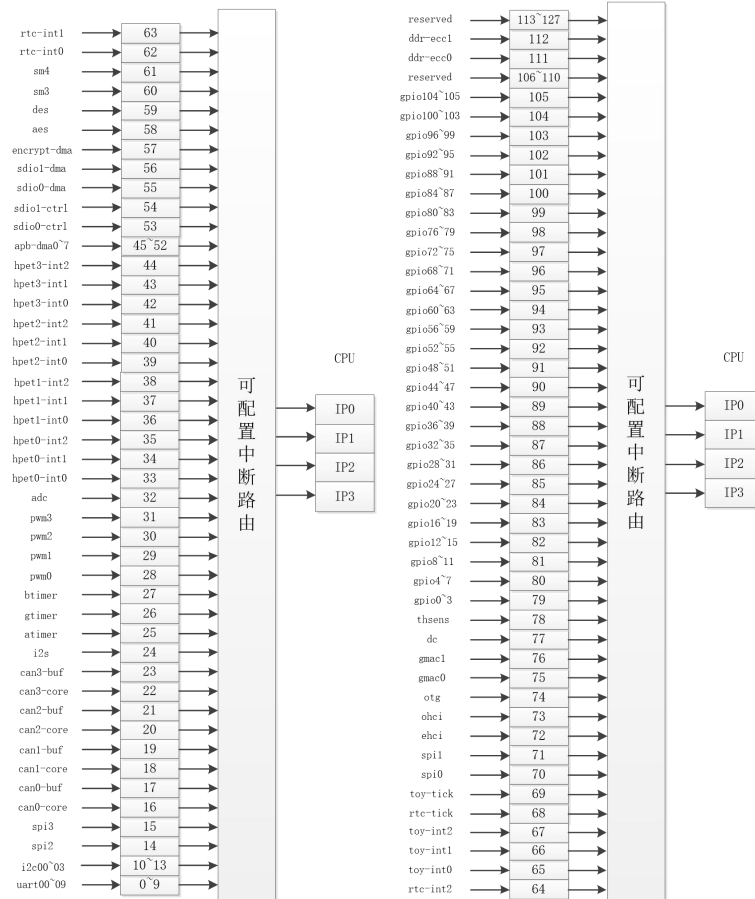


图3- 3 龙芯 2K0300 扩展 IO 中断路由示意图

龙芯 2K0300 芯片在使用扩展 IO 中断前，需要使能“芯片通用配置寄存器 0”中的扩展 IO 中断使能位。该寄存器配置地址为 0x16000100，对应配置位如下表：

位域	名称	访问	缺省值	描述
19	extioi_enable	RW	0x0	扩展中断使能控制位： 1：打开扩展中断(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 3-48。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- 45 系统 IO 扩展中断寄存器列表

地址	名称	描述
0x16001148	EXTIOI_ACK	扩展中断设备反馈寄存器
0x160014c0	EXTIOI_MAP	扩展中断设备路由寄存器
0x16001600	EXTIOI_IEN0	扩展中断设备使能寄存器0
0x16001604	EXTIOI_IEN1	扩展中断设备使能寄存器1
0x16001608	EXTIOI_IEN2	扩展中断设备使能寄存器2
0x1600160c	EXTIOI_IEN3	扩展中断设备使能寄存器3
0x16001640	EXTIOI_POL0	扩展中断电平配置寄存器0
0x16001644	EXTIOI_POL1	扩展中断电平配置寄存器1
0x16001648	EXTIOI_POL2	扩展中断电平配置寄存器2
0x1600164c	EXTIOI_POL3	扩展中断电平配置寄存器3
0x16001700	EXTIOI_ISR0	扩展中断状态寄存器0
0x16001704	EXTIOI_ISR1	扩展中断状态寄存器1
0x16001708	EXTIOI_ISR2	扩展中断状态寄存器2
0x1600170c	EXTIOI_ISR3	扩展中断状态寄存器3
0x16001800	EXTIOI_CORE_ISR0	路由至CORE扩展中断状态寄存器0
0x16001804	EXTIOI_CORE_ISR1	路由至CORE扩展中断状态寄存器1
0x16001808	EXTIOI_CORE_ISR2	路由至CORE扩展中断状态寄存器2
0x1600180c	EXTIOI_CORE_ISR3	路由至CORE扩展中断状态寄存器3

3.5.1 中断触发类型

对于龙芯 2K0300 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。

3.5.2 中断相关寄存器描述

表 3- 46 传统中断控制寄存器属性

位域	访问属性/缺省值						中断源
	Intedg	Inten	Intenset	Intencl	Intpol	Intentry	
0	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	UART00

位域	访问属性/缺省值						中断源
1	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	UART01
2	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	UART02~05
3	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	UART06~09
4	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	I2C0~1
5	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	I2C2~3
6	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	SPI2
7	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	SPI3
8	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	CAN0
9	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	CAN1
10	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	CAN2
11	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	CAN3
12	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	I2S
13	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	ATIMER
14	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	GTIMER
15	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	BTIMER
16	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	PWM0/1
17	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	PWM2/3
18	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	ADC
19	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	HPETO
20	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	HPET1
21	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	HPET2
22	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	HPET3
23	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	APB-DMA0
24	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	APB-DMA1
25	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	APB-DMA2
26	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	APB-DMA3
27	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	APB-DMA4
28	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	APB-DMA5
29	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	APB-DMA6
30	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	APB-DMA7
31	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	SDIO0-CTRL
32	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	SDIO1-CTRL
33	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	SDIO0-DMA
34	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	SDIO1-DMA
35	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	ENCRYPT-DMA
36	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	AES

位域	访问属性/缺省值						中断源
37	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	DES
38	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	SM3
39	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	SM4
40	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	RTC-INT
41	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	TOY-INT
42	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	RTC-TICK
43	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	TOY-TICK
44	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	SPIO
45	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	SPI1
46	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	ECHI
47	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	OHCI
48	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	OTG
49	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	GMACO
50	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	GMAC1
51	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	DC
52	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	THSENS
53	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	GPI00~15
54	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	GPI016~31
55	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	GPI032~47
56	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	GPI048~63
57	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	GPI064~79
58	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	GPI080~95
59	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	GPI096~105
60	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	Reserved
61	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	DDR-ECC0
62	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	DDR-ECC1
63	RW / 0	R / 0	W / 0	W / 0	RW / 0	RW / 0	-

表 3- 47 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	0x16001420	RO	NA	低 32 位中断状态寄存器
Inten_0	0x16001424	RO	NA	低 32 位中断使能状态寄存器
Intenset_0	0x16001428	WO	NA	低 32 位设置使能寄存器
Intencclr_0	0x1600142c	WO	NA	低 32 位清除使能寄存器和脉冲触发的中断
Intpol_0	0x16001430	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	0x16001434	WO	0x0	低 32 位触发方式寄存器（1：脉冲触发；0：电平触发）

名称	地址偏移	访问属性	缺省值	描述
Intisr_1	0x16001460	RO	NA	高 32 位中断状态寄存器
Inten_1	0x16001464	RO	NA	高 32 位中断使能状态寄存器
Intenset_1	0x16001468	WO	NA	高 32 位设置使能寄存器
Intenclr_1	0x1600146c	WO	NA	高 32 位清除使能寄存器和脉冲触发的中断
Intpol_1	0x16001470	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	0x16001474	WO	0x0	高 32 位触发方式寄存器（1：脉冲触发；0：电平触发）
CORE_IPISR	0x16001000	RO	NA	处理器核的 IPI_Status 寄存器
CORE_IPIEN	0x16001004	RW	0x0	处理器核的 IPI_Enalbe 寄存器
CORE_IPISET	0x16001008	WO	NA	处理器核的 IPI_Set 寄存器
CORE_IPI_CLR	0x1600100c	WO	NA	处理器核的 IPI_Clear 寄存器
CORE_INTISRO	0x16001040	RO	NA	路由给 CORE 的低 32 位中断状态
CORE_INTISR1	0x16001048	RO	NA	路由给 CORE 的高 32 位中断状态

表 3- 48 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	UART0	64	R / 0	W / 0	RW / 0	RTC-INT2
1	R / 0	W / 0	RW / 0	UART1	65	R / 0	W / 0	RW / 0	TOY-INT0
2	R / 0	W / 0	RW / 0	UART2	66	R / 0	W / 0	RW / 0	TOY-INT1
3	R / 0	W / 0	RW / 0	UART3	67	R / 0	W / 0	RW / 0	TOY-INT2
4	R / 0	W / 0	RW / 0	UART4	68	R / 0	W / 0	RW / 0	RTC-TICK
5	R / 0	W / 0	RW / 0	UART5	69	R / 0	W / 0	RW / 0	TOY-TICK
6	R / 0	W / 0	RW / 0	UART6	70	R / 0	W / 0	RW / 0	SPI0
7	R / 0	W / 0	RW / 0	UART7	71	R / 0	W / 0	RW / 0	SPI1
8	R / 0	W / 0	RW / 0	UART8	72	R / 0	W / 0	RW / 0	ECHI
9	R / 0	W / 0	RW / 0	UART9	73	R / 0	W / 0	RW / 0	OHCI
10	R / 0	W / 0	RW / 0	I2C0	74	R / 0	W / 0	RW / 0	OTG
11	R / 0	W / 0	RW / 0	I2C1	75	R / 0	W / 0	RW / 0	GMAC0
12	R / 0	W / 0	RW / 0	I2C2	76	R / 0	W / 0	RW / 0	GMAC1
13	R / 0	W / 0	RW / 0	I2C3	77	R / 0	W / 0	RW / 0	DC
14	R / 0	W / 0	RW / 0	SPI2	78	R / 0	W / 0	RW / 0	THSENS
15	R / 0	W / 0	RW / 0	SPI3	79	R / 0	W / 0	RW / 0	GPI00~3
16	R / 0	W / 0	RW / 0	CAN0-CORE	80	R / 0	W / 0	RW / 0	GPI04~7
17	R / 0	W / 0	RW / 0	CAN0-BUF	81	R / 0	W / 0	RW / 0	GPI08~11
18	R / 0	W / 0	RW / 0	CAN1-CORE	82	R / 0	W / 0	RW / 0	GPI012~15
19	R / 0	W / 0	RW / 0	CAN1-BUF	83	R / 0	W / 0	RW / 0	GPI016~19

20	R / 0	W / 0	RW / 0	CAN2-CORE	84	R / 0	W / 0	RW / 0	GPI020~23
21	R / 0	W / 0	RW / 0	CAN2-BUF	85	R / 0	W / 0	RW / 0	GPI024~27
22	R / 0	W / 0	RW / 0	CAN3-CORE	86	R / 0	W / 0	RW / 0	GPI028~31
23	R / 0	W / 0	RW / 0	CAN3-BUF	87	R / 0	W / 0	RW / 0	GPI032~35
24	R / 0	W / 0	RW / 0	I2S	88	R / 0	W / 0	RW / 0	GPI036~39
25	R / 0	W / 0	RW / 0	ATIMER	89	R / 0	W / 0	RW / 0	GPI040~43
26	R / 0	W / 0	RW / 0	GTIMER	90	R / 0	W / 0	RW / 0	GPI044~47
27	R / 0	W / 0	RW / 0	BTIMER	91	R / 0	W / 0	RW / 0	GPI048~51
28	R / 0	W / 0	RW / 0	PWM0	92	R / 0	W / 0	RW / 0	GPI052~55
29	R / 0	W / 0	RW / 0	PWM1	93	R / 0	W / 0	RW / 0	GPI056~59
30	R / 0	W / 0	RW / 0	PWM2	94	R / 0	W / 0	RW / 0	GPI060~63
31	R / 0	W / 0	RW / 0	PWM3	95	R / 0	W / 0	RW / 0	GPI064~67
32	R / 0	W / 0	RW / 0	ADC	96	R / 0	W / 0	RW / 0	GPI068~71
33	R / 0	W / 0	RW / 0	HPETO-INT0	97	R / 0	W / 0	RW / 0	GPI072~75
34	R / 0	W / 0	RW / 0	HPETO-INT1	98	R / 0	W / 0	RW / 0	GPI076~79
35	R / 0	W / 0	RW / 0	HPETO-INT2	99	R / 0	W / 0	RW / 0	GPI080~83
36	R / 0	W / 0	RW / 0	HPET1-INT0	100	R / 0	W / 0	RW / 0	GPI084~87
37	R / 0	W / 0	RW / 0	HPET1-INT1	101	R / 0	W / 0	RW / 0	GPI088~91
38	R / 0	W / 0	RW / 0	HPET1-INT2	102	R / 0	W / 0	RW / 0	GPI092~95
39	R / 0	W / 0	RW / 0	HPET2-INT0	103	R / 0	W / 0	RW / 0	GPI096~99
40	R / 0	W / 0	RW / 0	HPET2-INT1	104	R / 0	W / 0	RW / 0	GPI0100~103
41	R / 0	W / 0	RW / 0	HPET2-INT2	105	R / 0	W / 0	RW / 0	GPI0104~105
42	R / 0	W / 0	RW / 0	HPET3-INT0	106	R / 0	W / 0	RW / 0	-
43	R / 0	W / 0	RW / 0	HPET3-INT1	107	R / 0	W / 0	RW / 0	-
44	R / 0	W / 0	RW / 0	HPET3-INT2	108	R / 0	W / 0	RW / 0	-
45	R / 0	W / 0	RW / 0	APB-DMA0	109	R / 0	W / 0	RW / 0	-
46	R / 0	W / 0	RW / 0	APB-DMA1	110	R / 0	W / 0	RW / 0	-
47	R / 0	W / 0	RW / 0	APB-DMA2	111	R / 0	W / 0	RW / 0	DDR-ECC0
48	R / 0	W / 0	RW / 0	APB-DMA3	112	R / 0	W / 0	RW / 0	DDR-ECC1
49	R / 0	W / 0	RW / 0	APB-DMA4	113	R / 0	W / 0	RW / 0	-
50	R / 0	W / 0	RW / 0	APB-DMA5	114	R / 0	W / 0	RW / 0	-
51	R / 0	W / 0	RW / 0	APB-DMA6	115	R / 0	W / 0	RW / 0	-
52	R / 0	W / 0	RW / 0	APB-DMA7	116	R / 0	W / 0	RW / 0	-
53	R / 0	W / 0	RW / 0	SDI00-CTRL	117	R / 0	W / 0	RW / 0	-
54	R / 0	W / 0	RW / 0	SDI01-CTRL	118	R / 0	W / 0	RW / 0	-
55	R / 0	W / 0	RW / 0	SDI00-DMA	119	R / 0	W / 0	RW / 0	-
56	R / 0	W / 0	RW / 0	SDI01-DMA	120	R / 0	W / 0	RW / 0	-

57	R / 0	W / 0	RW / 0	ENCRYPT-DMA	121	R / 0	W / 0	RW / 0	-
58	R / 0	W / 0	RW / 0	AES	122	R / 0	W / 0	RW / 0	-
59	R / 0	W / 0	RW / 0	DES	123	R / 0	W / 0	RW / 0	-
60	R / 0	W / 0	RW / 0	SM3	124	R / 0	W / 0	RW / 0	-
61	R / 0	W / 0	RW / 0	SM4	125	R / 0	W / 0	RW / 0	-
62	R / 0	W / 0	RW / 0	RTC-INT0	126	R / 0	W / 0	RW / 0	-
63	R / 0	W / 0	RW / 0	RTC-INT1	127	R / 0	W / 0	RW / 0	-

3.5.3 中断路由寄存器描述

龙芯 2K0300 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 IP2 到 IP5。芯片传统中断中 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- 49 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: LA264 处理器核对应 0 中断号； 0010: LA264 处理器核对应 1 中断号； 0100: LA264 处理器核对应 2 中断号； 1000: LA264 处理器核对应 3 中断号。

表 3- 50 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	0x16001400	UART00	Entry32	0x16001440	SDIO1-CTRL
Entry1	0x16001401	UART01	Entry33	0x16001441	SDIO0-DMA
Entry2	0x16001402	UART02~05	Entry34	0x16001442	SDIO1-DMA
Entry3	0x16001403	UART06~09	Entry35	0x16001443	ENCRYPT-DMA
Entry4	0x16001404	I2C0~1	Entry36	0x16001444	AES
Entry5	0x16001405	I2C2~3	Entry37	0x16001445	DES
Entry6	0x16001406	SPI2	Entry38	0x16001446	SM3
Entry7	0x16001407	SPI3	Entry39	0x16001447	SM4
Entry8	0x16001408	CAN0	Entry40	0x16001448	RTC-INT
Entry9	0x16001409	CAN1	Entry41	0x16001449	TOY-INT
Entry10	0x1600140a	CAN2	Entry42	0x1600144a	RTC-TICK
Entry11	0x1600140b	CAN3	Entry43	0x1600144b	TOY-TICK

名称	地址偏移	描述	名称	地址偏移	描述
Entry12	0x1600140c	I2S	Entry44	0x1600144c	SPI0
Entry13	0x1600140d	ATIMER	Entry45	0x1600144d	SPI1
Entry14	0x1600140e	GTIMER	Entry46	0x1600144e	ECHI
Entry15	0x1600140f	BTIMER	Entry47	0x1600144f	OHCI
Entry16	0x16001410	PWM0/1	Entry48	0x16001450	OTG
Entry17	0x16001411	PWM2/3	Entry49	0x16001451	GMACO
Entry18	0x16001412	ADC	Entry50	0x16001452	GMAC1
Entry19	0x16001413	HPETO	Entry51	0x16001453	DC
Entry20	0x16001414	HPET1	Entry52	0x16001454	THSENS
Entry21	0x16001415	HPET2	Entry53	0x16001455	GPIO0~15
Entry22	0x16001416	HPET3	Entry54	0x16001456	GPIO16~31
Entry23	0x16001417	APB-DMA0	Entry55	0x16001457	GPIO32~47
Entry24	0x16001418	APB-DMA1	Entry56	0x16001458	GPIO48~63
Entry25	0x16001419	APB-DMA2	Entry57	0x16001459	GPIO64~79
Entry26	0x1600141a	APB-DMA3	Entry58	0x1600145a	GPIO80~95
Entry27	0x1600141b	APB-DMA4	Entry59	0x1600145b	GPIO96~111
Entry28	0x1600141c	APB-DMA5	Entry60	0x1600145c	GPIO112~127
Entry29	0x1600141d	APB-DMA6	Entry61	0x1600145d	DDR-ECC0
Entry30	0x1600141e	APB-DMA7	Entry62	0x1600145e	DDR-ECC1
Entry31	0x1600141f	SDIO0-CTRL	Entry63	0x1600145f	-

表 3- 51 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: LA264 处理器核对应 0 中断号； 0010: LA264 处理器核对应 1 中断号； 0100: LA264 处理器核对应 2 中断号； 1000: LA264 处理器核对应 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: LA264 处理器核对应 0 中断号； 0010: LA264 处理器核对应 1 中断号； 0100: LA264 处理器核对应 2 中断号； 1000: LA264 处理器核对应 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: LA264 处理器核对应 0 中断号； 0010: LA264 处理器核对应 1 中断号； 0100: LA264 处理器核对应 2 中断号； 1000: LA264 处理器核对应 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: LA264 处理器核对应 0 中断号； 0010: LA264 处理器核对应 1 中断号； 0100: LA264 处理器核对应 2 中断号； 1000: LA264 处理器核对应 3 中断号。

4 DDR4 控制器

龙芯 2K0300 处理器内部集成的 DDR4 SDRAM 内存控制器。

4.1 访问地址

DDR4 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 配置请参考芯片通用配置寄存器 3.4.1。功能概述：

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。

内存控制器具有如下特征：

- 接口上命令、读写数据全流水操作；
- 内存命令合并、排序提高整体带宽；
- 配置寄存器读写端口，可以修改内存设备的基本参数；
- 内建动态延迟补偿电路（DCC），用于数据的可靠发送和接收；
- 支持 DDR4 SDRAM，且参数配置支持 x8、x16 颗粒；
- 控制器与 PHY 频率比 1/2 ；
- 支持数据传输速率范围为 800Mbps-1600Mbps。

4.2 DDR4 控制器寄存器

由于系统中可能使用不同类型的 DDR4 SDRAM，因此，在系统上电复位以后，需要对 DDR4 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 DDR4 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。

内存处理器设计中，DDR4 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 DDR4 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_wrds_0	dll_wrds_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_bdy0_0[31:28]	w_bdy0_0[27:24]	w_bdy0_0[23:20]	w_bdy0_0[19:16]	w_bdy0_0[15:12]	w_bdy0_0[11:8]	w_bdy0_0[7:4]	w_bdy0_0[3:0]
0x0128		w_bdy0_0[59:56]	w_bdy0_0[55:52]	w_bdy0_0[51:48]	w_bdy0_0[47:44]	w_bdy0_0[43:40]	w_bdy0_0[39:36]	w_bdy0_0[35:32]
0x0130	w_bdy1_0[24:21]	w_bdy1_0[20:18]	w_bdy1_0[17:15]	w_bdy1_0[14:12]	w_bdy1_0[11:9]	w_bdy1_0[8:6]	w_bdy1_0[5:3]	w_bdy1_0[2:0]
0x0138								w_bdy1_0[27:26]
0x0140							rg_bdy0_0[7:4]	rg_bdy0_0[3:0]
0x0148								
0x0150	rdqsp_bdy0_0[31:28]	rdqsp_bdy0_0[27:24]	rdqsp_bdy0_0[23:20]	rdqsp_bdy0_0[19:16]	rdqsp_bdy0_0[15:12]	rdqsp_bdy0_0[11:8]	rdqsp_bdy0_0[7:4]	rdqsp_bdy0_0[3:0]
0x0158								rdqsp_bdy0_0[35:32]
0x0160	rdqsn_bdy0_0[31:28]	rdqsn_bdy0_0[27:24]	rdqsn_bdy0_0[23:20]	rdqsn_bdy0_0[19:16]	rdqsn_bdy0_0[15:12]	rdqsn_bdy0_0[11:8]	rdqsn_bdy0_0[7:4]	rdqsn_bdy0_0[3:0]
0x0168								rdqsn_bdy0_0[35:32]
0x0170	rdq_bdy0_0[24:21]	rdq_bdy0_0[20:18]	rdq_bdy0_0[17:15]	rdq_bdy0_0[14:12]	rdq_bdy0_0[11:9]	rdq_bdy0_0[8:6]	rdq_bdy0_0[5:3]	rdq_bdy0_0[2:0]
0x0178								rdq_bdy0_0[27:26]
0x0180					dll_1xdly_1	dll_1xgen_1	dll_wrds_1	dll_wrds_1
0x0188						dll_gate_1	dll_rdds_1_1	dll_rdds_0_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	rdqs_phase_1(RD)
0x01a0	w_bdy0_1[31:28]	w_bdy0_1[27:24]	w_bdy0_1[23:20]	w_bdy0_1[19:16]	w_bdy0_1[15:12]	w_bdy0_1[11:8]	w_bdy0_1[7:4]	w_bdy0_1[3:0]
0x01a8		w_bdy0_1[59:56]	w_bdy0_1[55:52]	w_bdy0_1[51:48]	w_bdy0_1[47:44]	w_bdy0_1[43:40]	w_bdy0_1[39:36]	w_bdy0_1[35:32]
0x01b0	w_bdy1_1[24:21]	w_bdy1_1[20:18]	w_bdy1_1[17:15]	w_bdy1_1[14:12]	w_bdy1_1[11:9]	w_bdy1_1[8:6]	w_bdy1_1[5:3]	w_bdy1_1[2:0]
0x01b8								w_bdy1_1[27:26]
0x01c0							rg_bdy1_1[7:4]	rg_bdy1_1[3:0]
0x01c8								
0x01d0	rdqsp_bdy1_1[31:28]	rdqsp_bdy1_1[27:24]	rdqsp_bdy1_1[23:20]	rdqsp_bdy1_1[19:16]	rdqsp_bdy1_1[15:12]	rdqsp_bdy1_1[11:8]	rdqsp_bdy1_1[7:4]	rdqsp_bdy1_1[3:0]
0x01d8								rdqsp_bdy1_1[35:32]
0x01e0	rdqsn_bdy1_1[31:28]	rdqsn_bdy1_1[27:24]	rdqsn_bdy1_1[23:20]	rdqsn_bdy1_1[19:16]	rdqsn_bdy1_1[15:12]	rdqsn_bdy1_1[11:8]	rdqsn_bdy1_1[7:4]	rdqsn_bdy1_1[3:0]
0x01e8								rdqsn_bdy1_1[35:32]
0x01f0	rdq_bdy1_1[24:21]	rdq_bdy1_1[20:18]	rdq_bdy1_1[17:15]	rdq_bdy1_1[14:12]	rdq_bdy1_1[11:9]	rdq_bdy1_1[8:6]	rdq_bdy1_1[5:3]	rdq_bdy1_1[2:0]
0x01f8								rdq_bdy1_1[27:26]
0x0200					dll_1xdly_2	dll_1xgen_2	dll_wrds_2	dll_wrds_2
0x0208						dll_gate_2	dll_rdds_1_2	dll_rdds_0_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	rdqs_phase_2(RD)
0x0220	w_bdy0_2[31:28]	w_bdy0_2[27:24]	w_bdy0_2[23:20]	w_bdy0_2[19:16]	w_bdy0_2[15:12]	w_bdy0_2[11:8]	w_bdy0_2[7:4]	w_bdy0_2[3:0]
0x0228		w_bdy0_2[59:56]	w_bdy0_2[55:52]	w_bdy0_2[51:48]	w_bdy0_2[47:44]	w_bdy0_2[43:40]	w_bdy0_2[39:36]	w_bdy0_2[35:32]
0x0230	w_bdy1_2[24:21]	w_bdy1_2[20:18]	w_bdy1_2[17:15]	w_bdy1_2[14:12]	w_bdy1_2[11:9]	w_bdy1_2[8:6]	w_bdy1_2[5:3]	w_bdy1_2[2:0]
0x0238								w_bdy1_2[27:26]
0x0240							rg_bdy2_2[7:4]	rg_bdy2_2[3:0]
0x0248								
0x0250	rdqsp_bdy2_2[31:28]	rdqsp_bdy2_2[27:24]	rdqsp_bdy2_2[23:20]	rdqsp_bdy2_2[19:16]	rdqsp_bdy2_2[15:12]	rdqsp_bdy2_2[11:8]	rdqsp_bdy2_2[7:4]	rdqsp_bdy2_2[3:0]
0x0258								rdqsp_bdy2_2[35:32]
0x0260	rdqsn_bdy2_2[31:28]	rdqsn_bdy2_2[27:24]	rdqsn_bdy2_2[23:20]	rdqsn_bdy2_2[19:16]	rdqsn_bdy2_2[15:12]	rdqsn_bdy2_2[11:8]	rdqsn_bdy2_2[7:4]	rdqsn_bdy2_2[3:0]
0x0268								rdqsn_bdy2_2[35:32]
0x0270	rdq_bdy2_2[24:21]	rdq_bdy2_2[20:18]	rdq_bdy2_2[17:15]	rdq_bdy2_2[14:12]	rdq_bdy2_2[11:9]	rdq_bdy2_2[8:6]	rdq_bdy2_2[5:3]	rdq_bdy2_2[2:0]
0x0278								rdq_bdy2_2[27:26]
0x0280					dll_1xdly_3	dll_1xgen_3	dll_wrds_3	dll_wrds_3
0x0288						dll_gate_3	dll_rdds_1_3	dll_rdds_0_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_wrdqs_4	dll_wrdq_4
0x0308						dll_gate_4	dll_rddqs1_4	dll_rddqs0_4
0x0310	rdodt_ctrl_4	rdgate_len_4	rdgate_mode_4	rdgate_ctrl_4			dqs_oe_ctrl_4	dq_oe_ctrl_4
0x0318					dly_2x_4		redge_sel_4	rddqs_phase_4(RD)
0x0320	w_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_wrdqs_5	dll_wrdq_5
0x0388						dll_gate_5	dll_rddqs1_5	dll_rddqs0_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	rddqs_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_wrdqs_6	dll_wrdq_6
0x0408						dll_gate_6	dll_rddqs1_6	dll_rddqs0_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								tRWGNTidle
0x1250							rfifo_age	
0x1258	prior_age3		prior_age2		prior_age1		prior_age0	
0x1260	retry_cnt(RD)					rbuffer_max(RD)	rdfifo_depth	stat_en
0x1268								
.....								
0x1280	aw_512_align		rd_before_wr	ecc_enable		int_vector(RD)	int_trigger(RD)	int_enable
0x1288								
0x1290						int_cnt_fatal(RD)	int_cnt_err(RD)	int_cnt
0x1298	ecc_cnt_cs_7(RD)	ecc_cnt_cs_6(RD)	ecc_cnt_cs_5(RD)	ecc_cnt_cs_4(RD)	ecc_cnt_cs_3(RD)	ecc_cnt_cs_2(RD)	ecc_cnt_cs_1(RD)	ecc_cnt_cs_0(RD)
0x12a0	ecc_data_dir(RD)	ecc_code_dir(RD)	ecc_code_256(RD)					ecc_code_64(RD)
0x12a8	ecc_addr(RD)							

0x12b0	ecc_data[63:0](RD)							
0x12b8	ecc_data[127:64] (RD)							
0x12c0	ecc_data[191:128] (RD)							
0x12c8	ecc_data[255:192] (RD)							
.....								
0x1300							ref_num	ref_sch_en
0x1308							Status_sref(RD)	srefresh_req
.....								
0x1340	hardware_pd_7	hardware_pd_6	hardware_pd_5	hardware_pd_4	hardware_pd_3	hardware_pd_2	hardware_pd_1	hardware_pd_0
0x1348	power_sta_7(RD)	power_sta_6(RD)	power_sta_5(RD)	power_sta_4(RD)	power_sta_3(RD)	power_sta_2(RD)	power_sta_1(RD)	power_sta_0(RD)
0x1350	selfref_age		slowpd_age		fastpd_age		active_age	
0x1358				power_up				Age_step
0x1360	tCONF_IDLE				tLPMC_IDLE			
.....								
0x1380								zq_overlap
0x1388								zq_stat_en
0x1390	zq_cnt_1(RD)				zq_cnt_0(RD)			
0x1398	zq_cnt_3(RD)				zq_cnt_2(RD)			
0x13a0	zq_cnt_5(RD)				zq_cnt_4(RD)			
0x13a8	zq_cnt_6(RD)				zq_cnt_6(RD)			
.....								
0x13c0					odt_wr_cs_map			
0x13c8							odt_wr_length	odt_wr_delay
0x13d0					odt_rd_cs_map			
0x13d8							odt_rd_length	odt_rd_delay
.....								
0x1400				tRESYNC_length	tRESYNC_delay	tRESYNC_shift	tRESYNC_max	tRESYNC_min
.....								
0x1440					pre_predict		tm_cmdq_num	burst_length
0x1448								ca_timing
0x1450						wr/rd_dbi_en	ca_par_en	crc_en
0x1458							tCA_PAR	tWR_CRC
0x1460	bit_map_7	bit_map_6	bit_map_5	bit_map_6	bit_map_3	bit_map_2	bit_map_1	bit_map_0
0x1468	bit_map_15	bit_map_14	bit_map_13	bit_map_12	bit_map_11	bit_map_10	bit_map_9	bit_map_8
0x1470							bit_map_17	bit_map_16
0x1478								bitmap_mirror
0x1480				alertn_misc(RD)			alertn_cnt	alertn_clr
0x1488	alertn_addr(RD)							
.....								
0x1500	win0_base							
0x1508	win1_base							
0x1510	win2_base							
0x1518	win3_base							
0x1520	win4_base							
0x1528	win5_base							
0x1530	win6_base							
0x1538	win7_base							

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

0x2168	wr_conflict(RD)							
0x2170	rtw_conflict(RD)							
0x2178	wtr_conflict(RD)							
0x2180	rd_conflict_last1(RD)							
0x2188	wr_conflict_last1(RD)							
0x2190	rtw_conflict_last1(RD)							
0x2198	wtr_conflict_last1(RD)							
0x21a0	wr_rd_turnaround(RD)							
0x21a8	cs_turnaround(RD)							
0x21b0	bg_conflict(RD)							
.....								
0x2300						sm_leveling		sm_init
0x2308								
0x2310		sm_rank_03		sm_rank_02		sm_rank_01		sm_rank_00
0x2318		sm_rank_07		sm_rank_06		sm_rank_05		sm_rank_04
0x2320		sm_rank_11		sm_rank_10		sm_rank_09		sm_rank_08
0x2328		sm_rank_15		sm_rank_14		sm_rank_13		sm_rank_12
0x2330		sm_rank_19		sm_rank_18		sm_rank_17		sm_rank_16
0x2338		sm_rank_23		sm_rank_22		sm_rank_21		sm_rank_20
0x2340		sm_rank_27		sm_rank_26		sm_rank_25		sm_rank_24
0x2348		sm_rank_31		sm_rank_30		sm_rank_29		sm_rank_28
.....								
TST								
0x3000						lpbk_mode	lpbk_start	lpbk_en
0x3008	lpbk_correct(RD)				lpbk_counter(RD)			lpbk_error(RD)
0x3010	lpbk_data_en[63:0]							
0x3018								lpbk_data_en[71:64]
0x3020							lpbk_data_mask_en	
0x3028								
0x3030	Lpbk_dat_w0[63:0]							
0x3038	Lpbk_dat_w0[127:64]							
0x3040	Lpbk_dat_w1[63:0]							
0x3048	Lpbk_dat_w1[127:64]							
0x3050		lpbk_ecc_mask_w0	lpbk_dat_mask_w0				lpbk_ecc_w0	
0x3058		lpbk_ecc_mask_w1	lpbk_dat_mask_w1				lpbk_ecc_w1	
0x3060								prbs_23
0x3068						prbs_init		
.....								
0x3100					fix_data_pattern_index	bus_width	page_size	test_engine_en
0x3108			cs_diff_tst	c_diff_tst	bg_diff_tst	ba_diff_tst	row_diff_tst	col_diff_tst
0x3120	addr_base_tst							
0x3128								
0x3130	user_data_pattern							
0x3138								
0x3140	valid_bits[63:0]							
0x3148								valid_bits[71:64]
0x3150	ctrl[63:0]							

0x3158	ctrl[127:64]						
0x3160	obs[63:0] (RD)						
0x3168	obs[127:64] (RD)						
0x3170	obs[191:128] (RD)						
0x3178	obs[255:192] (RD)						
0x3180	obs[319:256] (RD)						
0x3188	obs[383:320] (RD)						
0x3190	obs[447:384] (RD)						
0x3198	obs[511:448] (RD)						
0x31a0	obs[575:512] (RD)						
0x31a8	obs[639:576] (RD)						
0x31b0					obs[671:640](RD)		
.....							
0x3200							
0x3208							
0x3220	tud_i0						
0x3228	tud_i1						
0x3230	tud_o(RD)						
.....							
0x3300	tst_300						
0x3308	tst_308						
0x3310	tst_310						
0x3318	tst_318						
0x3320	tst_320						
0x3328	tst_328						
0x3330	tst_330						
0x3338	tst_338						
0x3340	tst_340						
0x3348	tst_348						
0x3350	tst_350						
0x3358	tst_358						
0x3360	tst_360						
0x3368	tst_368						
0x3370	tst_370						
0x3378	tst_378						

5 GMAC 控制器

5.1 寄存器描述

龙芯 2K0300 集成了两个 GMAC 控制器，即 GMAC0 和 GMAC1，二者在逻辑结构上完全相同。以下文档部分在未区分二者的情况下，表示该部分说明对二者均适用。

GMAC 控制器寄存器包括 GMAC 寄存器部分和 DMA 寄存器部分。

GMAC0 的 GMAC 寄存器的起始地址是 0x1602_0000；GMAC0 的 DMA 寄存器的起始地址是 0x1602_1000。

GMAC1 的 GMAC 寄存器的起始地址是 0x1603_0000；GMAC1 的 DMA 寄存器的起始地址是 0x1603_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 2K0300 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，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
0x1608,0000 - 0x1608,7fff	EHCI 寄存器	32KB
0x1608,8000 - 0x1608,ffff	OHCI 寄存器	32KB

7 OTG 控制器

7.1 概述

龙芯 2K0300 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 8 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
0x1604,0000 - 0x1607,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 显示控制器

8.1 概述

显示控制器从内存中取帧缓冲和光标信息输出到外部显示接口上。

龙芯 2K0300 的显示控制器支持的特性包括：

- 1) 一路 DVO 接口显示
- 2) 每路显示最大支持至 1920x1080@60Hz
- 3) Monochrome、ARGB8888 两种模式硬件光标
- 4) RGB444、RGB555、RGB565、RGB888 四种色深
- 5) 输出抖动和伽马校正
- 6) 可切换的双路线性帧缓冲
- 7) 中断和软复位

8.2 寄存器访问地址和引脚说明

显示控制器寄存器基址为：0x1609_0000。

对于显示控制器模块，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。

与显示控制器接口相关的引脚复用设置可查询 1.3 节中的外设功能引脚复用关系表，并根据 GPIO 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。

9 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。

9.1 SPI 控制器结构

芯片 2K0300 集成的 SPI0/1-flash 控制器仅可作为主控端，所连接的是从设备。芯片共集成 4 个 SPI 控制器，其中 SPI0/1 支持 IO/memory 两种空间访问，SPI2~3 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。2K0300 芯片 4 个 SPI 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
0x1c00,0000 - 0x1c0f,ffff	SPI0 Boot	1MB
0x1000,0000 - 0x11ff,ffff	SPI0 MEM	32MB
0x1200,0000 - 0x13ff,ffff	SPI1 MEM	32MB
0x1601,0000 - 0x1601,7fff	SPI0 IO/CFG	32KB
0x1601,8000 - 0x1601,ffff	SPI1 IO/CFG	32KB

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 1.3 节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。

其结构如图图9- 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)。

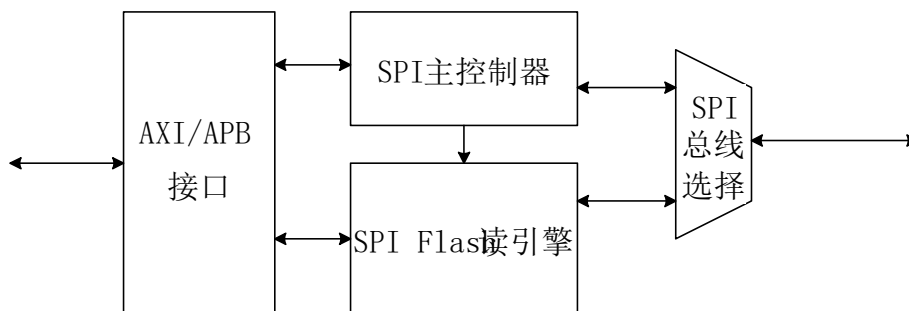


图9- 1 SPI 控制器结构

9.2 配置寄存器

表 9- 2 SPI0 配置寄存器列表

偏移	名称	描述	备注
0	SPCR	控制寄存器	SPI0 控制器支持
1	SPSR	状态寄存器	SPI0 控制器支持
2	TxFIFO/RxFIFO	数据寄存器	SPI0 控制器支持
3	SPER	外部寄存器	SPI0 控制器支持
4	SFC_PARAM	参数控制寄存器	SPI0 控制器支持
5	SFC_SOFTCS	片选控制寄存器	SPI0 控制器支持
6	SFC_TIMING	时序控制寄存器	SPI0 控制器支持

9.2.1 控制寄存器 (SPCR)

表 9- 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

9.2.2 状态寄存器 (SPSR)

表 9- 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

9.2.3 数据寄存器 (TxFIFO/RxFIFO)

表 9- 5 SPI 数据寄存器 (TxFIFO/RXFIFO)

位域	名称	访问	初值	描述
----	----	----	----	----

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

9.2.4 外部寄存器 (SPER)

表 9- 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
1:0	spre	R/W	0	与 spr 一起设定分频的比率

表 9- 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

9.2.5 参数控制寄存器 (SFC_PARAM)

表 9- 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 分频系数与 {spre, spr} 组合相同
3	dual_io	R/W	0	双 I/O 模式, 优先级高于快速读
2	fast_read	R/W	0	快速读模式
1	burst_en	R/W	0	SPI flash 支持连续地址读模式
0	memory_en	R/W	1	SPI flash 读使能, 无效时 csn[0] 可由软件控制。

9.2.6 片选控制寄存器 (SFC_SOFTCS)

表 9- 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

9.2.7 时序控制寄存器 (SFC_TIMING)

表 9- 10 SPI 时序控制寄存器(SFC_TIMING)

位域	名称	访问	初值	描述
7:3	—	—	—	保留
2	tFAST	R/W	0	SPI flash 读采样模式 0: 上沿采样, 间隔半个 SPI 周期 1: 上沿采样, 间隔一个 SPI 周期
1:0	tCSH	R/W	3	SPI Flash 的片选信号最短无效时间, 以分频后时钟周期 T 计算 00: 1T 01: 2T 10: 4T 11: 8T

9.3 接口时序

9.3.1 SPI 主控制器接口时序

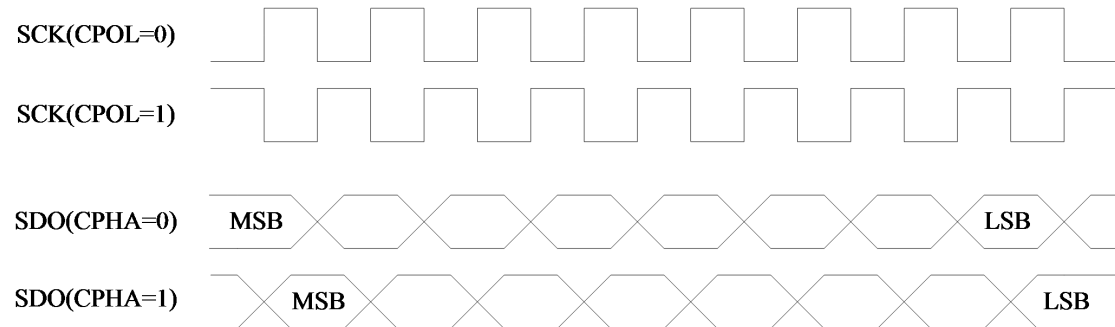


图9- 2 SPI 主控制器接口时序

9.3.2 SPI Flash 访问时序

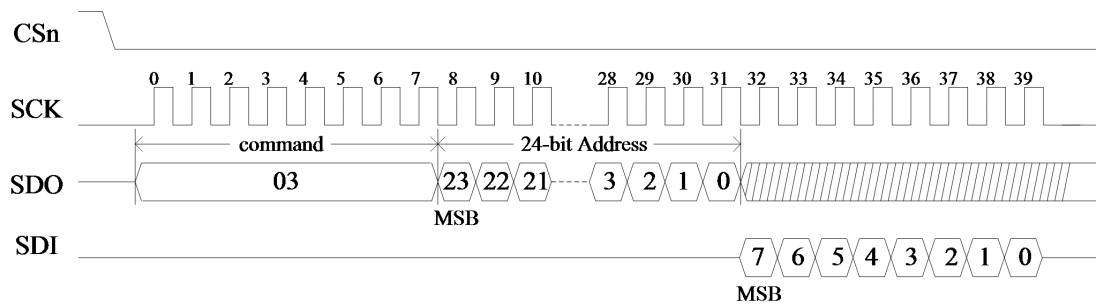


图9- 3 SPI Flash 标准读时序

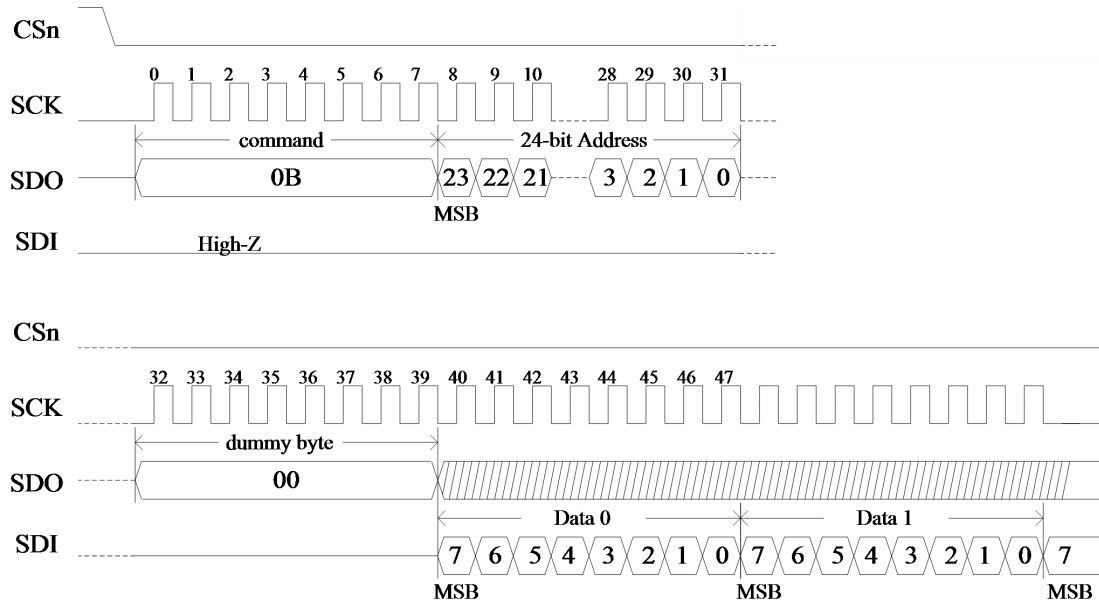


图9- 4 SPI Flash 快速读时序

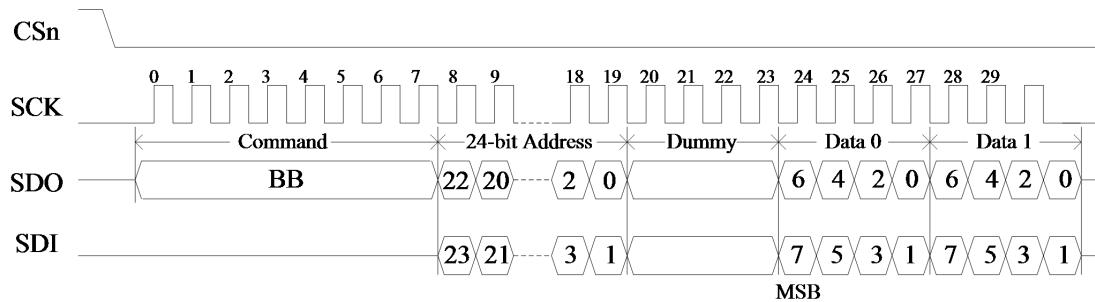


图9- 5 SPI Flash 双向 I/O 读时序

9.4 使用指南

9.4.1 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。

9.4.2 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。

9.4.3 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。

10 SPI-I/O 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。

10.1 SPI 控制器结构

龙芯 2K0300 共集成 2 个 SPI-I/O 控制器，支持 DMA 传输模式，其中 SPI2~3 仅支持 I/O 空间访问，2 个 SPI-I/O 控制器空间分布，如下表。

表 10- 1 SPI-I/O 控制器地址空间分布

地址空间	名称	大小
0x1610,c000 - 0x1610,dfff	SPI2 I/O	8KB
0x1610,e000 - 0x1610,ffff	SPI3 I/O	8KB

10.2 配置寄存器

表 10- 2 SPI-I/O 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

10.2.1 控制寄存器 1 (CR1)

表 10- 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。

10.2.2 控制寄存器 2 (CR2)

表 10- 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)

10.2.3 控制寄存器 3 (CR3)

表 10- 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)

10.2.4 控制寄存器 4 (CR4)

表 10- 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。

10.2.5 中断寄存器 (IER)

表 10- 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

10.2.6 状态寄存器 1 (SR1)

表 10- 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

10.2.7 状态寄存器 2 (SR2)

表 10- 9 SPI 状态寄存器 2 (SR2)

位域	名称	访问	初值	描述
31: 11	—	—	0	保留

位域	名称	访问	初值	描述
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

10.2.8 配置寄存器 1 (CFG1)

表 10- 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>=4bits
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

10.2.9 配置寄存器 2 (CFG2)

表 10- 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

10.2.10 配置寄存器 3 (CFG3)

表 10- 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

10.2.11 CRC 寄存器 1(CRC1)

表 10- 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

10.2.12 CRC 寄存器 2(CRC2)

表 10- 14 SPI CRC 寄存器 2(CRC2)

位域	名称	访问	初值	描述
31: 13	-	-	0	保留
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

位域	名称	访问	初值	描述
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;

10.2.13 数据寄存器 (DR)

表 10- 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。

10.3 功能描述

10.3.1 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 10- 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 10- 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 主 MOSI-从 MOSI
	从	主到从	0	0	1	0	
单工（仅从到主）	主	从到主	1	0	1	0	主 SCK-从 SCK 主 MISO-从 MISO
	从	从到主	0	0	0	1	

10.3.2 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 10- 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

10.3.3 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

10.3.4 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)。

10.3.5 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。

10.3.6 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 10- 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。

10.3.7 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。

10.3.8 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。

11 I2S 控制器

11.1 I2S 控制器概述

龙芯 2K0300 集成一个 I2S 控制器，数据宽度是 32 位，支持 DMA 传输，支持多家公司的 codec 芯片。I2S 控制器支持主或从模式。主模式时由 I2S 控制器产生位时钟信号、左右声道选择时钟信号和数据信号，从模式时 I2S 控制器接收位时钟信号、左右声道选择时钟信号和数据信号。主模式时，codec 系统时钟由控制器提供，从模式时，系统时钟可由控制器或晶振提供。I2S 的功能特性包括：

1. 支持 8、16、18、20、24、32 位的音频数据采样位宽。
2. 支持 8、16、32 位的左右声道处理字宽。
3. 包含两个缓存 FIFO，FIFO 的缓存容量为 8bytes。
4. I2S 的中断处理模式可配，在 I2S 的发送和接收中断功能都使能后，当两个通道的缓存 fifo 为满仍要写以及为空仍要读时，则向 CPU 发出中断信号。
5. I2S 可以为 codec 芯片提供系统时钟，时钟频率可配。

11.2 I2S 配置寄存器

I2S 配置寄存器地址空间，如下表。

表 11- 1 I2S 控制器地址空间分布

地址空间	名称	大小
0x1611,4000 - 0x1611,7fff	I2S 配置寄存器	16KB

表 11- 2 I2S 控制器地址空间分布

寄存器名称	偏移地址	读/写 (R/W)	功能描述	复位值
IISVersion	0x0000	R/W	I2S 标识寄存器	32'h0
IISConfig	0x0004	R/W	I2S 配置寄存器	32'h0
IISControl	0x0008	R/W	I2S 控制寄存器	32'h0
IISRxData	0x000c	R/W	I2S 接收数据寄存器（用于 DMA 接收数据）	32'h0
IISTxData	0x0010	R/W	I2S 发送数据寄存器（用于 DMA 发送数据）	32'h0
IISConfig1	0xd014	R/W	I2S 配置寄存器 1	32'h0

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

表 11- 3 I2S 标识寄存器

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

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

配置寄存器 0 是配置 I2S 的声道字长, 发送和接收音频数据的采样深度以及位时钟的分频系数。

表 11- 4 I2S 配置寄存器 0

寄存器名称	偏移地址	读/写 (R/W)	功能描述
IISConfig0	位	缺省值	描述
LR_LEN	31:24	'h0	左右声道处理的字长。
TX_RES_DEPTH	23:16	'h0	TX 采样深度设置: IIS 采样数据长度, 有效范围为 8-32, 如果发送的数据宽度小于采样数据长度, 则低位补 0; 如果发送的数据宽度大于采样数据长度, 则低位忽略。
BCLK_RATIO	15:8	'h0	位时钟 (BCLK) 分频系数: 位时钟分频系数, 分频数为 MCLK 时钟频率除以 2x (RATIO+1)
RX_RES_DEPTH	7:0	'h0	RX 采样深度设置: IIS 采样数据长度, 有效范围为 8-32, 如果接收到的数据宽度小于采样数据长度, 则低位补 0; 如果接收到的数据宽度大于采样数据长度, 则低位忽略。

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

表 11- 5 I2S 控制寄存器

寄存器名称	偏移地址	读/写 (R/W)	功能描述
IISControl	位	缺省值	描述
MCLK_READY	16	R	系统时钟 (MCLK) 输出稳定标志, 为 1 时时钟稳定输出, 为 0 时输出时钟不可用
MASTER	15	'h0	1: IIS 工作于主模式
MSB/LSB	14	'h0	1: 高位在左端 0: 高位在右端
RX_EN	13	'h0	控制器接收使能, 为 1 时有效, 开始接收数据
TX_EN	12	'h0	控制器发送使能, 为 1 时有效, 开始发送数据

寄存器名称	偏移地址	读/写(R/W)	功能描述
RX_DMA_EN	11	'h0	DMA 接收使能，为 1 时有效
Reserved	10: 9	'h0	
CLK_READY	8	R	位时钟和声道选择时钟输出稳定标志，为 1 时 时钟稳定输出，为 0 时输出时钟不可用
TX_DMA_EN	7	'h0	DMA 发送使能，为 1 时有效
Reserved	6:5	'h0	
RESETn	4	'h0	IIS 控制器软复位
MCLK_EN	3	'h0	使能时钟输出
RX_INT_EN	1	'h0	RX 中断使能，为 1 时使能中断，为 0 时禁止
TX_INT_EN	0	'h0	TX 中断使能，为 1 时使能中断，为 0 时禁止

配置寄存器 1 是配置系统时钟（MCLK）的分频系数。

表 11- 6 I2S 配置寄存器 1

寄存器名称	偏移地址	读/写(R/W)	功能描述
IISConfig1	位	缺省值	描述
MCLK_RATIO	15:0	'h0	系统时钟(MCLK)分频系数整数部分： 分频数为总线时钟频率除以系统时钟向下 取整的值，系统时钟作为 Codec 的 sysclk
MCLK_RATIO_F RAC	32:16	'h0	系统时钟(MCLK)分频系数小数部分： 分频数为总线时钟频率除以系统时钟得小 数部分乘以 2^{16}

12 I2C 控制器

12.1 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 2K0300 芯片 4 个 I2C 控制器空间分布，如下表。

表 12- 1 I2C 控制器地址空间分布

地址空间	名称	大小
0x1610,8000 - 0x1610,8fff	I2C0	4KB
0x1610,9000 - 0x1610,9fff	I2C1	4KB
0x1610,a000 - 0x1610,afff	I2C2	4KB
0x1610,b000 - 0x1610,bfff	I2C3	4KB

12.2 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。

12.3 功能描述

12.3.1 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 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。

12.3.2 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。

12.3.2.1 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_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。

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_BTf 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_AF）会置起，控制器回到空闲状态，等待下一个“开始”条件。

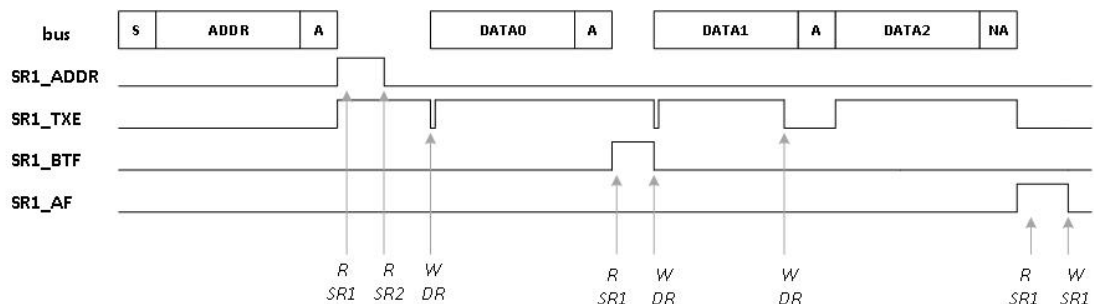


图12- 1 从发送模式示意图

12.3.2.2 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（SR_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。

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）。

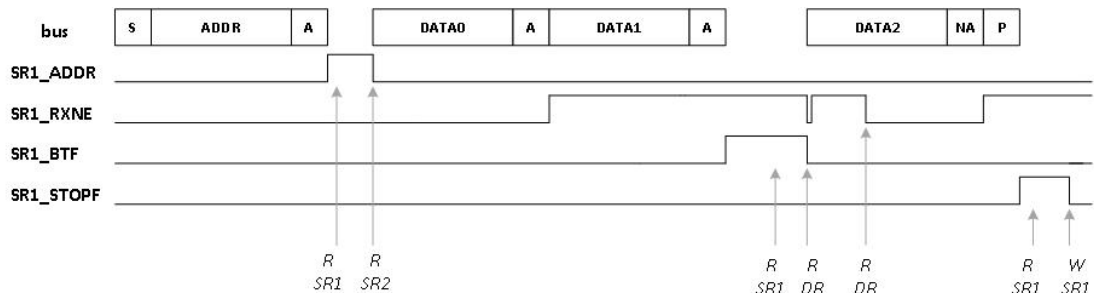


图12- 2 从接收模式示意图

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），在下次接收模式时（无论主从），会直接产生 SR1_BTF 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。

12.3.3 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。

12.3.3.1 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。

12.3.3.2 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。

12.3.3.3 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。

12.3.3.4 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。

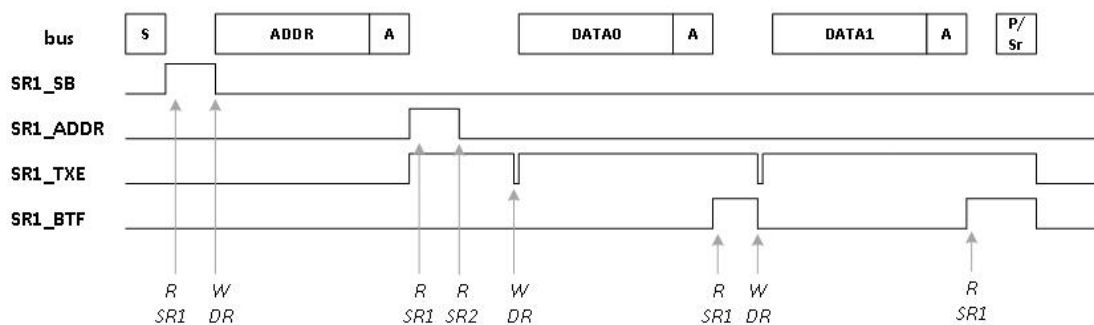


图12- 3 主发送模式示意图

12.3.3.5 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。

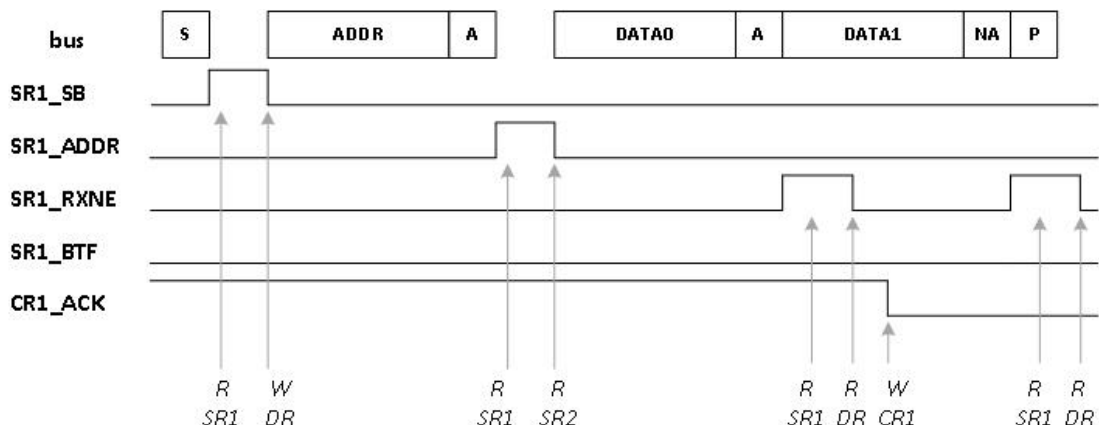


图12- 4 主接收模式（及时）示意图

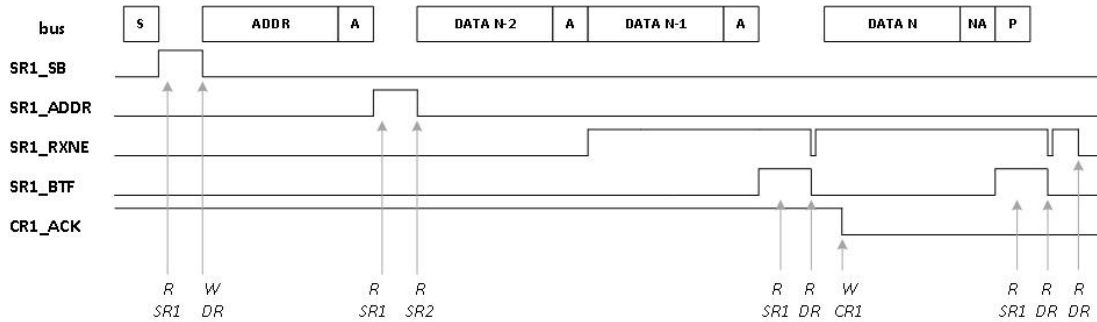


图12- 5 主接收模式（非及时、3 字节）示意图

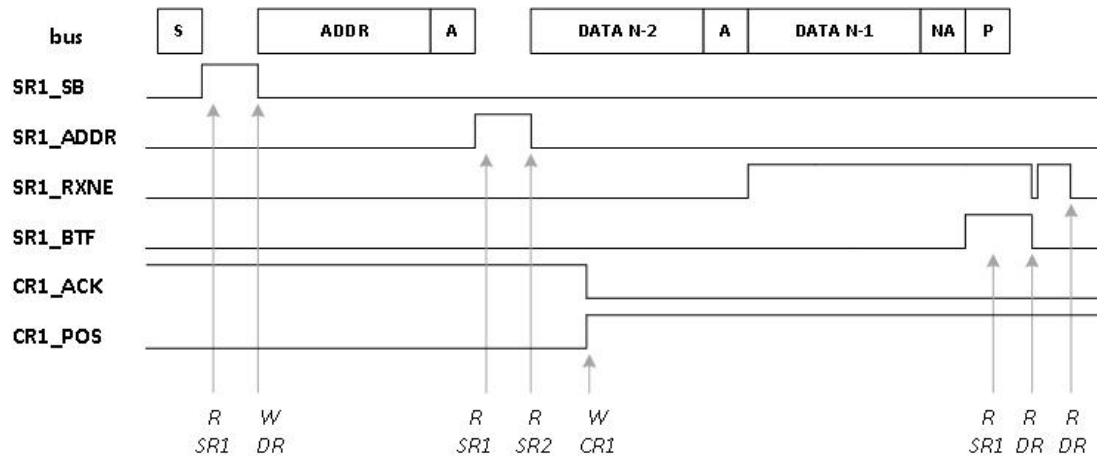


图12- 6 主接收模式（非及时、2 字节）示意图

12.3.4 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。

12.3.5 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。

12.4 中断

中断状态和对应的使能控制位如下表：

表 12- 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

12.5 寄存器定义

表 12- 3 I2C 控制器寄存器列表

地址偏移	寄存器名称	描述
0x00	I2C_CR1	I2C 控制寄存器 1
0x04	I2C_CR2	I2C 控制寄存器 2
0x08	I2C_OAR	I2C 从地址寄存器
0x10	I2C_DR	I2C 数据寄存器
0x14	I2C_SR1	I2C 状态寄存器 1
0x18	I2C_SR2	I2C 状态寄存器 2
0x1c	I2C_CCR	I2C 时钟控制寄存器
0x20	I2C_TRISE	I2C 上升时间寄存器

12.5.1 I2C 控制寄存器（I2C_CR1）

偏移量： 0x00

复位值： 0x00000000

表 12- 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。
13:12	Reserved	—	—
11	CR1_POS	RW	数据接收的应答位置 0: CR1_ACK 位控制当前移位寄存器正在接收的字节的应答; 1: CR1_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 CR1_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 CR1_ACK 表示的值。 软件可设置和清除该位,硬件会在 CR1_PE 为 0 时清除该位。
10	CR1_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 CR1_PE 为 0 时清除该位。
9	CR1_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	CR1_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 CR1_PE 为 0 时清除该位。
7	CR1_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	CR1_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	CR1_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。
---	--------	----	--

12.5.2 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 12- 5 I2C 控制寄存器 2

位域	名称	访问	描述
31:12	Reserved	—	—
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
7:6	Reserved	—	—
5:0	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）。

12.5.3 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 12- 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	—	—

12.5.4 I2C 数据寄存器 (I2C_DR)

偏移量： 0x10

复位值： 0x00000000

表 12- 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。

12.5.5 I2C 状态寄存器（I2C_SR1）

偏移量： 0x14

复位值： 0x00000000

表 12- 8 I2C 状态寄存器

位域	名称	访问	描述
31:12	Reserved	—	—
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。 要清除改位，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。

12.5.6 I2C 状态寄存器 2（I2C_SR2）

偏移量： 0x18

复位值： 0x00000000

表 12- 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。
---	---------	---	---

12.5.7 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 12- 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

12.5.8 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 12- 11 I2C 上升时间寄存器

位域	名称	访问	描述
31:6	Reserved	—	—

5:0	TRISE	RW	<p>SCL 最大上升时间（主模式）</p> <p>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。</p> <p>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。</p>
-----	-------	----	--

13 LocalIO 控制器

13.1 访问地址及引脚复用

LocalIO 总线访问地址空间如下表：

表 13- 1 LocalIO 地址空间分布

起始地址	结束名称	名称	说明
0x1400_0000	0x15ff_ffff	LIO-MEM	32MB 大小

LocalIO 总线地址共 23 位，支持 8 位/16 位数据位宽，采用双片选控制，可支持最大存储容量 16MB(8 位，CS0/低 8MB，CS1/高 8MB)和 32MB(16 位，CS0/低 16MB，CS1/高 16MB)。

LocalIO 总线访问参数寄存器配置可查询节 3.4.2 中芯片配置项，主要包括配置 LIO 总线访问延迟参数，即：LIO 总线读写控制信号 RDn/WRn 有效电平时长，该延迟参数范围为 1~32 个总线周期（即：LocalIO 控制器时钟周期）。具体配置方法：(1)总线访问延迟计数步长周期数 `clk_period[1:0]` (即：每累加一次访问延迟计数对应的 LIO 控制器时钟周期数，0：步长为 1；1：步长为 4；2：步长为 2；3：步长为 1)；(2)访问延迟初始值 `rom_count[4:0]` (即：访问延迟计数以该初始值开始计数，延迟最高计数至 32，该配置值内部按位取反，即，0：初值为 31；1：初值为 30；依次类推...，31：初值为 0)；(3)LIO 总线 8/16 位数据宽度配置项等。

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

与 LocalIO 接口相关的引脚复用设置可查询 1.3 节功能引脚复用关系表，并根据节 GPIO 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。

13.2 LocalIO 控制器功能概述

LocalIO 控制器提供了简单外设访问接口，主要用于连接存储 ROM 或扩展外设。它对外提供两个片选，具有可配置的数据位宽和访问延迟。其中 `wait` 参数指 `liord` 或 `liowr` 信号为低的周期数减一，读写时序可参考图 13-1，图 13-2。当数据位宽为 16 时，送出的地址由 CPU 物理地址右移一位得到。

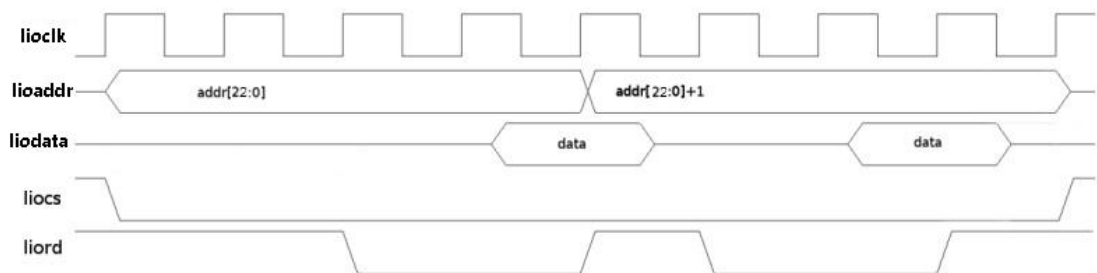


图13- 1 LocalIO 读时序

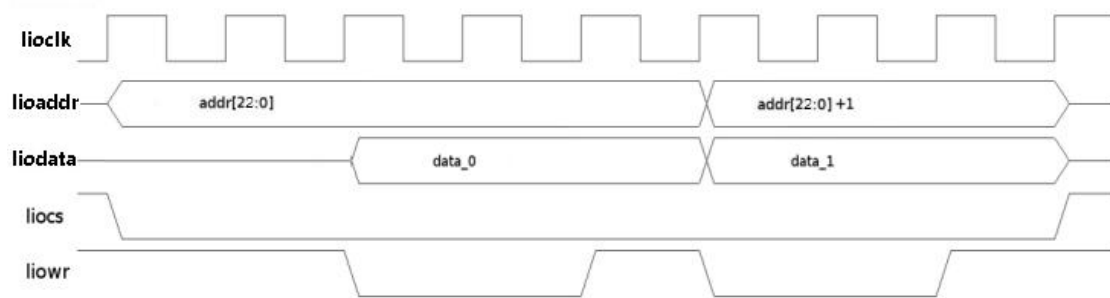


图13- 2 LocalIO 写时序

读写时序说明:

图中 LIOCLK 信号外部引脚并不存在, 属于 LocalIO 内部控制器时钟 (即: 时钟结构网络中 BOOT 模块时钟源), 只是方便时序描述。

liord 和 liowr 低有效时间与 LIO clk_period、rom_count_init 设置值有关:

1). LIO clk_period 设置为 1 时-有效电平持续最长 8 个内部时钟周期 (例: 当 rom 计数初值配置为 31, 即从 0 开始计时, 按照步长为 4 累加, 最高计数至 32 停止, 共累计 8 个时钟周期; 当增加 rom 计数初值时将缩短有效电平持续时长)。

2). LIO clk_period 设置为 2 时-有效电平持续最长 16 个内部时钟周期 (例: 当 rom 计数初值配置为 31, 即从 0 开始计时, 按照步长为 2 累加, 最高计数至 32 停止, 共累计 16 个时钟周期; 当增加 rom 计数初值时将缩短有效电平持续时长)。

3). LIO clk_period 设置为 3/0 时-有效电平持续最长 32 个内部时钟周期 (例: 当 rom 计数初值配置为 31, 即从 0 开始计时, 按照步长为 1 累加, 最高计数至 32 停止, 共累计 32 个时钟周期; 当增加 rom 计数初值时将缩短有效电平持续时长)。

4). 单次 CS 有效期间可能出现多次读写操作, 以上时序图仅作为一种示例。

14 UART 控制器

14.1 概述

龙芯 2K0300 集成了 10 个 UART 控制器，通过 APB 总线与总线桥通信，支持 DMA 传输。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。

14.2 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。

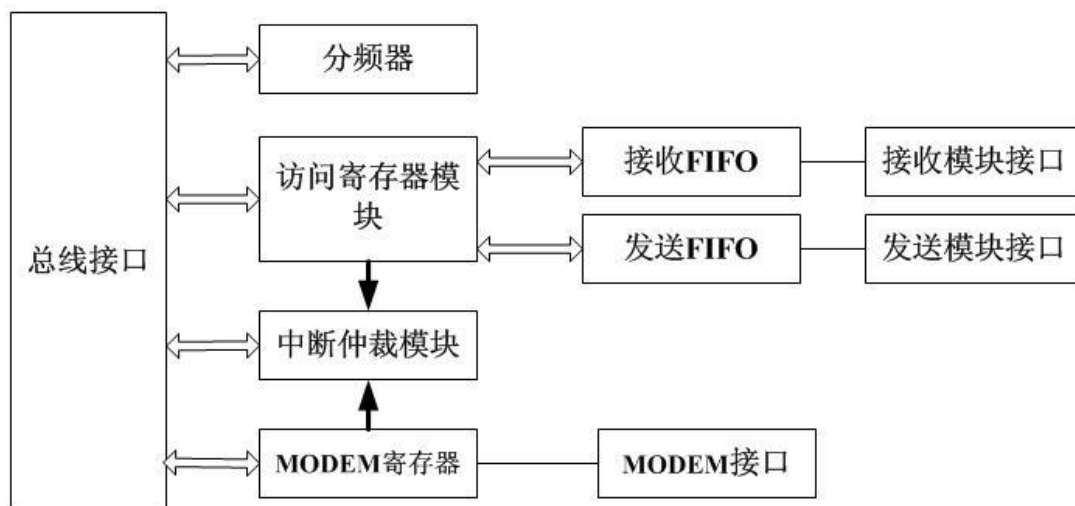


图14- 1 UART 控制器结构

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。

14.3 寄存器描述

龙芯2K0300芯片中集成10个UART控制器，每个UART仅访问基址不一样，其功能寄存器均保持一致。

表 14- 1 UART0~9 控制器地址空间分布

地址空间	名称	大小
0x1610,0000 - 0x1610,03ff	UART0 配置寄存器	1KB
0x1610,0400 - 0x1610,07ff	UART1 配置寄存器	1KB
0x1610,0800 - 0x1610,0bff	UART2 配置寄存器	1KB
0x1610,0c00 - 0x1610,0fff	UART3 配置寄存器	1KB
0x1610,1000 - 0x1610,13ff	UART4 配置寄存器	1KB
0x1610,1400 - 0x1610,17ff	UART5 配置寄存器	1KB
0x1610,1800 - 0x1610,1bff	UART6 配置寄存器	1KB
0x1610,1c00 - 0x1610,1fff	UART7 配置寄存器	1KB
0x1610,2000 - 0x1610,23ff	UART8 配置寄存器	1KB
0x1610,2400 - 0x1610,27ff	UART9 配置寄存器	1KB

UART 接口模块配置寄存器，如下表所示。

表 14- 2 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	Modem 控制寄存器
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 1.3 章节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。

14.3.1 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 14- 3 数据传输寄存器

位域	位域名称	位宽	访问	描述
7:0	DATA	8	RW	写访问为发送数据、读访问为接收数据

14.3.2 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 14- 4 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 未滿，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 未滿，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

14.3.3 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 14- 5 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

表 14- 6 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 个字符时间内没有任何操作，包括读和写操作	读接收 FIFO
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

14.3.4 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 14- 7 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’ - 2 字节 ‘10’ - 3 字节 ‘11’ - 4 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

14.3.5 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 14- 8 线路控制寄存器

位域	位域名称	位宽	访问	描述
----	------	----	----	----

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

14.3.6 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 14- 9 Modem 控制寄存器

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

14.3.7 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 14- 10 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
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。

14.3.8 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 14- 11 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

14.3.9 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 14- 12 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 14- 13 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 14- 14 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。

15 SDIO 控制器

15.1 功能概述

龙芯 2K0300 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

15.2 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 15- 1 SDIO 内部寄存器物理地址构成

地址	设备	备注
0x1614_0000	SDIO0	32KB 大小寄存器配置空间
0x1614_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 1.3 节 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。

15.3 寄存器描述

SDIO 控制器的寄存器详细说明如下：

表 15- 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 15- 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 15- 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 15- 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 15- 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 15- 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 15- 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 15- 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
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 15- 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 15- 11 SDI_CMD_STA 寄存器位域描述

SDI_CMD_STA	位	缺省值	描述
Reserved	31:13	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 15- 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 15- 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 15- 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 15- 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 15- 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 15- 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 15- 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 15- 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 15- 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 15- 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 15- 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 15- 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 15- 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 15- 25 SDI_DAT_CON 寄存器位域描述

SDI_DAT_CON	位	缺省值	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 15- 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 15- 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 15- 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 15- 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

SDI_DAT_STA	位	缺省值	描述
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 15- 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 15- 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	—
rx_full	8	0x0	Rx FIFO 满标志
rx_empty	7	0x0	Rx FIFO 空标志位
Reserved	6:0	0x0	—

表 15- 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 15- 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 15- 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	R0	SDIO 命令数据寄存器	0x0

表 15- 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 15- 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 15- 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
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_crc_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 15- 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 15- 39 dll_master_val 寄存器位域描述

dll_master_val	位	缺省值	描述
reserved	31:4	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 15- 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 15- 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 15- 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 15- 43 param_delay 寄存器位域描述

param_delay	位	缺省值	描述
reserved	31:16	0x0	—

param_delay	位	缺省值	描述
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 15- 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 15- 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 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

15.4 软件编程指南

15.4.1 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。

15.4.2 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。

15.4.3 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。

15.5 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

16 eMMC 控制器

16.1 功能概述

龙芯 2K0300 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

16.2 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 16- 1 eMMC 内部寄存器物理地址构成

地址	设备	备注
0x1614_0000	SDIO0/eMMC0	32KB 大小寄存器配置空间
0x1614_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 1.3 节中 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。

16.3 寄存器描述

eMMC 控制器的寄存器详细说明如下：

表 16- 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 16- 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 16- 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 16- 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 16- 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 16- 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 16- 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 16- 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
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 16- 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 16- 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 16- 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 16- 13 EMMC_RSP0 寄存器位域描述

EMMC_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 16- 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 16- 15 EMMC_RESP1 寄存器位域描述

EMMC_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 16- 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 16- 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 16- 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 16- 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 16- 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 16- 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 16- 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 16- 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 16- 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 16- 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 16- 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 16- 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 16- 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 16- 29 EMMC_DAT_STA 寄存器位域描述

EMMC_DAT_STA	位	缺省值	描述
Reserved	31:17	0x0	–

EMMC_DAT_STA	位	缺省值	描述
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 16- 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	RO	eMMC FIFO 状态寄存器	0x0

表 16- 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 16- 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 16- 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 16- 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 16- 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 16- 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 中断寄使能寄存器	0x0

表 16- 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
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 16- 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 16- 39 DLL_MASTER_VAL 寄存器位域描述

dll_master_val	位	缺省值	描述
reserved	31:4	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 16- 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 16- 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 16- 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 16- 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 16- 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 16- 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

16.4 专用 DMA 控制器

16.4.1 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。

16.4.2 DMA 描述符

DMA_ORDER_ADDR_LOW

偏移地址： 0x0

复位值：0x00000000

表 16- 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 16- 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 eMMC 设备，该寄存器指定了读内存的地址；内存写：从 eMMC 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 16- 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 eMMC 设备地址

DMA_LENGTH

偏移地址：0xc

复位值：0x00000000

表 16- 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 16- 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 16- 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 16- 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 16- 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 16- 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 16- 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 16- 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)

16.5 软件配置流程

16.5.1 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。

16.5.2 eMMC 初始化流程

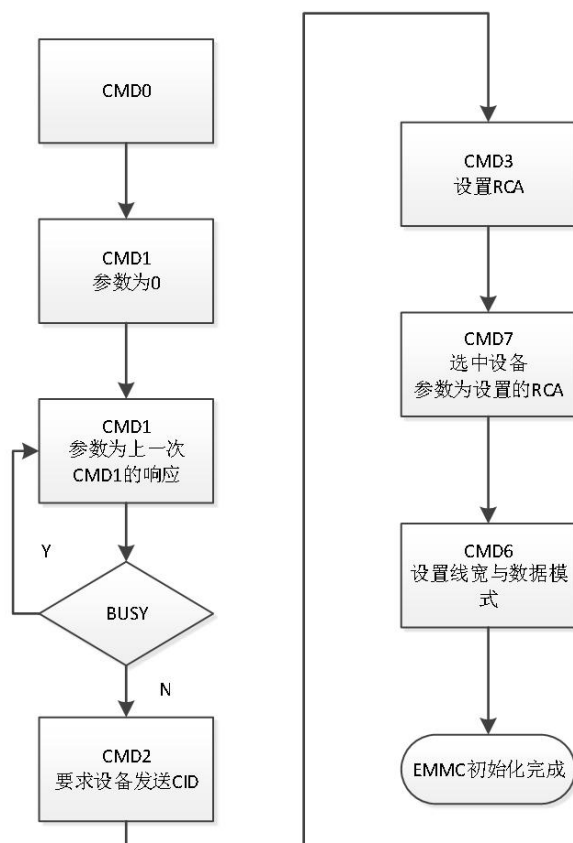


图16- 1 eMMC 初始化流程图

16.5.3 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。

17 CANFD 控制器

17.1 概述

龙芯 2K0300 集成了四路 CANFD 接口控制器。CAN 总线是由发送数据线 TX 和接收数据线 RX 构成的串行总线，可发送和接收数据。器件与器件之间进行双向传送，最高传送速率 10Mbps。

17.2 CANFD 控制器特性

该 CANFD 控制器主要特性有：

1. 支持 CANFD 协议，兼容 CAN2.0 协议；
2. RX 缓存区大小为 32（宽）x20（深）；
3. 8 个可容纳满载荷 CANFD 报文（最长 64 字节）的 TX 缓存区；
4. 支持 ISO 与 non-ISO CANFD 协议；
5. 支持接收添加时间戳与定时发送功能；
6. 支持 Loopback mode、Bus monitoring mode、ACK forbidden mode、Self-test mode 与 Restricted operation mode。

四路 CANFD 控制器的寄存器地址分配如下表：

表 17- 1 CANFD 控制器地址空间分布

地址空间	名称	大小
0x1611,0000 - 0x1610,03ff	CAN0	1KB
0x1611,0400 - 0x1610,07ff	CAN1	1KB
0x1611,0800 - 0x1610,0bff	CAN2	1KB
0x1611,0c00 - 0x1610,0fff	CAN3	1KB

18 ATIM 控制器

18.1 概述

龙芯 2K0300 集成了一个由 32 位自动装载计数器驱动的 ATIM。

它适用于多种应用，如测量输入信号的脉冲长度或产生指定的输出波形。同时 ATIM 具有三路互补输出通道，对于电机相关应用具有良好的支持。

ATIM 的时钟由内部 APB 总线时钟提供，基地址为 0x16118000，寄存器定义见下文。

18.2 功能描述

18.2.1 计数模式

定时器具有向上、向下、中央对齐 3 种计数模式，其中单向计数模式与中央对齐模式通过 CR1 寄存器中的 CMS 配置位进行选择。在单向计数模式中，用户通过配置 CR1 寄存器中的 DIR 位选择向上/向下计数。在中央对齐模式中，DIR 位只读，由硬件自动设置用于指示当前的计数方向。

用户可通过配置 PSC 寄存器对计数器时钟进行预分频，配置 ARR 寄存器用于配置计数周期，且可对 CNT 计数器实时进行读取与修改，实际完成一次计数周期的时长由 PSC 与 ARR 共同决定。

PSC 与 ARR 寄存器具有预装载功能，即当次的配置值会留到下个计数周期/更新事件产生时才生效，其中 PSC 寄存器预装载功能始终打开，ARR 寄存器预装载功能可自由选择开启或关闭。

ATIM 模块中 CNT 的计数时钟默认由内部 APB 总线时钟提供，用户也可以通过配置 SMCR 寄存器以使用外部输入的脉冲信号或芯片内部互联信号来进行计数，以实现更灵活的应用。

每次计数器溢出、设置 UG 位、从模式控制器产生硬件复位时可根据相关配置位配置产生更新事件。

ATIM 中具有重复计数模式。通过配置重复计数器，可以让用户控制在每 N 次计数上溢或下溢时产生一次更新事件，其中 N 为 RCR 重复计数寄存器中的值。重复计数器在任意模式下计数器发生上溢/下溢时计数。重复计数器始终开启预装载。

当发生更新事件时，PSC 内部计数被清零，CNT 根据计数模式进行重载，同时：

- SR 寄存器中的 UIF 标志位被设置。
- PSC 与 RCR 寄存器中的值被装载到实际的内部寄存器。
- 若设置了 ARPE 位，ARR 寄存器中的值被装载到实际的自动重载寄存器。

通过设置 CR1 寄存器中的 UDIS 位，可以禁止除写入 UG 位外的更新事件产生。如设置了 CR1 寄存器中的 URS 位，通过设置 EGR 寄存器中的 UG 位可以手动产生一个更新事件，但

不设置 UIF 标志位。

18.2.2 输入模式

ATIM 具有 4 个独立的输入/输出通道，每个通道都可独立地选择输入/输出模式，但同一个通道同时只能选择输入或输出模式中的一种。

在输入模式下，当检测到 CHx 信号上相应的边沿后，计数器的当前值被锁存到捕获/比较寄存器（CCR_x）中。当捕获事件发生时，SR 寄存器中相应通道的 CC_xIF 标志被置 1，如果在 DIER 中使能了相应的中断或 DMA，则将产生中断或 DMA 请求。如果捕获事件发生时 CC_xIF 标志已经为 1，那么 SR 寄存器中的重复捕获标志 CC_xOF 标志将被置 1，该标志有助于用户判断是否存在被错过的捕获事件。

写 CC_xIF=0 可清除 CC_xIF，或读取存储在对应通道的 CCR_x 寄存器中的数据也可清除 CC_xIF。写 CC_xOF=0 可清除 CC_xOF。

注：设置 EGR 寄存器中相应的 CC_xG 位，可以通过软件输入产生输入捕获事件，同时根据 DIER 寄存器的配置产生中断或 DMA 请求。

18.2.2.1 编码器接口模式

在编码器接口模式是输入模式下的一种特殊应用，可连接外部增量式编码器用于计数。

两个输入 CH1 和 CH2 被用来作为增量编码器的接口。根据两个输入信号的跳变顺序，产生了计数脉冲和方向信号，计数器向上或向下计数，同时硬件对 CR1 寄存器的 DIR 位进行相应的设置。

选择编码器接口模式的方法是：如果计数器只在 CH2 的边沿计数，则置 SMCR 寄存器中的 SMS=001；如果只在 CH1 边沿计数，则置 SMS=010；如果计数器同时在 CH1 和 CH2 边沿计数，则置 SMS=011。

编码器接口模式在开始计数之前必须配置 ARR 与其他寄存器，其配置与通常的计数模式保持一致。

在这个模式下，计数器依照增量编码器的速度和方向被自动的修改，因此计数器的内容始终指示着编码器的位置。计数方向与相连的传感器旋转的方向对应。

表 18- 1 工作模式配置

有效边沿	相对信号电平	CH1FP1		CH2FP2	
		上升沿	下降沿	上升沿	下降沿
仅在 CH1 计数	高	向下计数	向上计数	不计数	不计数
	低	向上计数	向下计数	不计数	不计数
仅在 CH2 计数	高	不计数	不计数	向上计数	向下计数
	低	不计数	不计数	向下计数	向上计数
在 CH1 和 CH2 均计数	高	向下计数	向上计数	向上计数	向下计数
	低	向上计数	向下计数	向下计数	向上计数

18.2.2.2 霍尔传感器模式

定时器为外部霍尔传感器应用进行优化。

通过配置 CR2 寄存器中的 TI1S 位, 可将 CH1、CH2、CH3 输入异或后作为通道 1 输入, 因此外部霍尔传感器的所有边沿均可通过通道 1 触发内部中断, 进而读取外部霍尔传感器输入电平确定转子位置, 完成换相配置。

18.2.3 输出模式

该功能用于控制一个输出波形, 或者通过电平变化指示时间的变化。其中通道 1-3 具有互补输出通道。在输出模式下捕获/比较寄存器可开启预装载功能。

18.2.3.1 电平输出模式

当计数器 CNT 与 CCRx 寄存器的值相同时, 即称为发生比较匹配事件, 此时 OCxREF 信号根据所配置的输出模式, 可以保持原本电平 (OCxM=000)、被设置为有效电平 (OCxM=001)、被设置为无效电平 (OCxM=010)、始终输出低电平 (OCxM=0x100)、始终输出高电平 (OCxM=0x101) 或将当前电平状态进行翻转 (OCxM=011), 再根据 CCER 寄存器中 CCxP 输出极性配置, 将符合期望的值输出到对应的引脚上。同时 SR 寄存器中对应的 CCxIF 标志位被设置, 并根据 DIER/CR2 寄存器相关配置产生中断或 DMA 请求。

例如: 配置 OC1M=101, CC1P=0, 即可强制 OC1 输出高电平。

18.2.3.2 PWM 模式

脉冲宽度调制模式可以产生一个由 ARR 寄存器确定频率、由 CCRx 寄存器确定占空比的信号。

在 CCMRx 寄存器中的 OCxM 位写入 110 (PWM 模式 1) 或 111 (PWM 模式 2), 能够独立地设置每个 OCx 输出通道产生一路 PWM。用户可设置 CCMRx 寄存器 OCxPE 位以使能相应的预装载寄存器。

仅当发生一个更新事件的时候, 被配置为预装载模式的寄存器中的配置值才可生效, 因此在计数器开始计数之前, 必须通过设置 EGR 寄存器中的 UG 位来初始化所有的寄存器。

CCER 寄存器中的 CCxP/CCxE 位控制 OCx 输出极性与使能。

根据 CR1 寄存器中 CMS 位的状态, 定时器能够产生边沿对齐的 PWM 信号或中央对齐的 PWM 信号。

当 CR1 寄存器中的 DIR 位为低的时候执行向上计数。以 PWM 模式 1 为例, 当 $CNT < CCRx$ 时 PWM 信号参考 OCxREF 为高, 否则为低。如果 CCRx 中的比较值大于自动重装载值 (ARR), 则 OCxREF 保持为 1。如果 CCRx 中的比较值为 0, 则 OCxREF 保持为 0。

当 CR1 寄存器的 DIR 位为高时执行向下计数。在 PWM 模式 1 时, 当 $CNT > CCRx$ 时参考信号 OCxREF 为低, 否则为高。如果 CCRx 中的比较值大于 ARR 中的自动重装载值, 则 OCxREF 保持为 1。该模式下不能产生占空比为 0% 的 PWM 波形。

对于一个给定的通道, 设置 CCMRx 寄存器中对应的 OCxCE 位为 1, 能够用 ETRF 输入端的

高电平把 OCxREF 信号拉低, OCxREF 信号将保持为低直到发生下一次的更新事件 UEV。

该功能只能用于输出比较和 PWM 模式, 而不能用于强置模式。

将 CR1 寄存器中的 OPM 位置为 1 将选择单脉冲模式, 这样可以让定时器自动地在产生下一个更新事件 UEV 时清除 CEN 位, 从而停止计数。这种模式可用于让定时器响应一个激励, 并在一个程序可控的延时之后, 产生一个脉宽可程序控制的脉冲。

18.2.3.3 互补输出和死区插入

ATIM 的 CH1-CH3 通道可以输出两路互补信号, 并且能够在互补信号中插入用户可配置的死区时间。每个互补通道都有一个 10 位的死区发生器, 参考信号 OCxREF 可以产生两路输出 OCx 和 OCxN, 且可为每一个输出独立选择极性。如果 OCx 和 OCxN 均为高有效:

- OCx 输出信号与参考信号相同, 只是其上升沿相对于参考信号的上升沿有一个延迟
- OCxN 输出信号与参考信号相反, 只是其上升沿相对于参考信号的下降沿有一个延迟

若延迟大于当前有效的输出宽度, 则不会产生相应的脉冲。

各通道的死区延迟都是相同的, 通过 BDTR 寄存器中的 DTG 位配置。

在输出模式下, 通过配置 CCER 寄存器的 CCxE 位和 CCxNE 位, OCxREF 可以被重定向到 OCx 或 OCxN 的输出。当只使能 OCx 或只使能 OCxN 时, OCxREF 直接被连接到 OCx 或 OCxN。当 OCx 与 OCxN 都被使能时, OCxREF 先经过互补输出与死区插入控制模块, 产生带有死区的互补信号, 再分别连接到 OCx 与 OCxN。

18.2.3.4 刹车输入

通过设置 BDTR 寄存器中的 BKE 位和 BKP 位可以使能刹车功能与选择刹车输入信号的极性。当在刹车输入端出现选中的电平时, 产生一个刹车事件, MOE 位被清除, 将输出关闭。同时若配置了 DIER 寄存器中的 BIE 位, 则产生一个中断。

如果设置了 BDTR 寄存器中的 AOE 位, 则在下一个更新事件产生时, MOE 被自动置位。

18.2.4 定时器外部控制

通过配置 SMCR 寄存器中的 SMS 位, 可使用外部信号对定时器进行一定程度的控制。

18.3 寄存器描述

18.3.1 寄存器地址列表

表 18- 2 ATIM 寄存器列表

名称	偏移地址	位宽	描述
ATIM_CR1	0x00	32	控制寄存器 1
ATIM_CR2	0x04	32	控制寄存器 2

ATIM_SMCR	0x08	32	从模式控制寄存器
ATIM_DIER	0x0C	32	DMA/中断使能寄存器
ATIM_SR	0x10	32	状态寄存器
ATIM_EGR	0x14	32	事件产生寄存器
ATIM_CCMR1	0x18	32	捕获/比较模式寄存器 1
ATIM_CCMR2	0x1C	32	捕获/比较模式寄存器 2
ATIM_CCER	0x20	32	捕获/比较使能寄存器
ATIM_CNT	0x24	32	计数器
ATIM_PSC	0x28	32	预分频器
ATIM_ARR	0x2C	32	自动重装载寄存器
ATIM_RCR	0x30	32	重复计数寄存器
ATIM_CCR1	0x34	32	捕获/比较寄存器 1
ATIM_CCR2	0x38	32	捕获/比较寄存器 2
ATIM_CCR3	0x3C	32	捕获/比较寄存器 3
ATIM_CCR4	0x40	32	捕获/比较寄存器 4
ATIM_BDTR	0x44	32	刹车/死区寄存器
ATIM_INSTA	0x50	32	输入通道状态寄存器

18.3.2 ATIM_CR1

中文名： ATIM 控制寄存器 1
寄存器位宽： [31:0]
偏移量： 0x00
复位值： 0x00000000

表 18- 3 控制寄存器 1

位域	位域名称	访问	描述
31:10	Reserved	—	保留
9: 8	CKD[1:0]	RW	时钟分频因子 配置定时器时钟 CK_INT 频率与数字滤波器使用的采样频率之间的分频关系 00: $t_{DTS}=t_{CK_INT}$ 01: $t_{DTS}=2*t_{CK_INT}$ 10: $t_{DTS}=4*t_{CK_INT}$ 11: 保留
7	ARPE	RW	自动重装载预装载使能 0: ARR 寄存器没有预装载 1: ARR 寄存器开启预装载

位域	位域名称	访问	描述
6:5	CMS[1:0]	RW	计数模式选择 00:边沿对齐模式。计数器依据方向位(dir)向上或向下计数。 01:中央对齐模式 1。计数器交替地向上或向下计数。配置为输出的通道的输出比较标志位，只在计数器向下计数时被设置。 10:中央对齐模式 2。计数器交替地向上或向下计数。配置为输出的通道的输出比较标志位，只在计数器向上计数时被设置。 11:中央对齐模式 3。计数器交替地向上或向下计数。配置为输出的通道的输出比较标志位，在计数器向上和向下计数时均被设置。
4	DIR	RW	方向 0:计数器向上计数 1:计数器向下计数 注：当计数器配置为中央对齐模式或编码器模式时，该位由硬件自动配置，只读
3	OPM	RW	单脉冲模式 0: 关闭单脉冲模式 1:发生更新事件时，清除 CEN 位，停止计数
2	URS	RW	更新请求源 0: 计数器上溢或下溢、设置 UG 位、通过从模式控制器产生的更新均可产生更新中断或 DMA 请求 1:只有计数器上溢或下溢可以产生更新中断或者 DMA 请求
1	UDIS	RW	禁止更新 0:计数器上溢或下溢、设置 UG 位、通过从模式控制器产生的更新可产生更新事件。产生更新事件后，开启预装载的寄存器被加载为预装载值。 1:禁止更新事件产生。此时开启预装载的寄存器将保持其内容无法被更改，但通过设置 UG 位或从模式控制器产生了硬件复位，计数器和预分频器将被重新初始化。
0	CEN	RW	计数使能 0:停止计数 1:使能计数

18.3.3 ATIM_CR2

中文名： ATIM 控制寄存器 2

寄存器位宽： [31: 0]

偏移量： 0x04

复位值： 0x00000000

表 18- 4 控制寄存器 2

位域	位域名称	访问	描述
----	------	----	----

位域	位域名称	访问	描述
31:15	Reserved	—	保留
14	IOS4	RW	空闲输出状态 4 参见 IOS1 位
13	IOS3N	RW	互补空闲输出状态 3 参见 IOS1N 位
12	IOS3	RW	空闲输出状态 3 参见 IOS1 位
11	IOS2N	RW	互补空闲输出状态 2 参见 IOS1N 位
10	IOS2	RW	空闲输出状态 2 参见 IOS1 位
9	IOS1N	RW	互补空闲输出状态 1 0: 当 MOE=0 且 OSI=1 时, 则死区后 OC1N=0 1: 当 MOE=0 且 OSI=1 时, 则死区后 OC1N=1 注 1: 为安全起见, 当 IOS1 与 IOS1N 同时为 1 时, OC1=OC1N=0 注 2: 只有当 OSI=1 时该控制位生效
8	IOS1	RW	空闲输出状态 1 0 : 当 MOE=0 且 OSI=1 时, 则死区后 OC1=0 1: 当 MOE=0 且 OSI=1 时, 则死区后 OC1=1
7	TI1S	RW	TI1 输入选择 0: CH1 引脚连接到 TI1 输入 1: CH1、CH2、CH3 引脚经异或后连到 TI1 输入
6: 4	MMS[2:0]	RW	主模式选择 该配置位用于选择在主模式下向从外设发送的触发输出来源。 000: 复位- 使用配置 UG 位和从模式控制器产生的硬件复位作为触发输出。 001: 使能- 使用计数器使能信号作为触发输出。计数器实际使能信号由 CEN 控制位与门控模式下触发输入共同决定。 010: 更新- 更新事件作为触发输出。 011: 比较脉冲- 当发生一次捕获或比较成功时, 当要设置 CC1IF 标志时 (即使已经为高), 触发输出一个脉冲。 100: OC1REF 信号被作为触发输出。 101: OC2REF 信号被作为触发输出。 110: OC3REF 信号被作为触发输出。 111: OC4REF 信号被作为触发输出。
3	CCDS	RW	捕获/比较的 DMA 选择 0: 当发生比较/捕获事件时, 发送 CCx 的 DMA 请求 1: 当发生更新事件时, 发送 CCx 的 DMA 请求
2	CCUS	RW	捕获/比较控制更新选择 0: 如果捕获/比较控制位是预装载的 (CCPC=1), 只能通过设

位域	位域名称	访问	描述
			置 COM 位被更新 1:如果捕获/比较控制位是预装载的(CCPC=1)，可以通过设置 COM 位或在 TRGI 上发生触发输入时更新
1	Reserved	—	保留
0	CCPS	RW	捕获/比较预装载控制 0:CCxE, CCxNE 和 OCxM 位不是预装载的 1:CCxE, CCxNE 和 OCxM 位是预装载的，设置该位后，只有在满足 CCUS 位条件的事件发生时才会被更新 注：该位只对具有互补输出的通道有效

18.3.4 ATIM_SMCR

中文名： ATIM 从模式控制寄存器

寄存器位宽： [31:0]

偏移量： 0x08

复位值： 0x00000000

表 18- 5 从模式控制寄存器

位域	位域名称	访问	描述
31:16	Reserved	—	保留
15	ETP	RW	外部触发极性 0:ETR 不反相 1:ETR 反相
14	ECE	RW	外部时钟使能位 0:禁止外部时钟模式 2 1:使能外部使能模式 2 注：复位模式、门控模式和触发模式可以与外部时钟模式 2 同时使用，但是此时 TRGI 不能连接到 ETR。当外部时钟模式 1 和外部时钟模式 2 同时被使能时，外部时钟的输入是 ETR。
13:12	Reserved	—	保留
11:8	ETF[3:0]	RW	外部触发滤波 配置对 ETR 信号采样的频率和数字滤波的带宽。数字滤波器是一个事件计数器，记录到 N 个事件后产生输出跳变。 0000:无滤波器，以 f_{DTS} 采样 0001:采样频率 $f_{SAMP}=f_{CK_INT}$ ，N=2 0010:采样频率 $f_{SAMP}=f_{CK_INT}$ ，N=4 0011:采样频率 $f_{SAMP}=f_{CK_INT}$ ，N=8 0100:采样频率 $f_{SAMP}=f_{DTS}/2$ ，N=6 0101:采样频率 $f_{SAMP}=f_{DTS}/2$ ，N=8 0110:采样频率 $f_{SAMP}=f_{DTS}/4$ ，N=6 0111:采样频率 $f_{SAMP}=f_{DTS}/4$ ，N=8 1000:采样频率 $f_{SAMP}=f_{DTS}/8$ ，N=6 1001:采样频率 $f_{SAMP}=f_{DTS}/8$ ，N=8

位域	位域名称	访问	描述
			1010: 采样频率 $f_{\text{SAMP}} = f_{\text{DTS}}/16$, N=5 1011: 采样频率 $f_{\text{SAMP}} = f_{\text{DTS}}/16$, N=6 1100: 采样频率 $f_{\text{SAMP}} = f_{\text{DTS}}/16$, N=8 1101: 采样频率 $f_{\text{SAMP}} = f_{\text{DTS}}/32$, N=5 1110: 采样频率 $f_{\text{SAMP}} = f_{\text{DTS}}/32$, N=6 1111: 采样频率 $f_{\text{SAMP}} = f_{\text{DTS}}/32$, N=8
7	TSYN	RW	触发同步 0: 关闭主从模式 1: 触发输入上的事件被延迟触发, 从而让当前定时器与被它的触发输出所控制的从定时器间达成同步
6:4	TS[2:0]	RW	触发来源选择 000: 内部触发 0, GTIM 001-011: 保留 100: TI1 的边沿检测器 101: 滤波后的 CH1 输入 110: 滤波后的 CH2 输入 111: 外部触发输入 ETR
3	Reserved	—	保留
2:0	SMS[2:0]	RW	从模式选择 000: 关闭从模式 001: 编码器模式 1-根据 TI1 的电平, 计数器在 TI2 的边沿向上/向下计数 010: 编码器模式 2-根据 TI2 的电平, 计数器在 TI1 的边沿向上/向下计数 011: 编码器模式 3-根据另一个信号的电平, 计数器在 TI1 和 TI2 的边沿向上/向下计数 100: 复位模式-在选中的触发输入 (TRGI) 的上升沿初始化计数器, 并产生一个更新寄存器的信号 101: 门控模式-当触发输入为高时, 计数器计数; 当触发输入为低时, 计数器停止计数但不复位。仅 CEN 位为 1 时门控模式有效 110: 触发模式-在触发输入的上升沿, CEN 位被硬件设置为 1 111: 外部时钟模式 1-在选中的触发输入的上升沿, 计数器计数

18.3.5 ATIM_DIER

中文名: ATIM DMA/中断使能寄存器
寄存器位宽: [31:0]
偏移量: 0x0c
复位值: 0x00000000

表 18- 6 DMA/中断使能寄存器

位域	位域名称	访问	描述
31:15	Reserved		保留
14	TDE	RW	触发事件 DMA 请求使能 0:禁止触发事件 DMA 请求 1:使能触发事件 DMA 请求
13	COMDE	RW	COM 事件 DMA 请求使能 0:禁止 COM 事件 DMA 请求 1:使能 COM 事件 DMA 请求
12	CC4DE	RW	捕获/请求通道 4 的 DMA 请求使能 0:禁止捕获/比较通道 4 的 DMA 请求 1:使能捕获/比较通道 4 的 DMA 请求
11	CC3DE	RW	捕获/请求通道 3 的 DMA 请求使能 0:禁止捕获/比较通道 3 的 DMA 请求 1:使能捕获/比较通道 3 的 DMA 请求
10	CC2DE	RW	捕获/请求通道 2 的 DMA 请求使能 0:禁止捕获/比较通道 2 的 DMA 请求 1:使能捕获/比较通道 2 的 DMA 请求
9	CC1DE	RW	捕获/请求通道 1 的 DMA 请求使能 0:禁止捕获/比较通道 1 的 DMA 请求 1:使能捕获/比较通道 1 的 DMA 请求
8	UDE	RW	更新事件 DMA 请求使能 0:禁止更新事件 DMA 请求 1:使能更新事件 DMA 请求
7	BIE	RW	刹车事件中断使能 0:禁止刹车事件中断 1:使能刹车事件中断
6	TIE	RW	触发事件中断使能 0:禁止触发事件中断 1:使能触发事件中断
5	COMIE	RW	COM 事件中断使能 0:禁止 COM 事件中断 1:使能 COM 事件中断
4	CC4IE	RW	捕获/请求通道 4 的中断使能 0:禁止捕获/比较通道 4 中断 1:使能捕获/比较通道 4 中断
3	CC3IE	RW	捕获/请求通道 3 的中断使能 0:禁止捕获/比较通道 3 中断 1:使能捕获/比较通道 3 中断
2	CC2IE	RW	捕获/请求通道 2 的中断使能 0:禁止捕获/比较通道 2 中断 1:使能捕获/比较通道 2 中断

位域	位域名称	访问	描述
1	CC1IE	RW	捕获/请求通道 1 的中断使能 0:禁止捕获/比较通道 1 中断 1:使能捕获/比较通道 1 中断
0	UIE	RW	更新事件中断使能 0:禁止更新事件中断 1:使能更新事件中断

18.3.6 ATIM_SR

中文名： ATIM 状态寄存器

寄存器位宽： [31:0]

偏移量： 0x10

复位值： 0x00000000

表 18- 7 状态寄存器

位域	位域名称	访问	描述
31:13	Reserved	—	保留
12	CC40F	R/W0	捕获/请求通道 4 重复捕获事件标志位 仅当相应通道被配置为输入捕获时有意义 0:无重复捕获事件 1:通道 4 发生重复捕获事件
11	CC30F	R/W0	捕获/请求通道 3 重复捕获事件标志位 0:无重复捕获事件 1:通道 3 发生重复捕获事件
10	CC20F	R/W0	捕获/请求通道 2 重复捕获事件标志位 0:无重复捕获事件 1:通道 2 发生重复捕获事件
9	CC10F	R/W0	捕获/请求通道 1 重复捕获事件标志位 0:无重复捕获事件 1:通道 1 发生重复捕获事件
8	Reserved	—	保留
7	BIF	R/W0	刹车事件标志位 硬件在产生刹车事件发生时设置该位，软件写 0 清除 0:未产生刹车事件 1:产生了刹车事件
6	TIF	R/W0	触发事件标志位 当发生触发事件（当从模式控制器处于除门控模式外的其他模式在触发输入端检测到有效边沿，或门控模式下任一边沿）时硬件置 1，软件写 0 清除。 0:无触发事件发生 1:发生触发事件
5	COMIF	R/W0	COM 事件标志位

位域	位域名称	访问	描述
			硬件在产生 COM 事件发生时设置该位，软件写 0 清除 0:未产生 COM 事件 1:产生了 COM 事件
4	CC4IF	R/W0	捕获/请求通道 4 事件标志位 0:无捕获/比较事件 1:通道 4 发生捕获/比较通道事件
3	CC3IF	R/W0	捕获/请求通道 3 事件标志位 0:无捕获/比较事件 1:通道 3 发生捕获/比较通道事件
2	CC2IF	R/W0	捕获/请求通道 2 事件标志位 0:无捕获/比较事件 1:通道 2 发生捕获/比较通道事件
1	CC1IF	R/W0	捕获/请求通道 1 事件标志位 0:无捕获/比较事件 1:通道 1 发生捕获/比较通道事件
0	UIF	R/W0	更新事件标志位 硬件在产生更新事件时设置该位，软件写 0 清除 0:未产生更新事件 1:产生了更新事件

18.3.7 ATIM_EGR

中文名: ATIM 事件产生寄存器

寄存器位宽: [31:0]

偏移量: 0x14

复位值: 0x00000000

表 18- 8 事件产生寄存器

位域	位域名称	访问	描述
31:8	Reserved	—	保留
7	BG	W	产生刹车事件 该位始终读为 0, 写 1 产生触发事件
6	TG	W	产生触发事件 该位始终读为 0, 写 1 产生触发事件
5	COMG	W	产生 COM 事件 该位始终读为 0, 写 1 产生触发事件
4	CC4G	W	产生捕获/比较 4 事件 该位始终读为 0, 写 1 产生捕获/比较事件。 若通道为输出:设置 CC1IF 为 1, 若开启对应的中断和 DMA, 则产生相应的中断和 DMA 若通道为输入:将当前计数器 CNT 的值捕获置 CCR1 寄存器, 设置 CC1IF 为 1, 若 CC1IF 已经为 1, 则设置 CC1OF 为 1

位域	位域名称	访问	描述
3	CC3G	W	产生捕获/比较 3 事件 该位始终读为 0, 写 1 产生捕获/比较事件
2	CC2G	W	产生捕获/比较 2 事件 该位始终读为 0, 写 1 产生捕获/比较事件
1	CC1G	W	产生捕获/比较 1 事件 该位始终读为 0, 写 1 产生捕获/比较事件
0	UG	W	产生更新事件 该位始终读为 0, 写 1 产生更新事件, 初始化计数器并更新带有预装载的寄存器

18.3.8 ATIM_CCMR1 (OUT)

中文名: ATIM 捕获/比较模式寄存器 1 (输出)

寄存器位宽: [31:0]

偏移量: 0x18

复位值: 0x00000000

表 18- 9 捕获/比较模式寄存器 1 (输出)

位域	位域名称	访问	描述
31:16	Reserved	—	保留
15	OC2CE	RW	输出比较 2 清零使能 0: 无影响 1: 一旦检测到 ETR 输入高电平, 清除 OC2REF=0
14:12	OC2M[2:0]	RW	输出比较 2 模式 000: 冻结, OC2REF 保持当前状态不变 001: 当计数器 CNT 的值与捕获/比较寄存器 CCR2 相同时, 强制 OC2REF 为高电平 010: 当计数器 CNT 的值与捕获/比较寄存器 CCR2 相同时, 强制 OC2REF 为低电平 011: 当计数器 CNT 的值与捕获/比较寄存器 CCR2 相同时, 翻转 OC2REF 的电平 100: 强制 OC2REF 为低电平 101: 强制 OC2REF 为高电平 110: PWM 模式 1-在向上计数时, 一旦 CNT<CCR2, OC2REF 为高电平, 否则为低电平; 在向下计数时, 一旦 CNT>CCR2, OC2REF 为低电平, 否则为高电平。 111: PWM 模式 2-在向上计数时, 一旦 CNT<CCR2, OC2REF 为低电平, 否则为高电平; 在向下计数时, 一旦 CNT>CCR2, OC2REF 为高电平, 否则为低电平。
11	OC2PE	RW	输出比较 2 预装载使能 0: 关闭 CCR2 寄存器的预装载功能 1: 开启 CCR2 寄存器的预装载功能
10	OC2FE	RW	输出比较 2 快速使能 0: 无影响

位域	位域名称	访问	描述
			1: 仅在单脉冲模式下且通道配置为 PWM1/PWM2 模式时生效, 当开始计数时, 立即输出有效电平, OC2REF 电平此时与比较结果无关
9:8	CC2S[1:0]	RW	输出/比较 2 选择 00: CC2 通道被配置为输出 01: CC2 通道被配置为输入, IC2 映射在 TI2 上 10: CC2 通道被配置为输入, IC2 映射在 TI1 上 11: CC2 通道被配置为输入, IC2 映射在 TRC 上
7	OC1CE	RW	输出比较 1 清零使能 0: 无影响 1: 一旦检测到 ETR 输入高电平, 清除 OC1REF=0
6:4	OC1M[2:0]	RW	输出比较 1 模式 参见 OC2M 说明
3	OC1PE	RW	输出比较 1 预装载使能 0: 关闭 CCR1 寄存器的预装载功能 1: 开启 CCR1 寄存器的预装载功能
2	OC1FE	RW	输出比较 1 快速使能 参见 OC2FE 说明
1:0	CC1S[1:0]	RW	输出/比较 1 选择 00: CC1 通道被配置为输出 01: CC1 通道被配置为输入, IC1 映射在 TI1 上 10: CC1 通道被配置为输入, IC1 映射在 TI2 上 11: CC1 通道被配置为输入, IC1 映射在 TRC 上

18.3.9 ATIM_CCMR1 (IN)

中文名: ATIM 捕获/比较模式寄存器 1 (输入)

寄存器位宽: [31:0]

偏移量: 0x18

复位值: 0x00000000

表 18- 10 捕获/比较模式寄存器 1 (输入)

位域	位域名称	访问	描述
31:16	Reserved	-	保留

位域	位域名称	访问	描述
15:12	IC2F[3:0]	RW	<p>输入捕获 2 滤波器</p> <p>配置对 TI2 输入信号采样的频率和数字滤波的带宽。数字滤波器是一个事件计数器，记录到 N 个事件后产生输出跳变。</p> <p>0000:无滤波器，以 f_{DTS} 采样</p> <p>0001:采样频率 $f_{SAMP}=f_{CK_INT}$，N=2</p> <p>0010:采样频率 $f_{SAMP}=f_{CK_INT}$，N=4</p> <p>0011:采样频率 $f_{SAMP}=f_{CK_INT}$，N=8</p> <p>0100:采样频率 $f_{SAMP}=f_{CK_INT}/2$，N=6</p> <p>0101:采样频率 $f_{SAMP}=f_{CK_INT}/2$，N=8</p> <p>0110:采样频率 $f_{SAMP}=f_{CK_INT}/4$，N=6</p> <p>0111:采样频率 $f_{SAMP}=f_{CK_INT}/4$，N=8</p> <p>1000:采样频率 $f_{SAMP}=f_{CK_INT}/8$，N=6</p> <p>1001:采样频率 $f_{SAMP}=f_{CK_INT}/8$，N=8</p> <p>1010:采样频率 $f_{SAMP}=f_{CK_INT}/16$，N=5</p> <p>1011:采样频率 $f_{SAMP}=f_{CK_INT}/16$，N=6</p> <p>1100:采样频率 $f_{SAMP}=f_{CK_INT}/16$，N=8</p> <p>1101:采样频率 $f_{SAMP}=f_{CK_INT}/32$，N=5</p> <p>1110:采样频率 $f_{SAMP}=f_{CK_INT}/32$，N=6</p> <p>1111:采样频率 $f_{SAMP}=f_{CK_INT}/32$，N=8</p>
11:10	IC2PSC[1:0]	RW	<p>输入捕获 2 预分频器</p> <p>定义 CC2 输入（IC2）的预分频系数</p> <p>00:无预分频</p> <p>01:每 2 个事件触发一次捕获</p> <p>10:每 4 个事件触发一次捕获</p> <p>11:每 8 个事件触发一次捕获</p>
9:8	CC2S[1:0]	RW	<p>输出/比较 2 选择</p> <p>00:CC2 通道被配置为输出</p> <p>01:CC2 通道被配置为输入，IC2 映射在 TI2 上</p> <p>10:CC2 通道被配置为输入，IC2 映射在 TI1 上</p> <p>11:CC2 通道被配置为输入，IC2 映射在 TRC 上</p>
7:4	IC1F[3:0]	RW	<p>输入捕获 1 滤波器</p> <p>参考 IC2F 说明</p>
3:2	IC1PSC[1:0]	RW	<p>输入捕获 1 预分频器</p> <p>参考 IC2PSC 说明</p>
1:0	CC1S[1:0]	RW	<p>输出/比较 1 选择</p> <p>00:CC1 通道被配置为输出</p> <p>01:CC1 通道被配置为输入，IC1 映射在 TI1 上</p> <p>10:CC1 通道被配置为输入，IC1 映射在 TI2 上</p> <p>11:CC1 通道被配置为输入，IC1 映射在 TRC 上</p>

18.3.10 ATIM_CCMR2(OUT)

中文名: ATIM 捕获/比较模式寄存器 2(输出)

寄存器位宽: [31:0]

偏移量: 0x1c

复位值: 0x00000000

表 18- 11 捕获/比较模式寄存器 2(输出)

位域	位域名称	访问	描述
31:16	Reserved	—	保留
15	OC4CE	RW	输出比较 4 清零使能 0:无影响 1:一旦检测到 ETR 输入高电平, 清除 OC4REF=0
14:12	OC4M[2:0]	RW	输出比较 4 模式 000:冻结, OC4REF 保持当前状态不变 001:当计数器 CNT 的值与捕获/比较寄存器 CCR4 相同时, 强制 OC4REF 为高电平 010:当计数器 CNT 的值与捕获/比较寄存器 CCR4 相同时, 强制 OC4REF 为低电平 011:当计数器 CNT 的值与捕获/比较寄存器 CCR4 相同时, 翻转 OC4REF 的电平 100:强制 OC4REF 为低电平 101:强制 OC4REF 为高电平 110:PWM 模式 1-在向上计数时, 一旦 CNT<CCR4, OC4REF 为高电平, 否则为低电平;在向下计数时, 一旦 CNT>CCR4, OC4REF 为低电平, 否则为高电平。 111:PWM 模式 2-在向上计数时, 一旦 CNT<CCR4, OC4REF 为低电平, 否则为高电平;在向下计数时, 一旦 CNT>CCR4, OC4REF 为高电平, 否则为低电平。
11	OC4PE	RW	输出比较 4 预装载使能 0:关闭 CCR4 寄存器的预装载功能 1:开启 CCR4 寄存器的预装载功能
10	OC4FE	RW	输出比较 4 快速使能 0:无影响 1:仅在单脉冲模式下且通道配置为 PWM1/PWM2 模式时生效, 当开始计数时, 立即输出有效电平, OC4REF 电平此时与比较结果无关
9:8	CC4S[1:0]	RW	输出/比较 4 选择 00:CC4 通道被配置为输出 01:CC4 通道被配置为输入, IC4 映射在 TI4 上 10:CC4 通道被配置为输入, IC4 映射在 TI3 上 11:CC4 通道被配置为输入, IC4 映射在 TRC 上
7	OC3CE	RW	输出比较 3 清零使能 0:无影响 1:一旦检测到 ETR 输入高电平, 清除 OC3REF=0

位域	位域名称	访问	描述
6:4	OC3M[2:0]	RW	输出比较 3 模式 参见 OC4M 说明
3	OC3PE	RW	输出比较 3 预装载使能 0:关闭 CCR3 寄存器的预装载功能 1:开启 CCR3 寄存器的预装载功能
2	OC3FE	RW	输出比较 3 快速使能 参见 OC4FE 说明
1:0	CC3S[1:0]	RW	输出/比较 3 选择 00:CC3 通道被配置为输出 01:CC3 通道被配置为输入, IC3 映射在 TI3 上 10:CC3 通道被配置为输入, IC3 映射在 TI4 上 11:CC3 通道被配置为输入, IC3 映射在 TRC 上

18.3.11 ATIM_CCMR2(IN)

中文名: ATIM 捕获/比较模式寄存器 2(输入)

寄存器位宽: [31:0]

偏移量: 0x1c

复位值: 0x00000000

表 18- 12 捕获/比较模式寄存器 2(输入)

位域	位域名称	访问	描述
31:16	Reserved	-	保留
15:12	IC4F[3:0]	RW	<p>输入捕获 4 滤波器</p> <p>配置对 TI4 输入信号采样的频率和数字滤波的带宽。数字滤波器是一个事件计数器,记录到 N 个事件后产生输出跳变。</p> <p>0000:无滤波器,以 f_{DTS} 采样</p> <p>0001:采样频率 $f_{SAMP}=f_{CK_INT}$, N=2</p> <p>0010:采样频率 $f_{SAMP}=f_{CK_INT}$, N=4</p> <p>0011:采样频率 $f_{SAMP}=f_{CK_INT}$, N=8</p> <p>0100:采样频率 $f_{SAMP}=f_{CK_INT}/2$, N=6</p> <p>0101:采样频率 $f_{SAMP}=f_{CK_INT}/2$, N=8</p> <p>0110:采样频率 $f_{SAMP}=f_{CK_INT}/4$, N=6</p> <p>0111:采样频率 $f_{SAMP}=f_{CK_INT}/4$, N=8</p> <p>1000:采样频率 $f_{SAMP}=f_{CK_INT}/8$, N=6</p> <p>1001:采样频率 $f_{SAMP}=f_{CK_INT}/8$, N=8</p> <p>1010:采样频率 $f_{SAMP}=f_{CK_INT}/16$, N=5</p> <p>1011:采样频率 $f_{SAMP}=f_{CK_INT}/16$, N=6</p> <p>1100:采样频率 $f_{SAMP}=f_{CK_INT}/16$, N=8</p> <p>1101:采样频率 $f_{SAMP}=f_{CK_INT}/32$, N=5</p> <p>1110:采样频率 $f_{SAMP}=f_{CK_INT}/32$, N=6</p> <p>1111:采样频率 $f_{SAMP}=f_{CK_INT}/32$, N=8</p>

位域	位域名称	访问	描述
11:10	IC4PSC[1:0]	RW	输入捕获 4 预分频器 定义 CC4 输入 (IC4) 的预分频系数 00:无预分频 01:每 2 个事件触发一次捕获 10:每 4 个事件触发一次捕获 11:每 8 个事件触发一次捕获
9:8	CC4S[1:0]	RW	输出/比较 4 选择 00:CC4 通道被配置为输出 01:CC4 通道被配置为输入, IC4 映射在 TI4 上 10:CC4 通道被配置为输入, IC4 映射在 TI3 上 11:CC4 通道被配置为输入, IC4 映射在 TRC 上
7:4	IC3F[3:0]	RW	输入捕获 3 滤波器 参考 IC4F 说明
3:2	IC3PSC[1:0]	RW	输入捕获 3 预分频器 参考 IC4PSC 说明
1:0	CC3S[1:0]	RW	输出/比较 3 选择 00:CC3 通道被配置为输出 01:CC3 通道被配置为输入, IC3 映射在 TI3 上 10:CC3 通道被配置为输入, IC3 映射在 TI4 上 11:CC3 通道被配置为输入, IC3 映射在 TRC 上

18.3.12 ATIM_CCER

中文名: ATIM 捕获/比较使能寄存器

寄存器位宽: [31:0]

偏移量: 0x20

复位值: 0x00000000

表 18- 13 捕获/比较使能寄存器

位域	位域名称	访问	描述
31:14	Reserved	—	保留
13	CC4P	RW	输入/捕获 4 极性 CC4 通道配置为输出: 0:OC4=OC4REF 1:OC4 为 OC4REF 反相输出 CC4 通道配置为输入: 0:不反相:捕获发生在 IC4 的上升沿 1:反相:捕获发生在 IC4 的下降沿

位域	位域名称	访问	描述
12	CC4E	RW	输入/捕获 4 使能 CC4 通道配置为输出： 0:OC4 禁止输出 1:OC4 信号输出到对应的引脚 CC4 通道配置为输入： 0:捕获禁止 1:捕获使能
11	CC3NP	RW	输入/捕获 3 互补输出极性 0: OC3N 高电平有效 1: OC3N 低电平有效
10	CC3NE	RW	输入/捕获 3 互补输出使能 0: 关闭— 关闭 OC1N 输出 1: 开启— 开启 OC1N 输出
9	CC3P	RW	输入/捕获 3 极性 参考 CC4P 描述
8	CC3E	RW	输入/捕获 3 使能 参考 CC4E 描述
7	CC2NP	RW	输入/捕获 2 互补输出极性 参考 CC3NP 描述
6	CC2NE	RW	输入/捕获 2 互补输出极性 参考 CC3NE 描述
5	CC2P	RW	输入/捕获 2 极性 参考 CC4P 描述
4	CC2E	RW	输入/捕获 2 使能 参考 CC4E 描述
3	CC1NP	RW	输入/捕获 1 互补输出极性 参考 CC3NP 描述
2	CC1NE	RW	输入/捕获 1 互补输出极性 参考 CC3NE 描述
1	CC1P	RW	输入/捕获 1 极性 参考 CC4P 描述
0	CC1E	RW	输入/捕获 1 使能 参考 CC4E 描述

18.3.13 ATIM_CNT

中文名： ATIM 计数器
寄存器位宽： [31:0]
偏移量： 0x24
复位值： 0x00000000

表 18- 14 计数器

位域	位域名称	访问	描述
31:0	CNT	RW	计数器数值

18.3.14 ATIM_PSC

中文名： ATIM 预分频器

寄存器位宽： [31:0]

偏移量： 0x28

复位值： 0x00000000

表 18- 15 预分频器

位域	位域名称	访问	描述
31:0	PSC	RW	<p>预分频器数值</p> <p>计数器的实际时钟频率为 $f_{CK_INT}/(psc+1)$</p> <p>预分频器始终开启预装载功能，在发生更新事件时，psc 的数值被装载到实际的预分频寄存器中。</p>

18.3.15 ATIM_ARR

中文名： ATIM 自动重装载寄存器

寄存器位宽： [31:0]

偏移量： 0x2c

复位值： 0x00000000

表 18- 16 自动重装载寄存器

位域	位域名称	访问	描述
31:0	ARR	RW	<p>自动重装载数值</p> <p>当计数器 cnt 与 arr 相等时，产生上溢事件，并根据相关配置位产生更新事件。</p> <p>注：当 arr 为 0 时，计数器停止。</p>

18.3.16 ATIM_RCR

中文名： ATIM 重复计数寄存器

寄存器位宽： [31:0]

偏移量： 0x30

复位值： 0x00000000

表 18- 17 重复计数寄存器

位域	位域名称	访问	描述
31:8	Reserved	—	保留
7:0	RCR	RW	<p>重复计数器值</p> <p>这些位允许用户设置更新事件的产生频率。</p> <p>在产生 RCR 次计数器上/下溢事件后产生一次更新事件。</p> <p>重复计数器始终开启预装载功能，在发生更新事件时，RCR 的数值被装载到实际的重复计数寄存器中。</p>

18.3.17 ATIM_CCR1

中文名： ATIM 捕获/比较寄存器 1
寄存器位宽： [31:0]
偏移量： 0x34
复位值： 0x00000000

表 18- 18 捕获/比较寄存器 1

位域	位域名称	访问	描述
31:0	CCR1	RW	捕获/比较 1 数值 CC1 通道配置为输出： CCR1 包含了装入当前捕获/比较 1 寄存器的值(预装载值) CC1 通道配置为输入： CCR1 包含了上次输入捕获 1 事件时传输的计数器 CNT 值

18.3.18 ATIM_CCR2

中文名： ATIM 捕获/比较寄存器 2
寄存器位宽： [31:0]
偏移量： 0x38
复位值： 0x00000000

表 18- 19 捕获/比较寄存器 2

位域	位域名称	访问	描述
31:0	CCR2	RW	捕获/比较 2 数值 CC2 通道配置为输出： CCR2 包含了装入当前捕获/比较 2 寄存器的值(预装载值) CC2 通道配置为输入： CCR2 包含了上次输入捕获 2 事件时传输的计数器 CNT 值

18.3.19 ATIM_CCR3

中文名： ATIM 捕获/比较寄存器 3
寄存器位宽： [31:0]
偏移量： 0x3c
复位值： 0x00000000

表 18- 20 捕获/比较寄存器 3

位域	位域名称	访问	描述
31:0	CCR3	RW	捕获/比较 3 数值 CC3 通道配置为输出： CCR3 包含了装入当前捕获/比较 3 寄存器的值(预装载值) CC3 通道配置为输入： CCR3 包含了上次输入捕获 3 事件时传输的计数器 CNT 值

18.3.20 ATIM_CCR4

中文名： ATIM 捕获/比较寄存器 4

寄存器位宽： [31:0]

偏移量： 0x40

复位值： 0x00000000

表 18- 21 捕获/比较寄存器 4

位域	位域名称	访问	描述
31:0	CCR4	RW	捕获/比较 4 数值 CC4 通道配置为输出： CCR4 包含了装入当前捕获/比较 4 寄存器的值(预装载值) CC4 通道配置为输入： CCR4 包含了上次输入捕获 4 事件时传输的计数器 CNT 值

18.3.21 ATIM_BDTR

中文名： ATIM 刹车和死区寄存器

寄存器位宽： [31:0]

偏移量： 0x44

复位值： 0x00000000

表 18- 22 刹车和死区寄存器

位域	位域名称	访问	描述
31:16	Reserved	—	保留
15	MOE	RW	主输出使能 一旦刹车输入有效，该位被硬件清 0。根据 AOE 位的设置值，该位可以由软件清 0 或被自动置 1。它仅对配置为输出的通道有效。 0：禁止 OC 和 OCN 输出； 1：如果设置了相应的使能位，则开启 OC 和 OCN 输出
14	AOE	RW	自动输出使能 0：MOE 只能被软件置 1； 1：MOE 能被软件置 1 或在下一个更新事件被自动置 1(如果刹车输入无效)。
13	BKP	RW	刹车输入极性 0：刹车输入低电平有效； 1：刹车输入高电平有效。
12	BKE	RW	刹车输入使能 0：禁止刹车输入； 1：使能刹车输入。

位域	位域名称	访问	描述
11	OSR	RW	运行模式下输出状态选择 该位用于当 MOE=1 且通道为互补输出时。 0: 当 CCxE=0 或 CCxNE=0 时, 禁止 OC/OCN 输出; 1: 当 CCxE=0 或 CCxNE=0 时, 开启 OC/OCN 并输出无效电平
10	OSI	RW	关闭模式下输出状态选择 该位用于当 MOE=0 且通道为互补输出时。 0: 禁止 OC/OCN 输出; 1: 当 CCxE=1 或 CCxNE=1 时, 开启 OC 与 OCN 并输出空闲电平。
9:8	LOCK[1:0]	RW	锁定设置 该位为防止软件错误而提供写保护。 00: 锁定关闭, 寄存器无写保护; 01: 锁定级别 1, 不能写入 BDTR 寄存器的 DTG、BKE、BKP、AOE 位和 CR2 寄存器的 IOSx/IOSxN 位; 10: 锁定级别 2, 不能写入锁定级别 1 中的各位, 也不能写入 CC 极性位以及 OSR/OSI 位; 11: 锁定级别 3, 不能写入锁定级别 2 中的各位, 也不能写入 CC 控制位; 注: 在系统复位后, 只能写一次 LOCK 位, 一旦写入 BDTR 寄存器, 则其内容冻结直至复位。
7:0	UTG[7:0]	RW	死区发生器设置 这些位定义了插入互补输出之间的死区持续时间。假设 DT 表示其持续时间: $DTG[7:5]=0xx \Rightarrow DT=DTG[7:0] * T_{dtg}, T_{dtg} = T_{DTS};$ $DTG[7:5]=10x \Rightarrow DT=(64+DTG[5:0]) * T_{dtg}, T_{dtg} = 2 * T_{DTS};$ $DTG[7:5]=110 \Rightarrow DT=(32+DTG[4:0]) * T_{dtg}, T_{dtg} = 8 * T_{DTS};$ $DTG[7:5]=111 \Rightarrow DT=(32+DTG[4:0]) * T_{dtg}, T_{dtg} = 16 * T_{DTS};$

18.3.22 ATIM_INSTA

中文名: ATIM 输入通道状态寄存器

寄存器位宽: [31:0]

偏移量: 0x50

复位值: 0x00000000

表 18- 23 输入通道状态寄存器

位域	位域名称	访问	描述
31:6	Reserved	—	保留

位域	位域名称	访问	描述
5	BKIN	R	BKIN 输入
4	ETR_IN	R	ETR 输入
3	CH4_IN	R	CH4 输入
2	CH3_IN	R	CH3 输入
1	CH2_IN	R	CH2 输入
0	CH1_IN	R	CH1 输入

19 GTIM 控制器

19.1 概述

龙芯 2K0300 集成了一个由 32 位自动装载计数器驱动的 GTIM。

它适用于多种应用，如测量输入信号的脉冲长度或产生指定的输出波形。GTIM 支持编码器模式与霍尔传感器模式，可为电机相关应用提供良好的支持。

GTIM 的时钟由内部 APB 总线时钟提供，基地址为 0x16119000，寄存器定义见下文。

19.2 功能描述

19.2.1 计数模式

定时器具有向上、向下、中央对齐 3 种计数模式，其中单向计数模式与中央对齐模式通过 CR1 寄存器中的 CMS 配置位进行选择。在单向计数模式中，用户通过配置 CR1 寄存器中的 DIR 位选择向上/向下计数。在中央对齐模式中，DIR 位只读，由硬件自动设置用于指示当前的计数方向。

用户可通过配置 PSC 寄存器对计数器时钟进行预分频，配置 ARR 寄存器用于配置计数周期，且可对 CNT 计数器实时进行读取与修改，实际完成一次计数周期的时长由 PSC 与 ARR 共同决定。

PSC 与 ARR 寄存器具有预装载功能，即当次的配置值会留到下个计数周期/更新事件产生时才生效，其中 PSC 寄存器预装载功能始终打开，ARR 寄存器预装载功能可自由选择开启或关闭。

GTIM 模块中 CNT 的计数时钟默认由内部 APB 总线时钟提供，用户也可以通过配置 SMCR 寄存器以使用外部输入的脉冲信号或芯片内部互联信号来进行计数，以实现更灵活的应用。

每次计数器溢出、设置 UG 位、从模式控制器产生硬件复位时可根据相关配置位配置产生更新事件。

当发生更新事件时，PSC 内部计数被清零，CNT 根据计数模式进行重装载，同时：

- SR 寄存器中的 UIF 标志位被设置。
- PSC 寄存器中的值被装载到实际的内部寄存器。
- 若设置了 ARPE 位，ARR 寄存器中的值被装载到实际的自动重装载寄存器。

通过设置 CR1 寄存器中的 UDIS 位，可以禁止除写入 UG 位外的更新事件产生。如设置了 CR1 寄存器中的 URS 位，通过设置 EGR 寄存器中的 UG 位可以手动产生一个更新事件，但不设置 UIF 标志位。

19.2.2 输入模式

GTIM 具有 4 个独立的输入/输出通道，每个通道都可独立地选择输入/输出模式，但同一个通道同时只能选择输入或输出模式中的一种。

在输入模式下，当检测到 CHx 信号上相应的边沿后，计数器的当前值被锁存到捕获/比较寄存器（CCR_x）中。当捕获事件发生时，SR 寄存器中相应通道的 CC_xIF 标志被置 1，如果在 DIER 中使能了相应的中断或 DMA，则将产生中断或 DMA 请求。如果捕获事件发生时 CC_xIF 标志已经为 1，那么 SR 寄存器中的重复捕获标志 CC_xOF 标志将被置 1，该标志有助于用户判断是否存在被错过的捕获事件。

写 CC_xIF=0 可清除 CC_xIF，或读取存储在对应通道的 CCR_x 寄存器中的数据也可清除 CC_xIF。写 CC_xOF=0 可清除 CC_xOF。

注：设置 EGR 寄存器中相应的 CC_xG 位，可以通过软件输入产生输入捕获事件，同时根据 DIER 寄存器的配置产生中断或 DMA 请求。

19.2.2.1 编码器接口模式

在编码器接口模式是输入模式下的一种特殊应用，可连接外部增量式编码器用于计数。

两个输入 CH1 和 CH2 被用来作为增量编码器的接口。根据两个输入信号的跳变顺序，产生了计数脉冲和方向信号，计数器向上或向下计数，同时硬件对 CR1 寄存器的 DIR 位进行相应的设置。

选择编码器接口模式的方法是：如果计数器只在 CH2 的边沿计数，则置 SMCR 寄存器中的 SMS=001；如果只在 CH1 边沿计数，则置 SMS=010；如果计数器同时在 CH1 和 CH2 边沿计数，则置 SMS=011。

编码器接口模式在开始计数之前必须配置 ARR 与其他寄存器，其配置与通常的计数模式保持一致。

在这个模式下，计数器依照增量编码器的速度和方向被自动的修改，因此计数器的内容始终指示着编码器的位置。计数方向与相连的传感器旋转的方向对应。

表 19- 1 工作模式配置

有效边沿	相对信号电平	CH1FP1		CH2FP2	
		上升沿	下降沿	上升沿	下降沿
仅在 CH1 计数	高	向下计数	向上计数	不计数	不计数
	低	向上计数	向下计数	不计数	不计数
仅在 CH2 计数	高	不计数	不计数	向上计数	向下计数
	低	不计数	不计数	向下计数	向上计数
在 CH1 和 CH2 均计数	高	向下计数	向上计数	向上计数	向下计数
	低	向上计数	向下计数	向下计数	向上计数

19.2.2.2 霍尔传感器模式

定时器为外部霍尔传感器应用进行优化。

通过配置 CR2 寄存器中的 TI1S 位, 可将 CH1、CH2、CH3 输入异或后作为通道 1 输入, 因此外部霍尔传感器的所有边沿均可通过通道 1 触发内部中断, 进而读取外部霍尔传感器输入电平确定转子位置, 完成换相配置。

19.2.3 输出模式

该功能用于控制一个输出波形, 或者通过电平变化指示时间的变化。其中通道 1-3 具有互补输出通道。在输出模式下捕获/比较寄存器可开启预装载功能。

19.2.3.1 电平输出模式

当计数器 CNT 与 CCRx 寄存器的值相同时, 即称为发生比较匹配事件, 此时 OCxREF 信号根据所配置的输出模式, 可以保持原本电平 (OCxM=000)、被设置为有效电平 (OCxM=001)、被设置为无效电平 (OCxM=010)、始终输出低电平 (OCxM=0x100)、始终输出高电平 (OCxM=0x101) 或将当前电平状态进行翻转 (OCxM=011), 再根据 CCER 寄存器中 CCxP 输出极性配置, 将符合期望的值输出到对应的引脚上。同时 SR 寄存器中对应的 CCxIF 标志位被设置, 并根据 DIER/CR2 寄存器相关配置产生中断或 DMA 请求。

例如: 配置 OC1M=101, CC1P=0, 即可强制 OC1 输出高电平。

19.2.3.2 PWM 模式

脉冲宽度调制模式可以产生一个由 ARR 寄存器确定频率、由 CCRx 寄存器确定占空比的信号。

在 CCMRx 寄存器中的 OCxM 位写入 110 (PWM 模式 1) 或 111 (PWM 模式 2), 能够独立地设置每个 OCx 输出通道产生一路 PWM。用户可设置 CCMRx 寄存器 OCxPE 位以使能相应的预装载寄存器。

仅当发生一个更新事件的时候, 被配置为预装载模式的寄存器中的配置值才可生效, 因此在计数器开始计数之前, 必须通过设置 EGR 寄存器中的 UG 位来初始化所有的寄存器。

CCER 寄存器中的 CCxP/CCxE 位控制 OCx 输出极性与使能。

根据 CR1 寄存器中 CMS 位的状态, 定时器能够产生边沿对齐的 PWM 信号或中央对齐的 PWM 信号。

当 CR1 寄存器中的 DIR 位为低的时候执行向上计数。以 PWM 模式 1 为例, 当 $CNT < CCRx$ 时 PWM 信号参考 OCxREF 为高, 否则为低。如果 CCRx 中的比较值大于自动重装载值 (ARR), 则 OCxREF 保持为 1。如果 CCRx 中的比较值为 0, 则 OCxREF 保持为 0。

当 CR1 寄存器的 DIR 位为高时执行向下计数。在 PWM 模式 1 时, 当 $CNT > CCRx$ 时参考信号 OCxREF 为低, 否则为高。如果 CCRx 中的比较值大于 ARR 中的自动重装载值, 则 OCxREF 保持为 1。该模式下不能产生占空比为 0% 的 PWM 波形。

对于一个给定的通道, 设置 CCMRx 寄存器中对应的 OCxCE 位为 1, 能够用 ETRF 输入端的

高电平把 OCxREF 信号拉低, OCxREF 信号将保持为低直到发生下一次的更新事件 UEV。

该功能只能用于输出比较和 PWM 模式, 而不能用于强置模式。

将 CR1 寄存器中的 OPM 位置为 1 将选择单脉冲模式, 这样可以让定时器自动地在产生下一个更新事件 UEV 时清除 CEN 位, 从而停止计数。这种模式可用于让定时器响应一个激励, 并在一个程序可控的延时之后, 产生一个脉宽可程序控制的脉冲。

19.2.4 定时器外部控制

通过配置 SMCR 寄存器中的 SMS 位, 可使用外部信号对定时器进行一定程度的控制。

19.3 寄存器描述

19.3.1 寄存器地址列表

表 19- 2 GTIM 寄存器列表

名称	偏移地址	位宽	描述
GTIM_CR1	0x00	32	控制寄存器 1
GTIM_CR2	0x04	32	控制寄存器 2
GTIM_SMCR	0x08	32	从模式控制寄存器
GTIM_DIER	0x0C	32	DMA/中断使能寄存器
GTIM_SR	0x10	32	状态寄存器
GTIM_EGR	0x14	32	事件产生寄存器
GTIM_CCMR1	0x18	32	捕获/比较模式寄存器 1
GTIM_CCMR2	0x1C	32	捕获/比较模式寄存器 2
GTIM_CCER	0x20	32	捕获/比较使能寄存器
GTIM_CNT	0x24	32	计数器
GTIM_PSC	0x28	32	预分频器
GTIM_ARR	0x2C	32	自动重装载寄存器
GTIM_CCR1	0x34	32	捕获/比较寄存器 1
GTIM_CCR2	0x38	32	捕获/比较寄存器 2
GTIM_CCR3	0x3C	32	捕获/比较寄存器 3
GTIM_CCR4	0x40	32	捕获/比较寄存器 4
GTIM_INSTA	0x50	32	输入通道状态寄存器

19.3.2 GTIM_CR1

中文名: GTIM 控制寄存器 1

寄存器位宽: [31:0]

偏移量: 0x00

复位值: 0x00000000

表 19- 3 控制寄存器 1

位域	位域名称	访问	描述
31:10	Reserved	—	保留
9: 8	CKD[1:0]	RW	<p>时钟分频因子</p> <p>配置定时器时钟 CK_INT 频率与数字滤波器使用的采样频率之间的分频关系</p> <p>00: $t_{DTS}=t_{CK_INT}$</p> <p>01: $t_{DTS}=2*t_{CK_INT}$</p> <p>10: $t_{DTS}=4*t_{CK_INT}$</p> <p>11: 保留</p>
7	ARPE	RW	<p>自动重装载预装载使能</p> <p>0: ARR 寄存器没有预装载</p> <p>1: ARR 寄存器开启预装载</p>
6:5	CMS[1:0]	RW	<p>计数模式选择</p> <p>00: 边沿对齐模式。计数器依据方向位(dir)向上或向下计数。</p> <p>01: 中央对齐模式 1。计数器交替地向上或向下计数。配置为输出的通道的输出比较标志位, 只在计数器向下计数时被设置。</p> <p>10: 中央对齐模式 2。计数器交替地向上或向下计数。配置为输出的通道的输出比较标志位, 只在计数器向上计数时被设置。</p> <p>11: 中央对齐模式 3。计数器交替地向上或向下计数。配置为输出的通道的输出比较标志位, 在计数器向上和向下计数时均被设置。</p>
4	DIR	RW	<p>方向</p> <p>0: 计数器向上计数</p> <p>1: 计数器向下计数</p> <p>注: 当计数器配置为中央对齐模式或编码器模式时, 该位由硬件自动配置, 只读</p>
3	OPM	RW	<p>单脉冲模式</p> <p>0: 关闭单脉冲模式</p> <p>1: 发生更新事件时, 清除 CEN 位, 停止计数</p>
2	URS	RW	<p>更新请求源</p> <p>0: 计数器上溢或下溢、设置 UG 位、通过从模式控制器产生的更新均可产生更新中断或 DMA 请求</p> <p>1: 只有计数器上溢或下溢可以产生更新中断或者 DMA 请求</p>
1	UDIS	RW	<p>禁止更新</p> <p>0: 计数器上溢或下溢、设置 UG 位、通过从模式控制器产生的更新可产生更新事件。产生更新事件后, 开启预装载的寄存器被加载为预装载值。</p> <p>1: 禁止更新事件产生。此时开启预装载的寄存器将保持其内容无法被更改, 但通过设置 UG 位或从模式控制器产生了</p>

			硬件复位，计数器和预分频器将被重新初始化。
0	CEN	RW	计数使能 0:停止计数 1:使能计数

19.3.3 GTIM_CR2

中文名： GTIM 控制寄存器 2

寄存器位宽： [31: 0]

偏移量： 0x04

复位值： 0x00000000

表 19- 4 控制寄存器 2

位域	位域名称	访问	描述
31:8	Reserved	—	保留
7	TIIS	RW	TI1 输入选择 0:CH1 引脚连接到 TI1 输入 1:CH1、CH2、CH3 引脚经异或后连到 TI1 输入
6: 4	MMS[2:0]	RW	主模式选择 该配置位用于选择在主模式下向从外设发送的触发输出来源。 000:复位- 使用配置 UG 位和从模式控制器产生的硬件复位作为触发输出。 001:使能- 使用计数器使能信号作为触发输出。计数器实际使能信号由 CEN 控制位与门控模式下触发输入共同决定。 010:更新-更新事件作为触发输出。 011:比较脉冲-当发生一次捕获或比较成功时，当要设置 CC1IF 标志时（即使已经为高），触发输出一个脉冲。 100:OC1REF 信号被作为触发输出。 101:OC2REF 信号被作为触发输出。 110:OC3REF 信号被作为触发输出。 111:OC4REF 信号被作为触发输出。
3	CCDS	RW	捕获/比较的 DMA 选择 0:当发生比较/捕获事件时，发送 CCx 的 DMA 请求 1:当发生更新事件时，发送 CCx 的 DMA 请求
2:0	Reserved	—	保留

19.3.4 GTIM_SMCR

中文名： GTIM 从模式控制寄存器

寄存器位宽： [31:0]

偏移量： 0x08

复位值： 0x00000000

表 19- 5 从模式控制寄存器

位域	位域名称	访问	描述
31:16	Reserved	–	保留
15	ETP	RW	外部触发极性 0:ETR 不反相 1:ETR 反相
14	ECE	RW	外部时钟使能位 0:禁止外部时钟模式 2 1:使能外部使能模式 2 注：复位模式、门控模式和触发模式可以与外部时钟模式 2 同时使用，但是此时 TRGI 不能连接到 ETR。当外部时钟模式 1 和外部时钟模式 2 同时被使能时，外部时钟的输入是 ETR。
13:12	Reserved	–	保留
11:8	ETF[3:0]	RW	外部触发滤波 配置对 ETR 信号采样的频率和数字滤波的带宽。数字滤波器是一个事件计数器，记录到 N 个事件后产生输出跳变。 0000:无滤波器，以 f_{DTS} 采样 0001:采样频率 $f_{SAMP}=f_{CK_INT}$, N=2 0010:采样频率 $f_{SAMP}=f_{CK_INT}$, N=4 0011:采样频率 $f_{SAMP}=f_{CK_INT}$, N=8 0100:采样频率 $f_{SAMP}=f_{DTS}/2$, N=6 0101:采样频率 $f_{SAMP}=f_{DTS}/2$, N=8 0110:采样频率 $f_{SAMP}=f_{DTS}/4$, N=6 0111:采样频率 $f_{SAMP}=f_{DTS}/4$, N=8 1000:采样频率 $f_{SAMP}=f_{DTS}/8$, N=6 1001:采样频率 $f_{SAMP}=f_{DTS}/8$, N=8 1010:采样频率 $f_{SAMP}=f_{DTS}/16$, N=5 1011:采样频率 $f_{SAMP}=f_{DTS}/16$, N=6 1100:采样频率 $f_{SAMP}=f_{DTS}/16$, N=8 1101:采样频率 $f_{SAMP}=f_{DTS}/32$, N=5 1110:采样频率 $f_{SAMP}=f_{DTS}/32$, N=6 1111:采样频率 $f_{SAMP}=f_{DTS}/32$, N=8
7	TSYN	RW	触发同步 0:关闭主从模式 1:触发输入上的事件被延迟触发，从而让当前定时器与被它的触发输出所控制的从定时器间达成同步
6:4	TS[2:0]	RW	触发来源选择 000:内部触发 0, GTIM 001-011:保留 100:TI1 的边沿检测器 101:滤波后的 CH1 输入 110:滤波后的 CH2 输入 111:外部触发输入 ETR

位域	位域名称	访问	描述
3	Reserved	—	保留
2:0	SMS[2:0]	RW	<p>从模式选择</p> <p>000: 关闭从模式</p> <p>001: 编码器模式 1-根据 TI1 的电平, 计数器在 TI2 的边沿向上/向下计数</p> <p>010: 编码器模式 2-根据 TI2 的电平, 计数器在 TI1 的边沿向上/向下计数</p> <p>011: 编码器模式 3-根据另一个信号的电平, 计数器在 TI1 和 TI2 的边沿向上/向下计数</p> <p>100: 复位模式-在选中的触发输入 (TRGI) 的上升沿初始化计数器, 并产生一个更新寄存器的信号</p> <p>101: 门控模式-当触发输入为高时, 计数器计数; 当触发输入为低时, 计数器停止计数但不复位。仅 CEN 位为 1 时门控模式有效</p> <p>110: 触发模式-在触发输入的上升沿, CEN 位被硬件设置为 1</p> <p>111: 外部时钟模式 1-在选中的触发输入的上升沿, 计数器计数</p>

19.3.5 GTIM_DIER

中文名: GTIM DMA/中断使能寄存器

寄存器位宽: [31:0]

偏移量: 0x0c

复位值: 0x00000000

表 19- 6 DMA/中断使能寄存器

位域	位域名称	访问	描述
31:15	Reserved	—	保留
14	TDE	RW	<p>触发事件 DMA 请求使能</p> <p>0: 禁止触发事件 DMA 请求</p> <p>1: 使能触发事件 DMA 请求</p>
13	Reserved	—	保留
12	CC4DE	RW	<p>捕获/请求通道 4 的 DMA 请求使能</p> <p>0: 禁止捕获/比较通道 4 的 DMA 请求</p> <p>1: 使能捕获/比较通道 4 的 DMA 请求</p>
11	CC3DE	RW	<p>捕获/请求通道 3 的 DMA 请求使能</p> <p>0: 禁止捕获/比较通道 3 的 DMA 请求</p> <p>1: 使能捕获/比较通道 3 的 DMA 请求</p>
10	CC2DE	RW	<p>捕获/请求通道 2 的 DMA 请求使能</p> <p>0: 禁止捕获/比较通道 2 的 DMA 请求</p> <p>1: 使能捕获/比较通道 2 的 DMA 请求</p>

位域	位域名称	访问	描述
9	CC1DE	RW	捕获/请求通道 1 的 DMA 请求使能 0:禁止捕获/比较通道 1 的 DMA 请求 1:使能捕获/比较通道 1 的 DMA 请求
8	UDE	RW	更新事件 DMA 请求使能 0:禁止更新事件 DMA 请求 1:使能更新事件 DMA 请求
7	Reserved	—	保留
6	TIE	RW	触发事件中断使能 0:禁止触发事件中断 1:使能触发事件中断
5	Reserved	—	保留
4	CC4IE	RW	捕获/请求通道 4 的中断使能 0:禁止捕获/比较通道 4 中断 1:使能捕获/比较通道 4 中断
3	CC3IE	RW	捕获/请求通道 3 的中断使能 0:禁止捕获/比较通道 3 中断 1:使能捕获/比较通道 3 中断
2	CC2IE	RW	捕获/请求通道 2 的中断使能 0:禁止捕获/比较通道 2 中断 1:使能捕获/比较通道 2 中断
1	CC1IE	RW	捕获/请求通道 1 的中断使能 0:禁止捕获/比较通道 1 中断 1:使能捕获/比较通道 1 中断
0	UIE	RW	更新事件中断使能 0:禁止更新事件中断 1:使能更新事件中断

19.3.6 GTIM_SR

中文名: GTIM 状态寄存器

寄存器位宽: [31:0]

偏移量: 0x10

复位值: 0x00000000

表 19- 7 状态寄存器

位域	位域名称	访问	描述
31:13	Reserved	—	保留
12	CC4OF	R/W0	捕获/请求通道 4 重复捕获事件标志位 仅当相应通道被配置为输入捕获时有意义 0:无重复捕获事件 1:通道 4 发生重复捕获事件

位域	位域名称	访问	描述
11	CC30F	R/W0	捕获/请求通道 3 重复捕获事件标志位 0:无重复捕获事件 1:通道 3 发生重复捕获事件
10	CC20F	R/W0	捕获/请求通道 2 重复捕获事件标志位 0:无重复捕获事件 1:通道 2 发生重复捕获事件
9	CC10F	R/W0	捕获/请求通道 1 重复捕获事件标志位 0:无重复捕获事件 1:通道 1 发生重复捕获事件
8:7	Reserved	—	保留
6	TIF	R/W0	触发事件标志位 当发生触发事件（当从模式控制器处于除门控模式外的其他模式在触发输入端检测到有效边沿，或门控模式下任一边沿）时硬件置 1，软件写 0 清除。 0:无触发事件发生 1:发生触发事件
5	Reserved	—	保留
4	CC4IF	R/W0	捕获/请求通道 4 事件标志位 0:无捕获/比较事件 1:通道 4 发生捕获/比较通道事件
3	CC3IF	R/W0	捕获/请求通道 3 事件标志位 0:无捕获/比较事件 1:通道 3 发生捕获/比较通道事件
2	CC2IF	R/W0	捕获/请求通道 2 事件标志位 0:无捕获/比较事件 1:通道 2 发生捕获/比较通道事件
1	CC1IF	R/W0	捕获/请求通道 1 事件标志位 0:无捕获/比较事件 1:通道 1 发生捕获/比较通道事件
0	UIF	R/W0	更新事件标志位 硬件在产生更新事件时设置该位，软件写 0 清除 0:未产生更新事件 1:产生了更新事件

19.3.7 GTIM_EGR

中文名： GTIM 事件产生寄存器

寄存器位宽： [31:0]

偏移量： 0x14

复位值： 0x00000000

表 19- 8 事件产生寄存器

位域	位域名称	访问	描述
31:7	Reserved	—	保留
6	TG	W	产生触发事件 该位始终读为 0, 写 1 产生触发事件
5	Reserved	—	保留
4	CC4G	W	产生捕获/比较 4 事件 该位始终读为 0, 写 1 产生捕获/比较事件。 若通道为输出: 设置 CC1IF 为 1, 若开启对应的中断和 DMA, 则产生相应的中断和 DMA 若通道为输入: 将当前计数器 CNT 的值捕获置 CCR1 寄存器, 设置 CC1IF 为 1, 若 CC1IF 已经为 1, 则设置 CC1OF 为 1
3	CC3G	W	产生捕获/比较 3 事件 该位始终读为 0, 写 1 产生捕获/比较事件
2	CC2G	W	产生捕获/比较 2 事件 该位始终读为 0, 写 1 产生捕获/比较事件
1	CC1G	W	产生捕获/比较 1 事件 该位始终读为 0, 写 1 产生捕获/比较事件
0	UG	W	产生更新事件 该位始终读为 0, 写 1 产生更新事件, 初始化计数器并更新带有预装载的寄存器

19.3.8 GTIM_CCMR1 (OUT)

中文名: GTIM 捕获/比较模式寄存器 1 (输出)

寄存器位宽: [31:0]

偏移量: 0x18

复位值: 0x00000000

表 19- 9 捕获/比较模式寄存器 1 (输出)

位域	位域名称	访问	描述
31:16	Reserved	—	保留
15	OC2CE	RW	输出比较 2 清零使能 0: 无影响 1: 一旦检测到 ETR 输入高电平, 清除 OC2REF=0

位域	位域名称	访问	描述
14:12	OC2M[2:0]	RW	<p>输出比较 2 模式</p> <p>000:冻结, OC2REF 保持当前状态不变</p> <p>001:当计数器 CNT 的值与捕获/比较寄存器 CCR2 相同时, 强制 OC2REF 为高电平</p> <p>010:当计数器 CNT 的值与捕获/比较寄存器 CCR2 相同时, 强制 OC2REF 为低电平</p> <p>011:当计数器 CNT 的值与捕获/比较寄存器 CCR2 相同时, 翻转 OC2REF 的电平</p> <p>100:强制 OC2REF 为低电平</p> <p>101:强制 OC2REF 为高电平</p> <p>110:PWM 模式 1-在向上计数时, 一旦 $CNT < CCR2$, OC2REF 为高电平, 否则为低电平;在向下计数时, 一旦 $CNT > CCR2$, OC2REF 为低电平, 否则为高电平。</p> <p>111:PWM 模式 2-在向上计数时, 一旦 $CNT < CCR2$, OC2REF 为低电平, 否则为高电平;在向下计数时, 一旦 $CNT > CCR2$, OC2REF 为高电平, 否则为低电平。</p>
11	OC2PE	RW	<p>输出比较 2 预装载使能</p> <p>0:关闭 CCR2 寄存器的预装载功能</p> <p>1:开启 CCR2 寄存器的预装载功能</p>
10	OC2FE	RW	<p>输出比较 2 快速使能</p> <p>0:无影响</p> <p>1:仅在单脉冲模式下且通道配置为 PWM1/PWM2 模式时生效, 当开始计数时, 立即输出有效电平, OC2REF 电平此时与比较结果无关</p>
9:8	CC2S[1:0]	RW	<p>输出/比较 2 选择</p> <p>00:CC2 通道被配置为输出</p> <p>01:CC2 通道被配置为输入, IC2 映射在 TI2 上</p> <p>10:CC2 通道被配置为输入, IC2 映射在 TI1 上</p> <p>11:CC2 通道被配置为输入, IC2 映射在 TRC 上</p>
7	OC1CE	RW	<p>输出比较 1 清零使能</p> <p>0:无影响</p> <p>1:一旦检测到 ETR 输入高电平, 清除 OC1REF=0</p>
6:4	OC1M[2:0]	RW	<p>输出比较 1 模式</p> <p>参见 OC2M 说明</p>
3	OC1PE	RW	<p>输出比较 1 预装载使能</p> <p>0:关闭 CCR1 寄存器的预装载功能</p> <p>1:开启 CCR1 寄存器的预装载功能</p>
2	OC1FE	RW	<p>输出比较 1 快速使能</p> <p>参见 OC2FE 说明</p>

位域	位域名称	访问	描述
1:0	CC1S[1:0]	RW	输出/比较 1 选择 00:CC1 通道被配置为输出 01:CC1 通道被配置为输入, IC1 映射在 TI1 上 10:CC1 通道被配置为输入, IC1 映射在 TI2 上 11:CC1 通道被配置为输入, IC1 映射在 TRC 上

19.3.9 GTIM_CCMR1 (IN)

中文名: GTIM 捕获/比较模式寄存器 1(输入)

寄存器位宽: [31:0]

偏移量: 0x18

复位值: 0x00000000

表 19- 10 捕获/比较模式寄存器 1(输入)

位域	位域名称	访问	描述
31:16	Reserved	—	保留
15:12	IC2F[3:0]	RW	输入捕获 2 滤波器 配置对 TI2 输入信号采样的频率和数字滤波的带宽。数字滤波器是一个事件计数器, 记录到 N 个事件后产生输出跳变。 0000:无滤波器, 以 f_{DTS} 采样 0001:采样频率 $f_{SAMP}=f_{CK_INT}$, $N=2$ 0010:采样频率 $f_{SAMP}=f_{CK_INT}$, $N=4$ 0011:采样频率 $f_{SAMP}=f_{CK_INT}$, $N=8$ 0100:采样频率 $f_{SAMP}=f_{CK_INT}/2$, $N=6$ 0101:采样频率 $f_{SAMP}=f_{CK_INT}/2$, $N=8$ 0110:采样频率 $f_{SAMP}=f_{CK_INT}/4$, $N=6$ 0111:采样频率 $f_{SAMP}=f_{CK_INT}/4$, $N=8$ 1000:采样频率 $f_{SAMP}=f_{CK_INT}/8$, $N=6$ 1001:采样频率 $f_{SAMP}=f_{CK_INT}/8$, $N=8$ 1010:采样频率 $f_{SAMP}=f_{CK_INT}/16$, $N=5$ 1011:采样频率 $f_{SAMP}=f_{CK_INT}/16$, $N=6$ 1100:采样频率 $f_{SAMP}=f_{CK_INT}/16$, $N=8$ 1101:采样频率 $f_{SAMP}=f_{CK_INT}/32$, $N=5$ 1110:采样频率 $f_{SAMP}=f_{CK_INT}/32$, $N=6$ 1111:采样频率 $f_{SAMP}=f_{CK_INT}/32$, $N=8$
11:10	IC2PSC[1:0]	RW	输入捕获 2 预分频器 定义 CC2 输入 (IC2) 的预分频系数 00:无预分频 01:每 2 个事件触发一次捕获 10:每 4 个事件触发一次捕获 11:每 8 个事件触发一次捕获

位域	位域名称	访问	描述
9:8	CC2S[1:0]	RW	输出/比较 2 选择 00:CC2 通道被配置为输出 01:CC2 通道被配置为输入, IC2 映射在 TI2 上 10:CC2 通道被配置为输入, IC2 映射在 TI1 上 11:CC2 通道被配置为输入, IC2 映射在 TRC 上
7:4	IC1F[3:0]	RW	输入捕获 1 滤波器 参考 IC2F 说明
3:2	IC1PSC[1:0]	RW	输入捕获 1 预分频器 参考 IC2PSC 说明
1:0	CC1S[1:0]	RW	输出/比较 1 选择 00:CC1 通道被配置为输出 01:CC1 通道被配置为输入, IC1 映射在 TI1 上 10:CC1 通道被配置为输入, IC1 映射在 TI2 上 11:CC1 通道被配置为输入, IC1 映射在 TRC 上

19.3.10 GTIM_CCMR2 (OUT)

中文名: GTIM 捕获/比较模式寄存器 2(输出)

寄存器位宽: [31:0]

偏移量: 0x1c

复位值: 0x00000000

表 19- 11 捕获/比较模式寄存器 2(输出)

位域	位域名称	访问	描述
31:16	Reserved	—	保留
15	OC4CE	RW	输出比较 4 清零使能 0:无影响 1:一旦检测到 ETR 输入高电平, 清除 OC4REF=0
14:12	OC4M[2:0]	RW	输出比较 4 模式 000:冻结, OC4REF 保持当前状态不变 001:当计数器 CNT 的值与捕获/比较寄存器 CCR4 相同时, 强制 OC4REF 为高电平 010:当计数器 CNT 的值与捕获/比较寄存器 CCR4 相同时, 强制 OC4REF 为低电平 011:当计数器 CNT 的值与捕获/比较寄存器 CCR4 相同时, 翻转 OC4REF 的电平 100:强制 OC4REF 为低电平 101:强制 OC4REF 为高电平 110:PWM 模式 1-在向上计数时, 一旦 CNT<CCR4, OC4REF 为高电平, 否则为低电平;在向下计数时, 一旦 CNT>CCR4, OC4REF 为低电平, 否则为高电平。 111:PWM 模式 2-在向上计数时, 一旦 CNT<CCR4, OC4REF 为低电平, 否则为高电平;在向下计数时, 一旦 CNT>CCR4, OC4REF 为高电平, 否则为低电平。

位域	位域名称	访问	描述
11	OC4PE	RW	输出比较 4 预装载使能 0:关闭 CCR4 寄存器的预装载功能 1:开启 CCR4 寄存器的预装载功能
10	OC4FE	RW	输出比较 4 快速使能 0:无影响 1:仅在单脉冲模式下且通道配置为 PWM1/PWM2 模式时生效, 当开始计数时, 立即输出有效电平, OC4REF 电平此时与比较结果无关
9:8	CC4S[1:0]	RW	输出/比较 4 选择 00:CC4 通道被配置为输出 01:CC4 通道被配置为输入, IC4 映射在 TI4 上 10:CC4 通道被配置为输入, IC4 映射在 TI3 上 11:CC4 通道被配置为输入, IC4 映射在 TRC 上
7	OC3CE	RW	输出比较 3 清零使能 0:无影响 1:一旦检测到 ETR 输入高电平, 清除 OC3REF=0
6:4	OC3M[2:0]	RW	输出比较 3 模式 参见 OC4M 说明
3	OC3PE	RW	输出比较 3 预装载使能 0:关闭 CCR3 寄存器的预装载功能 1:开启 CCR3 寄存器的预装载功能
2	OC3FE	RW	输出比较 3 快速使能 参见 OC4FE 说明
1:0	CC3S[1:0]	RW	输出/比较 3 选择 00:CC3 通道被配置为输出 01:CC3 通道被配置为输入, IC3 映射在 TI3 上 10:CC3 通道被配置为输入, IC3 映射在 TI4 上 11:CC3 通道被配置为输入, IC3 映射在 TRC 上

19.3.11 GTIM_CCMR2(IN)

中文名: GTIM 捕获/比较模式寄存器 2(输入)

寄存器位宽: [31:0]

偏移量: 0x1c

复位值: 0x00000000

表 19- 12 捕获/比较模式寄存器 2(输入)

位域	位域名称	访问	描述
31:16	Reserved	—	保留

位域	位域名称	访问	描述
15:12	IC4F[3:0]	RW	<p>输入捕获 4 滤波器</p> <p>配置对 TI4 输入信号采样的频率和数字滤波的带宽。数字滤波器是一个事件计数器，记录到 N 个事件后产生输出跳变。</p> <p>0000:无滤波器，以 f_{DTS} 采样</p> <p>0001:采样频率 $f_{SAMP}=f_{CK_INT}$，N=2</p> <p>0010:采样频率 $f_{SAMP}=f_{CK_INT}$，N=4</p> <p>0011:采样频率 $f_{SAMP}=f_{CK_INT}$，N=8</p> <p>0100:采样频率 $f_{SAMP}=f_{CK_INT}/2$，N=6</p> <p>0101:采样频率 $f_{SAMP}=f_{CK_INT}/2$，N=8</p> <p>0110:采样频率 $f_{SAMP}=f_{CK_INT}/4$，N=6</p> <p>0111:采样频率 $f_{SAMP}=f_{CK_INT}/4$，N=8</p> <p>1000:采样频率 $f_{SAMP}=f_{CK_INT}/8$，N=6</p> <p>1001:采样频率 $f_{SAMP}=f_{CK_INT}/8$，N=8</p> <p>1010:采样频率 $f_{SAMP}=f_{CK_INT}/16$，N=5</p> <p>1011:采样频率 $f_{SAMP}=f_{CK_INT}/16$，N=6</p> <p>1100:采样频率 $f_{SAMP}=f_{CK_INT}/16$，N=8</p> <p>1101:采样频率 $f_{SAMP}=f_{CK_INT}/32$，N=5</p> <p>1110:采样频率 $f_{SAMP}=f_{CK_INT}/32$，N=6</p> <p>1111:采样频率 $f_{SAMP}=f_{CK_INT}/32$，N=8</p>
11:10	IC4PSC[1:0]	RW	<p>输入捕获 4 预分频器</p> <p>定义 CC4 输入（IC4）的预分频系数</p> <p>00:无预分频</p> <p>01:每 2 个事件触发一次捕获</p> <p>10:每 4 个事件触发一次捕获</p> <p>11:每 8 个事件触发一次捕获</p>
9:8	CC4S[1:0]	RW	<p>输出/比较 4 选择</p> <p>00:CC4 通道被配置为输出</p> <p>01:CC4 通道被配置为输入，IC4 映射在 TI4 上</p> <p>10:CC4 通道被配置为输入，IC4 映射在 TI3 上</p> <p>11:CC4 通道被配置为输入，IC4 映射在 TRC 上</p>
7:4	IC3F[3:0]	RW	<p>输入捕获 3 滤波器</p> <p>参考 IC4F 说明</p>
3:2	IC3PSC[1:0]	RW	<p>输入捕获 3 预分频器</p> <p>参考 IC4PSC 说明</p>
1:0	CC3S[1:0]	RW	<p>输出/比较 3 选择</p> <p>00:CC3 通道被配置为输出</p> <p>01:CC3 通道被配置为输入，IC3 映射在 TI3 上</p> <p>10:CC3 通道被配置为输入，IC3 映射在 TI4 上</p> <p>11:CC3 通道被配置为输入，IC3 映射在 TRC 上</p>

19.3.12 GTIM_CCER

中文名: GTIM 捕获/比较使能寄存器

寄存器位宽: [31:0]

偏移量: 0x20

复位值: 0x00000000

表 19- 13 捕获/比较使能寄存器

位域	位域名称	访问	描述
31:14	Reserved	—	保留
13	CC4P	RW	输入/捕获 4 极性 CC4 通道配置为输出: 0:OC4=OC4REF 1:OC4 为 OC4REF 反相输出 CC4 通道配置为输入: 0:不反相:捕获发生在 IC4 的上升沿 1:反相:捕获发生在 IC4 的下降沿
12	CC4E	RW	输入/捕获 4 使能 CC4 通道配置为输出: 0:OC4 禁止输出 1:OC4 信号输出到对应的引脚 CC4 通道配置为输入: 0:捕获禁止 1:捕获使能
11:10	Reserved	—	保留
9	CC3P	RW	输入/捕获 3 极性 参考 CC4P 描述
8	CC3E	RW	输入/捕获 3 使能 参考 CC4E 描述
7:6	Reserved	—	保留
5	CC2P	RW	输入/捕获 2 极性 参考 CC4P 描述
4	CC2E	RW	输入/捕获 2 使能 参考 CC4E 描述
3:2	Reserved	—	保留
1	CC1P	RW	输入/捕获 1 极性 参考 CC4P 描述
0	CC1E	RW	输入/捕获 1 使能 参考 CC4E 描述

19.3.13 GTIM_CNT

中文名: GTIM 计数器

寄存器位宽: [31:0]

偏移量: 0x24

复位值: 0x00000000

表 19- 14 计数器

位域	位域名称	访问	描述
31:0	CNT	RW	计数器数值

19.3.14 GTIM_PSC

中文名: GTIM 预分频器

寄存器位宽: [31:0]

偏移量: 0x28

复位值: 0x00000000

表 19- 15 预分频器

位域	位域名称	访问	描述
31:0	PSC	RW	预分频器数值 计数器的实际时钟频率为 $f_{CK_INT}/(psc+1)$ 预分频器始终开启预装载功能, 在发生更新事件时, psc 的数值被装载到实际的预分频寄存器中。

19.3.15 GTIM_ARR

中文名: GTIM 自动重装载寄存器

寄存器位宽: [31:0]

偏移量: 0x2c

复位值: 0x00000000

表 19- 16 自动重装载寄存器

位域	位域名称	访问	描述
31:0	ARR	RW	自动重装载数值 当计数器 cnt 与 arr 相等时, 产生上溢事件, 并根据相关配置位产生更新事件。 注: 当 arr 为 0 时, 计数器停止。

19.3.16 GTIM_CCR1

中文名: GTIM 捕获/比较寄存器 1

寄存器位宽: [31:0]

偏移量: 0x34

复位值: 0x00000000

表 19- 17 捕获/比较寄存器 1

位域	位域名称	访问	描述
----	------	----	----

位域	位域名称	访问	描述
31:0	CCR1	RW	捕获/比较 1 数值 CC1 通道配置为输出： CCR1 包含了装入当前捕获/比较 1 寄存器的值（预装载值） CC1 通道配置为输入： CCR1 包含了上次输入捕获 1 事件时传输的计数器 CNT 值

19.3.17 GTIM_CCR2

中文名： GTIM 捕获/比较寄存器 2

寄存器位宽： [31:0]

偏移量： 0x38

复位值： 0x00000000

表 19- 18 捕获/比较寄存器 2

位域	位域名称	访问	描述
31:0	CCR2	RW	捕获/比较 2 数值 CC2 通道配置为输出： CCR2 包含了装入当前捕获/比较 2 寄存器的值（预装载值） CC2 通道配置为输入： CCR2 包含了上次输入捕获 2 事件时传输的计数器 CNT 值

19.3.18 GTIM_CCR3

中文名： GTIM 捕获/比较寄存器 3

寄存器位宽： [31:0]

偏移量： 0x3c

复位值： 0x00000000

表 19- 19 捕获/比较寄存器 3

位域	位域名称	访问	描述
31:0	CCR3	RW	捕获/比较 3 数值 CC3 通道配置为输出： CCR3 包含了装入当前捕获/比较 3 寄存器的值（预装载值） CC3 通道配置为输入： CCR3 包含了上次输入捕获 3 事件时传输的计数器 CNT 值

19.3.19 GTIM_CCR4

中文名： GTIM 捕获/比较寄存器 4

寄存器位宽： [31:0]

偏移量： 0x40

复位值： 0x00000000

表 19- 20 捕获/比较寄存器 4

位域	位域名称	访问	描述
31:0	CCR4	RW	捕获/比较 4 数值 CCR4 通道配置为输出： CCR4 包含了装入当前捕获/比较 4 寄存器的值（预装载值） CCR4 通道配置为输入： CCR4 包含了上次输入捕获 4 事件时传输的计数器 CNT 值

19.3.20 GTIM_INSTA

中文名： GTIM 输入通道状态寄存器

寄存器位宽： [31:0]

偏移量： 0x50

复位值： 0x00000000

表 19- 21 输入通道状态寄存器

位域	位域名称	访问	描述
31:5	Reserved	—	保留
4	ETR_IN	R	ETR 输入
3	CH4_IN	R	CH4 输入
2	CH3_IN	R	CH3 输入
1	CH2_IN	R	CH2 输入
0	CH1_IN	R	CH1 输入

20 BTIM 控制器

20.1 概述

龙芯 2K0300 集成了一个与 HPET 工作方式类似的具有 32 位自动装载寄存器的定时器，由可编程预分频器驱动，可以用于提供时间基准。

定时器主要包括 CNT、PSC、ARR 三个寄存器，当时钟计数器 CNT 的计数值与 ARR 相同时触发中断。。

BTIM 的时钟由内部 APB 总线时钟提供，基地址为 0x1611a000，寄存器定义见下文。

20.2 功能描述

计数器从 0 开始累加计数到 ARR 寄存器中存储的自动重装载数值，然后重新从 0 开始计数并产生一个计数器溢出事件。每次计数器溢出、设置 UG 位、从模式控制器产生硬件复位时可根据相关配置位配置产生更新事件。

用户可通过配置 PSC 寄存器对计数器时钟进行预分频，配置 ARR 寄存器用于配置计数周期，且可对 CNT 计数器实时进行读取与修改，实际完成一次计数周期的时长由 PSC 与 ARR 共同决定。

PSC 与 ARR 寄存器具有预装载功能，即当次的配置值会留到下个计数周期/更新事件产生时才生效，其中 PSC 寄存器预装载功能始终打开，ARR 寄存器预装载功能可自由选择开启或关闭。

当发生更新事件时，PSC 内部计数被清零，CNT 根据计数模式进行重装载，同时：

- SR 寄存器中的 UIF 标志位被设置。
- PSC 寄存器中的值被装载到实际的内部寄存器。
- 若设置了 ARPE 位，ARR 寄存器中的值被装载到实际的自动重装载寄存器。

20.3 寄存器描述

20.3.1 寄存器地址列表

表 20- 1 GTIM 寄存器列表

名称	偏移地址	位宽	描述
BTIM_CR1	0x00	32	控制寄存器 1
BTIM_CR2	0x04	32	控制寄存器 2
BTIM_DIER	0x0C	32	DMA/中断使能寄存器
BTIM_SR	0x10	32	状态寄存器

BTIM_EGR	0x14	32	事件产生寄存器
BTIM_CNT	0x24	32	计数器
BTIM_PSC	0x28	32	预分频器
BTIM_ARR	0x2C	32	自动重装载寄存器

20.3.2 BTIM_CR1

中文名： BTIM 控制寄存器 1

寄存器位宽： [31:0]

偏移量： 0x00

复位值： 0x00000000

表 20- 2 控制寄存器 1

位域	位域名称	访问	描述
31:8	Reserved	—	保留
7	ARPE	RW	自动重装载预装载使能 0:ARR 寄存器没有预装载 1:ARR 寄存器开启预装载
6:4	Reserved	—	保留
3	OPM	RW	单脉冲模式 0: 关闭单脉冲模式 1:发生更新事件时，清除 CEN 位，停止计数
2	URS	RW	更新请求源 0: 计数器上溢或下溢、设置 UG 位、通过从模式控制器产生的更新均可产生更新中断或 DMA 请求 1:只有计数器上溢或下溢可以产生更新中断或者 DMA 请求
1	UDIS	RW	禁止更新 0:计数器上溢或下溢、设置 UG 位、通过从模式控制器产生的更新可产生更新事件。产生更新事件后，开启预装载的寄存器被加载为预装载值。 1:禁止更新事件产生。此时开启预装载的寄存器将保持其内容无法被更改，但通过设置 UG 位或从模式控制器产生了硬件复位，计数器和预分频器将被重新初始化。
0	CEN	RW	计数使能 0:停止计数 1:使能计数

20.3.3 BTIM_CR2

中文名： BTIM 控制寄存器 2

寄存器位宽: [31: 0]

偏移量: 0x04

复位值: 0x00000000

表 20- 3 控制寄存器 2

位域	位域名称	访问	描述
31:7	Reserved	–	保留
6: 4	MMS[2:0]	RW	主模式选择 该配置位用于选择在主模式下向从外设发送的触发输出来源。 000:复位- 使用配置 UG 位和从模式控制器产生的硬件复位作为触发输出。 001:使能- 使用计数器使能信号作为触发输出。计数器实际使能信号由 CEN 控制位与门控模式下触发输入共同决定。 010:更新-更新事件作为触发输出。
3:0	Reserved	–	保留

20.3.4 BTIM_DIER

中文名: BTIM DMA/中断使能寄存器

寄存器位宽: [31:0]

偏移量: 0x0c

复位值: 0x00000000

表 20- 4 DMA/中断使能寄存器

位域	位域名称	访问	描述
31:9	Reserved	–	保留
8	UDE	RW	更新事件 DMA 请求使能 0:禁止更新事件 DMA 请求 1:使能更新事件 DMA 请求
7:1	Reserved	–	保留
0	UIE	RW	更新事件中断使能 0:禁止更新事件中断 1:使能更新事件中断

20.3.5 BTIM_SR

中文名: BTIM 状态寄存器

寄存器位宽: [31:0]

偏移量: 0x10

复位值: 0x00000000

表 20- 5 状态寄存器

位域	位域名称	访问	描述
31:1	Reserved	—	保留
0	UIF	R/W0	更新事件标志位 硬件在产生更新事件时设置该位，软件写 0 清除 0: 未产生更新事件 1: 产生了更新事件

20.3.6 BTIM_EGR

中文名： BTIM 事件产生寄存器

寄存器位宽： [31:0]

偏移量： 0x14

复位值： 0x00000000

表 20- 6 事件产生寄存器

位域	位域名称	访问	描述
31:1	Reserved	—	保留
0	UG	W	产生更新事件 该位始终读为 0，写 1 产生更新事件，初始化计数器并更新带有预装载的寄存器

20.3.7 BTIM_CNT

中文名： BTIM 计数器

寄存器位宽： [31:0]

偏移量： 0x24

复位值： 0x00000000

表 20- 7 计数器

位域	位域名称	访问	描述
31:0	CNT	RW	计数器数值

20.3.8 BTIM_PSC

中文名： BTIM 预分频器

寄存器位宽： [31:0]

偏移量： 0x28

复位值： 0x00000000

表 20- 8 预分频器

位域	位域名称	访问	描述
----	------	----	----

位域	位域名称	访问	描述
31:0	PSC	RW	<p>预分频器数值</p> <p>计数器的实际时钟频率为 $f_{CK_INT}/(psc+1)$</p> <p>预分频器始终开启预装载功能，在发生更新事件时，psc 的数值被装载到实际的预分频寄存器中。</p>

20.3.9 BTIM_ARR

中文名： BTIM 自动重装载寄存器

寄存器位宽： [31:0]

偏移量： 0x2c

复位值： 0x00000000

表 20- 9 自动重装载寄存器

位域	位域名称	访问	描述
31:0	ARR	RW	<p>自动重装载数值</p> <p>当计数器 cnt 与 arr 相等时，产生上溢事件，并根据相关配置位产生更新事件。</p> <p>注：当 arr 为 0 时，计数器停止。</p>

21 PWM 控制器

21.1 概述

龙芯 2K0300 芯片实现了 4 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，计数寄存器和参考寄存器均 32 位数据宽度。

21.2 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 21- 1 PWM 寄存器地址

地址	设备	备注
0x1611_b000	PWM0~3	每个 PWM 占用 16B 寄存器配置空间： 0x1611_b000-pwm0, 0x1611_b010-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。

21.3 寄存器描述

每路控制器共有五个寄存器，具体描述如下：

表 21- 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 21- 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

位域	名称	访问	复位值	说明
4	SINGLE	R/W	0	单脉冲控制位 置 1 时：脉冲仅产生一次 置 0 时：脉冲持续产生
5	INTE	R/W	0	中断使能位 置 1 时：当 full_pulse 到 1 时送中断 置 0 时：不产生中断
6	INT	R/W	0	中断位 读操作：1 表示有中断产生, 0 表示没有中断 写入 1：清中断
7	RST	R/W	0	使得 Low_level 和 full_pulse 计数器重置 置 1 时：计数器重置（从 buffer 读，输出低电平） 置 0 时：计数器正常工作
8	CAPTE	R/W	0	测量脉冲使能 置 1 时：测量脉冲模式 置 0 时：非测量脉冲模式（一般而言则是脉冲输出模式）
9	INVERT	R/W	0	输出翻转使能 置 1 时：使脉冲在输出去发生信号翻转（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

21.4 功能说明

21.4.1 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,在写入新数之后再将 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。

21.4.2 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。

21.4.3 防死区功能

四路 PWM 都配备了防死区功能,可以防止四路脉冲输出同时发生跳变。

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

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

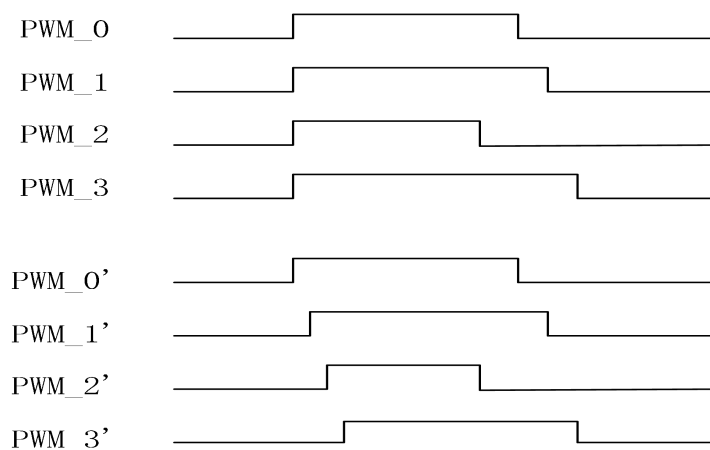


图21- 1 防死区功能

22 ADC 控制器

22.1 概述

龙芯 2K0300 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 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 JEXTSEL2:0]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 0x1611_c000, 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
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

位域	位域名称	访问	描述
			1: 启用差分模式
19:16	Reserved		保留
15:13	DISCNUM	RW	<p>中断模式通道计数</p> <p>在中断模式下，收到外部触发后转换规则通道的数目</p> <p>000: 1 个通道</p> <p>001: 2 个通道</p> <p>.....</p> <p>111: 8 个通道</p>
12	JDISCEN	RW	<p>在注入通道上的中断模式</p> <p>用于开启或关闭注入通道组上的中断模式</p> <p>0: 注入通道停用中断模式</p> <p>1: 注入通道启用中断模式</p>
11	DISCEN	RW	<p>在规则通道上的中断模式</p> <p>用于开启或关闭规则通道组上的中断模式</p> <p>0: 规则通道停用中断模式</p> <p>1: 规则通道启用中断模式</p>
10	JAUTO	RW	<p>自动注入通道组开转换</p> <p>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</p> <p>0: 关闭自动的注入通道组转换</p> <p>1: 开启自动的注入通道组转换</p>
9	AWDSGL	RW	<p>扫描模式中在一个单一的通道上使用看门狗</p> <p>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</p> <p>0: 在所有的通道上使用模拟看门狗</p> <p>1: 在单一通道上使用模拟看门狗</p>
8	SCAN	RW	<p>扫描模式</p> <p>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。</p> <p>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</p> <p>0: 关闭扫描模式</p> <p>1: 使用扫描模式</p>
7	JEOCIE	RW	<p>注入通道中断使能</p> <p>用于禁止或允许所有注入通道转换结束后产生中断。</p> <p>0: 禁止 JEOC 中断</p> <p>1: 允许 JEOC 中断。当硬件设置 JEOC 位时产生中断</p>
6	AWDIE	RW	<p>模拟看门狗中断使能</p> <p>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。</p> <p>0: 禁止模拟看门狗中断</p>

位域	位域名称	访问	描述
			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	ADCEDGE	RW	ADC 时钟触发沿选择 0: 上升沿触发 1: 下降沿触发
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

位域	位域名称	访问	描述
21	JSWSTART	RW	<p>开始转换注入通道</p> <p>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</p> <p>0: 复位状态</p> <p>1: 开始转换注入通道</p>
20	EXTTRIG	RW	<p>规则通道的外部触发转换模式</p> <p>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</p> <p>0: 不用外部事件启动转换</p> <p>1: 使用外部事件启动转换</p>
19:17	EXTSEL	RW	<p>选择启动规则通道组转换的外部事件</p> <p>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</p> <p>3'b000: ATIM_CC1 事件</p> <p>3'b001: ATIM_CC2 事件</p> <p>3'b010: ATIM_CC3 事件</p> <p>3'b011: GTIM_CC2 事件</p> <p>3'b110: EXTI 线 11</p> <p>3'b111: swstart</p>
16	Reserved		保留
15	JEXTTRIG	RW	<p>注入通道的外部触发转换模式</p> <p>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</p> <p>0: 不用外部事件启动转换</p> <p>1: 使用外部事件启动转换</p>
14:12	JEXTSEL	RW	<p>选择启动注入通道组转换的外部事件</p> <p>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</p> <p>3'b000: ATIM_TRGO 事件</p> <p>3'b001: ATIM_CC4 事件</p> <p>3'b010: GTIM_TRGO 事件</p> <p>3'b011: GTIM_CC1 事件</p> <p>3'b110: EXTI 线 15</p> <p>3'b111: JSWSTART</p>
11	ALIGN	RW	<p>数据对齐</p> <p>0: 右对齐</p> <p>1: 左对齐</p>
10:9	Reserved		保留

位域	位域名称	访问	描述
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
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

位域	位域名称	访问	描述
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_JOFRx (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

位域	位域名称	访问	描述
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 DMA 控制器

23.1 概述

龙芯 2K0300 的 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。

23.2 寄存器定义

DMA 控制器寄存器的基地址为 0x1612_c000。

部分表格内以 x 代替通道编号 0-7。

表 23- 1 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
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		

偏移	名称	描述
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

23.2.1 DMA 中断状态寄存器 (DMA_ISR)

偏移量: 0x00

复位值: 0x00000000

表 23- 2 DMA 中断状态寄存器

位域	名称	访问	描述
$4*x + 3$	TEIFx	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。
$4*x + 2$	HTIFx	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。

$4 \times x + 1$	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。
$4 \times x$	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。

23.2.2 DMA 中断标志清除寄存器 (DMA_IFCR)

偏移量: 0x04

复位值: 0x00000000

表 23- 3 DMA 中断标志清除寄存器

位域	名称	访问	描述
$4 \times x + 3$	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。
$4 \times x + 2$	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。
$4 \times x + 1$	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。
$4 \times x$	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。

23.2.3 DMA 通道 x 配置寄存器 (DMA_CCR_x)

偏移量: $0x08 + 0x14 \times x$

复位值: 0x00000000

表 23- 4 DMA 通道 x 配置寄存器

位域	名称	访问	描述
31:15	Reserved	—	—
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。

23.2.4 DMA 通道 x 传输数量寄存器 (DMA_CNDTRx)

偏移量: $0x0c + 0x14 * x$

复位值: 0x00000000

表 23- 5 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。

23.2.5 DMA 通道 x 外设地址寄存器 (DMA_CPARx)

偏移量: $0x10 + 0x14 * x$

复位值: 0x00000000

表 23- 6 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。
------	-------	----	----------------------------------

23.2.6 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 23- 7 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。

23.3 功能描述

23.3.1 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；

23.3.2 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。

23.3.3 通道映射

外设各请求类型与各通道的映射可在芯片通用配置寄存器 13、14、15 中配置（TIMER 除外），可选的通道映射见下表，其中“*”符号代表默认映射配置。ATIM/GTIM/BTIM 为固定映射，无法修改。

表 23- 8 外设请求的通道映射

外设 DMA 通道	CH0	CH1	CH2	CH3	CH4	CH5	CH6	CH7
--------------	-----	-----	-----	-----	-----	-----	-----	-----

UART0	RX*	TX*	RX	TX	RX	TX	RX	TX
UART1	RX	TX	RX*	TX*	RX	TX	RX	TX
UART2	RX	TX	RX	TX	RX*	TX*	RX	TX
UART3	RX	TX	RX	TX	RX	TX	RX*	TX*
UART4	RX*	TX*	RX	TX	RX	TX	RX	TX
UART5	RX	TX	RX*	TX*	RX	TX	RX	TX
UART6	RX	TX	RX	TX	RX*	TX*	RX	TX
UART7	RX	TX	RX	TX	RX	TX	RX*	TX*
UART8	RX*	TX*	RX	TX	RX	TX	RX	TX
UART9	RX	TX	RX*	TX*	RX	TX	RX	TX
I2C0	RX*	TX*	RX	TX	RX	TX	RX	TX
I2C1	RX	TX	RX*	TX*	RX	TX	RX	TX
I2C2	RX	TX	RX	TX	RX*	TX*	RX	TX
I2C3	RX	TX	RX	TX	RX	TX	RX*	TX*
SPI2	RX*	TX*	RX	TX	RX	TX	RX	TX
SPI3	RX	TX	RX*	TX*	RX	TX	RX	TX
I2S	RX*	TX*	RX	TX	RX	TX	RX	TX
ADC	RX*	RX	RX	RX	RX	RX	RX	RX
CAN0	RX*	RX	RX	RX	RX	RX	RX	RX
CAN1	RX	RX*	RX	RX	RX	RX	RX	RX
CAN2	RX	RX	RX*	RX	RX	RX	RX	RX
CAN3	RX	RX	RX	RX*	RX	RX	RX	RX
ATIM	CH1	CH2	CH3	CH4	COM	UP	TRG	—
GTIM	CH1	CH2	CH3	CH4	—	UP	TRG	—
BTIM	—	—	—	—	—	UP	—	—

24 HPET 控制器

24.1 概述

龙芯 2K0300 芯片实现了 4 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 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。

24.2 访问地址

HPET 控制器内部寄存器的物理地址构成如下：

表 24- 1 HPET 寄存器地址

地址	设备	备注
0x1612_0000	HPET0	4KB 寄存器配置空间
0x1612_1000	HPET1	4KB 寄存器配置空间
0x1612_2000	HPET2	4KB 寄存器配置空间
0x1612_3000	HPET3	4KB 寄存器配置空间

24.3 寄存器描述

下表列出了 HPET 的寄存器：

表 24- 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

寄存器偏移地址	寄存器	类型
028-0EFh	Reserved	
0F0-0F7h	Main Counter Value Register	R/W
100-107h	Timer 0 Configuration and Capability Register	R/W
108-10Fh	Timer 0 Comparator Value Register	R/W
110-11Fh	Reserved	
120-127h	Timer 1 Configuration and Capability Register	R/W
128-12Fh	Timer 1 Comparator Value Register	R/W
130-13Fh	Reserved	
140-147h	Timer 2 Configuration and Capability Register	R/W
148-14Fh	Timer 2 Comparator Value Register	R/W
150-15Fh	Reserved	

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

24.3.1 General Capabilities and ID Register

表 24- 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

24.3.2 General Configuration Register

表 24- 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	R/W

		中断： 1: 主计时器计时且允许定时器产生中断；	
--	--	-----------------------------	--

24.3.3 General Interrupt Status Register

表 24- 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 Capability 寄存器的 Tn_TYPE_CNF 位确定。

24.3.4 Main Counter Value Register

表 24- 6 Main Counter Value 寄存器

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

24.3.5 Timer N Configuration and Capabilities Register

表 24- 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
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	R/W

位	名称	描述	读写特性
		Timer1, Timer2, 这位默认为 0, 且为只读。	
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		

24.3.6 Timer N Comparator Value Register

表 24- 8 Timer N Comparator Value 寄存器

位	名称	描述	读写特性
63: 32	Reserved		
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,	R/W

位	名称	描述	读写特性
		<p>那么当主计时器的值为 0x123h 时，产生中断； 比较器的值被硬件修改为 0x246h； 当主计时器的值达到 0x246h 时，产生另外一个中断； 比较器的值被硬件修改为 0x369h。</p> <p>◆ 只要产生中断，那么比较器的值都会累加；直到比较器的值达到最大 (0xffffffff)，那么累加器的值将会继续累加。比如当比较器的值是 FFFF0000h，而最后一次由软件写入比较器的值是 20000。当中断发生后，比较器的值变为 00010000h。</p>	

25 RTC

25.1 概述

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

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

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

25.2 寄存器描述

RTC 模块寄存器位于 0x16128000——0x1612bfff 的 16KB 地址空间内，其基地址为 0x16128000，所有寄存器位宽均为 32 位。

25.2.1 寄存器地址列表

表 25- 1 RTC 寄存器列表

名称	偏移地址	位宽	RW	描述
sys_toytrim	0x20	32	RW	软件必须初始化为 0
sys_toywrite0	0x24	32	W	TOY 低 32 位数值写入
sys_toywritel	0x28	32	W	TOY 高 32 位数值写入
sys_toyread0	0x2C	32	R	TOY 低 32 位数值读出
sys_toyreadl	0x30	32	R	TOY 高 32 位数值读出
sys_toymatch0	0x34	32	RW	TOY 定时中断 0
sys_toymatch1	0x38	32	RW	TOY 定时中断 1
sys_toymatch2	0x3C	32	RW	TOY 定时中断 2
sys_rtcctrl	0x40	32	RW	TOY 和 RTC 控制寄存器 软件必须初始化
sys_rtctrim	0x60	32	RW	软件必须初始化为 0
sys_rtcwrite0	0x64	32	W	RTC 定时计数写入
sys_rtcread0	0x68	32	R	RTC 定时计数读出
sys_rtcmatch0	0x6C	32	RW	RTC 时钟定时中断 0
sys_rtcmatch1	0x70	32	RW	RTC 时钟定时中断 1
sys_rtcmatch2	0x74	32	RW	RTC 时钟定时中断 2

25.2.2 SYS_TOYWRITE0

中文名： TOY 计数器低 32 位数值
寄存器位宽： [31: 0]
偏移量： 0x24
复位值： 0x00000000

表 25- 2 TOY 计数器低 32 位写入寄存器

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

25.2.3 SYS_TOYWRITE1

中文名： TOY 计数器高 32 位数值
寄存器位宽： [31: 0]
偏移量： 0x28
复位值： 0x00000000

表 25- 3 TOY 计数器高 32 位写入寄存器

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

25.2.4 SYS_TOYREAD0

中文名： TOY 计数器低 32 位数值
寄存器位宽： [31: 0]
偏移量： 0x2C
复位值： 0x00000000

表 25- 4 TOY 计数器低 32 位读出寄存器

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

25.2.5 SYS_TOYREAD1

中文名: TOY 计数器高 32 位数值
寄存器位宽: [31: 0]
偏移量: 0x30
复位值: 0x00000000

表 25- 5 TOY 计数器高 32 位读出寄存器

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

25.2.6 SYS_TOYMATCH0/1/2

中文名: TOY 计数器中断寄存器 0/1/2
寄存器位宽: [31: 0]
偏移量: 0x34/38/3C
复位值: 0x00000000

表 25- 6 TOY 计数器中断寄存器 0/1/2

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

25.2.7 SYS_RTCCTRL

中文名: RTC 定时器中断寄存器 0/1/2
寄存器位宽: [31: 0]
偏移量: 0x40
复位值: 无

表 25- 7 RTC 定时器中断寄存器 0/1/2

位域	位域名称	访问	缺省	描述
31:24	保留	R	0	保留, 置 0
23	ERS	R	0	REN(bit13) 写状态
22:21	保留	R	0	保留, 置 0
20	RTS	R	0	Sys_rtctrim 写状态
19	RM2	R	0	Sys_rtcmatch2 写状态
18	RM2	R	0	Sys_rtcmatch2 写状态
17	RM0	R	0	Sys_rtcmatch0 写状态

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

25.2.8 SYS_RTCWRITE

中文名： RTC 计数器写入端口

寄存器位宽： [31: 0]

偏移量： 0x64

复位值： 0x00000000

表 25- 8 RTC 计数器写入寄存器

位域	位域名称	访问	缺省	描述
31:0	RTCWRITE	W	-	RTC 计数器写入寄存器

25.2.9 SYS_RTCREAD

中文名： RTC 计数器读出端口

寄存器位宽： [31: 0]

偏移量： 0x68

复位值： 0x00000000

表 25- 9 RTC 计数器读出寄存器

位域	位域名称	访问	缺省	描述
31:0	RTCREAD	R	0	RTC 计数器读出寄存器

25.2.10 SYS_RTCMATCH0/1/2

中文名： RTC 定时器中断寄存器 0/1/2

寄存器位宽： [31: 0]

偏移量： 0x6C/70/74

复位值： 0x00000000

表 25- 10 RTC 定时器中断寄存器 0/1/2

位域	位域名称	访问	缺省	描述
31:26	RTCMATCH0/1/2	RW	0	RTC 定时比较寄存器 0/1/2

26 功耗管理模块

26.1 概述

龙芯 2K0300 功耗管理模块提供系统功耗管理实现机制。

- 支持 Dynamic Frequency Scaling (DFS)，处理器核 DFS 控制。
- 系统时钟控制，模块时钟门控，多种方式调节频率。

26.2 动态功耗管理

龙芯 2K0300 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。

26.2.1 DFS 功能描述

龙芯 2K0300 芯片支持 NODE (LA264 处理器核+SCACHE) 动态频率调节 (DFS) 控制，软件直接控制实现。

龙芯 2K0300 芯片动态调频控制具体实现：

软件配置 LA264-DFS 控制寄存器 (LA264_DFS_CTRL) 实现，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 (120MHz)，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。

龙芯 2K0300 芯片 NODE (LA264+SCACHE) 动态频率调节 (DFS) 具体操作流程如下图所示：

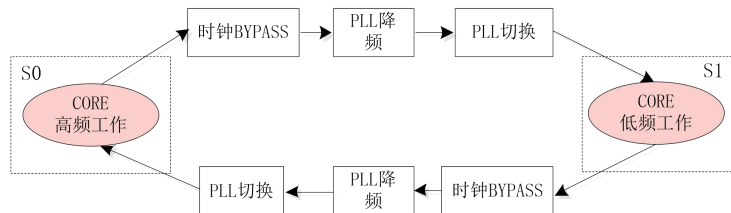


图26- 1 处理器核 DFS 操作流程

软件操作：

首先，2K0300 芯片进入低负荷工作模式，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 2K0300 处理器核低频工作模式切换。

与此类似，2K0300 芯片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 2K0300 芯片高频模式切换。

26.3 寄存器描述

本节介绍电源管理控制器相关寄存器，使用方法可参见下一节描述。

动态频率管理(DFS)的物理基地址为：0x16000108；

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）

26.3.1 DFS 寄存器描述

DFS 配置寄存器基址为：0x16000108。

26.3.1.1 LA264_DFS_CTRL : LA264/SCache DFS Control Register

表 26- 1 LA264 DFS Control 寄存器描述

地址偏移		电压域	属性
0x00		SOC	R/W
位域	描述		电压域
31:12	保留		
11:4	DFS_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
3	保留		-
2	DFS_STPCLK_n - 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	DFS_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	DFS_EN - R/W NODE 模块动态调频(DFS)使能位，高电平有效，默认值为 0。		SOC

26.3.2 WDT 寄存器描述

芯片内部看门狗(WDT)共一路，寄存器基址如下表：

表 26- 2 WDT 寄存器地址

WDT 基地址	设备	备注
0x1612_4000	WDT	WDT 独立控制

26.3.2.1 WD_EN : Watch Dog Enable Register

表 26- 3 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	

26.3.2.2 WD_SET : Watch Dog Set Register

表 26- 4 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。	

26.3.2.3 WD_Timer : Watch Dog Timer Register

表 26- 5 WD_TIMER 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 重填的值，复位后为全 1。	

27 GPIO

27.1 概述

龙芯 2K0300 共有 106 个 GPIO 引脚，全部与其他功能引脚复用。

27.2 寄存器描述

芯片内 GPIO 寄存器基地址分别为：

表 27- 1 GPIO 寄存器地址

GPIO 基地址	设备	备注
0x1610_4000	GPIO0~105	每个 GPIO 引脚均由一组寄存器控制

每个 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 27- 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		

27.3 访问地址

GPIO 的寄存器访问地址等于寄存器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 27- 3 GPIO 模块内部寄存器物理地址

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

表 27- 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	共 106 位	GPIO 输出使能，低有效。每位控制一个 GPIO 引脚。
0x10	GPIO_O	共 106 位	GPIO 输出值。每位控制一个 GPIO 引脚。
0x20	GPIO_I	共 106 位	GPIO 输入值。每位控制一个 GPIO 引脚。
0x30	GPIO_INT_EN	共 106 位	GPIO 中断使能。每位控制一个 GPIO 引脚。
0x40	GPIO_INT_POL	共 106 位	GPIO 中断极性。每位控制一个 GPIO 引脚。
0x50	GPIO_INT_EDGE	共 106 位	GPIO 中断边沿性。每位控制一个 GPIO 引脚。
0x60	GPIO_INT_CLR	共 106 位	GPIO 中断清除。每位控制一个 GPIO 引脚。
0x70	GPIO_INT_STS	共 106 位	GPIO 中断状态。每位控制一个 GPIO 引脚。
0x80	GPIO_INT_DUAL	共 106 位	GPIO 中断双沿模式。每位控制一个 GPIO 引脚。

表 27- 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	共 106 字节	GPIO 输出使能，低有效。每个字节控制一个 GPIO 引脚。
0x900	GPIO_O	共 106 字节	GPIO 输出值。每个字节控制一个 GPIO 引脚。
0xA00	GPIO_I	共 106 字节	GPIO 输入值。每个字节控制一个 GPIO 引脚。
0xB00	GPIO_INT_EN	共 106 字节	GPIO 中断使能。每个字节控制一个 GPIO 引脚。
0xC00	GPIO_INT_POL	共 106 字节	GPIO 中断极性。每个字节控制一个 GPIO 引脚。

0xD00	GPIO_INT_EDGE	共 106 字节	GPIO 中断边沿性。每个字节控制一个 GPIO 引脚。
0xE00	GPIO_INT_CLR	共 106 字节	GPIO 中断清除。每个字节控制一个 GPIO 引脚。
0xF00	GPIO_INT_STS	共 106 字节	GPIO 中断状态。每个字节控制一个 GPIO 引脚。
0xF80	GPIO_INT_DUAL	共 106 字节	GPIO 中断双沿模式。每个字节控制一个 GPIO 引脚。

27.4 控制寄存器

27.4.1 GPIO 方向控制

地址偏移：00-0fh 属性：R/W

默认值：FFFFFFFh 大小：16

表 27- 6 GPIO 方向按位控制寄存器

位域	名称	访问	描述
127:0	GPIO_OEN	R/W	每位对应于 GPIO[127:0] (实际有效路数共 106 路) 的方向控制。 0: 输出 1: 输入

地址偏移：800-87fh 属性：R/W

默认值：01010101h 大小：128

表 27- 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[127:0] (实际有效路数共 106 路) 的方向控制。 0: 输出 1: 输入

27.4.2 GPIO 输出值

地址偏移：10-1fh 属性：R/W

默认值：00000000h 大小：16

表 27- 8 GPIO 按位输出寄存器

位域	名称	访问	描述
127:0	GPIO_O	R/W	每位对应于 GPIO[127:0] (实际有效路数共 106 路) 的输出值。

地址偏移：900-97fh 属性：R/W

默认值：00000000h 大小：128

表 27- 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[127:0] (实际有效路数共 106 路) 的输出值。

27.4.3 GPIO 输入值

地址偏移: 20-2fh 属性: RO

默认值: N/A 大小: 16

表 27- 10 GPIO 按位输入寄存器

位域	名称	访问	描述
127:0	GPIO_I	RO	每位对应于 GPIO[127:0] (实际有效路数共 106 路) 的输入值。

地址偏移: a00-a7fh 属性: RO

默认值: 00000000h 大小: 128

表 27- 11 GPIO 按字节输入寄存器

位域	名称	访问	描述
7:0	GPIO_I	RO	每字节对应于每一位 GPIO[127:0] (实际有效路数共 106 路) 的输入值。

27.4.4 GPIO 中断使能

地址偏移: 30-3fh 属性: R/W

默认值: 00000000h 大小: 16

表 27- 12 GPIO 按位中断使能寄存器

位域	名称	访问	描述
127:0	GPIO_INT_EN	R/W	每位对应于 GPIO[127:0] (实际有效路数共 106 路) 的输入中断使能。 0: 关闭中断 1: 使能中断

地址偏移: b00-b7fh 属性: R/W

默认值: 00000000h 大小: 128

表 27- 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[127:0] (实际有效路数共 106 路) 的输入中断使能。 0: 关闭中断 1: 使能中断

27.4.5 GPIO 中断极性

地址偏移：40-4fh 属性：R/W

默认值：00000000h 大小：16

表 27- 14 GPIO 按位中断极性寄存器

位域	名称	访问	描述
127:0	GPIO_INT_POL	R/W	每位对应于 GPIO[127:0] (实际有效路数共 106 路) 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-c7fh 属性：R/W

默认值：00000000h 大小：128

表 27- 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[127:0] (实际有效路数共 106 路) 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。

27.4.6 GPIO 中断边沿

地址偏移：50-5fh 属性：R/W

默认值：00000000h 大小：16

表 27- 16 GPIO 按位中断边沿寄存器

位域	名称	访问	描述		
127:0	GPIO_INT_EDGE	R/W	每位对应于 GPIO[127:0]（实际有效路数共 106 路）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-d7fh 属性：R/W

默认值：00000000h 大小：128

表 27- 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[127:0]（实际有效路数共 106 路）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

27.4.7 GPIO 中断清除

地址偏移：60-6fh 属性：R/W

默认值：00000000h 大小：16

表 27- 18 GPIO 按位中断清除寄存器

位域	名称	访问	描述
127:0	GPIO_INT_CLR	R/W	对应于 GPIO[127:0]（实际有效路数共 106 路）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-e7fh 属性：R/W

默认值：00000000h 大小：128

表 27- 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[127:0]（实际有效路数共 106 路）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。

27.4.8 GPIO 中断状态

地址偏移：70-7fh 属性：R/W

默认值：00000000h 大小：16

表 27- 20 GPIO 按位中断状态寄存器

位域	名称	访问	描述
127:0	GPIO_INT_STS	R/W	对应于 GPIO[127:0]（实际有效路数共 106 路）的中断状态。 1：有中断 0：无中断

地址偏移：f00-f7fh 属性：R/W

默认值：00000000h

大小：128

表 27- 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[127:0]（实际有效路数共 106 路）的中断状态。 1：有中断 0：无中断

27.4.9 GPIO 中断双沿模式

地址偏移：80-8fh

属性：R/W

默认值：00000000h

大小：16

表 27- 22 GPIO 按位中断双沿寄存器

位域	名称	访问	描述
127:0	GPIO_INT_DUAL	R/W	每位对应于 GPIO[127:0]（实际有效路数共 106 路）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-ffffh

属性：R/W

默认值：00000000h

大小：128

表 27- 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[127:0]（实际有效路数共 106 路）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	初版发布
V1.01	<ol style="list-style-type: none"> 1. 修改 1 章节概述芯片主频描述，补充 QSPI 模式支持描述； 2. 补充 1.2.6~1.2.12 章节低速接口 DMA 功能； 3. 补充 1.2.7 章节 QSPI 功能描述； 4. 补充 2.2 章节 APB 时钟描述； 5. 补充 2.6 章节 STABLE/THSENS 时钟描述； 6. 修改 3.2 章节 CAN 地址空间描述； 7. 修正 3.4.2 章节 LIO 寄存器描述； 8. 修改 3.4.3 章节 PMU 寄存器描述； 9. 补充 3.4.5 章节寄存器中引脚驱动配置内容； 10. 修正 3.4.31 章节、3.4.34 章节、表 3-48 中寄存器描述笔误； 11. 修改 12.3.1、12.5.2、12.5.7、12.5.8 章节 I2C 描述内容； 12. 修正 17.2 章节 CAN 寄存器地址内容； 13. 修正 24.3 章节寄存器 HPET 部分功能配置描述； 14. 补充 26.3 章节相关寄存器地址信息。

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