

**LOONGSON**

**龙芯 1B 处理器  
用户手册**

V2.6

2021 年 4 月

龙芯中科技术股份有限公司

自主决定命运, 创新成就未来

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



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

## 版权声明

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

## 免责声明

本文档仅提供阶段性信息，所含内容可根据产品的实际情况随时更新，恕不另行通知。如因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, 电话(Tel):  
010-62546668

传真(Fax): 010-62600826

---

## 阅读指南

《龙芯 1B 处理器用户手册》主要介绍龙芯 1B 架构与寄存器描述。

## 修订历史

文档更新记录	文档名	龙芯 1B 处理器用户手册		
	版本号	V2.6		
	创建人	研发中心		
	创建日期	2021-4-20		
更新历史				
序号.	更新日期	更新人	版本号	更新内容
1	2010-6-7	研发中心	V1.0	1B 处理器初稿完成
2	2010-11-13	研发中心	V1.1	增加了芯片引脚排布, DDR 控制器信息等
3	2010-11-15	研发中心	V1.2	修改并进行标准排版
4	2010-11-15	研发中心	V1.3	修正了第五章 DDR 的部分错误
5	2011-05-08	研发中心	V1.4	修订了调试发现的错误
6	2011-05-17	研发中心	V1.5	修订了多个小问题
7	2011-11-15	研发中心	V1.6	GMAC0/1 的 RGMII 和 MII 模式需要配置才能使用 GPIO 配置和复用中修改 bug SPI 部分, 分频时钟明确是 DDR2_clk/2 DDR2 部分, 配置 16/32 位可配置 时钟分频部分有改动 添加了 LCD PAD 在不同显示模式下的对应关系 GPIO 寄存器描述修改
8	2012-4-11	研发中心	V1.7	NAND 部分寄存器说明修改 XTALI/O 与外部有源晶振、无源晶体连接方法
9	2012-4-20	研发中心	V1.8	PAD 封装位置和封装延迟 GPIO 复位值和方向 GMAC0/1 在 MII 模式下信号处理 Wdog 地址修改 USB 启动需要复位
10	2012-05-26	研发中心	V1.9	针对修改意见, 做了 GPIO/LCD/DMA/SPI/UART /I2C/NAND/CLOCK 的修改
11	2014-07-30	研发中心	V2.0	增加质量等级和封装顶视图
12	2015-3-11	研发中心	V2.1	增加电特性, CAN 的速率计算

13	2015-4-1	研发中心	V2.2	补充质量等级描述
14	2016-5-4	研发中心	V2.3	24.1 节增加 RTC 功耗说明，增加焊接要求
15	2020-1-5	研发中心	V2.4	增加不建议使用 RGMII 接口的说明 表 2-1 修正 XTALI 接有源晶振的描述 表 22-1 增加 NAND_D7 配置描述 增加管脚上下拉，DC 特性描述
16	2021-4-10	研发中心	V2.5	质量等级增加普通工业级 推荐工作条件 VDD_1V2 改为 1.25V
17	2021-4-20	研发中心	V2.6	更新质量等级相关参数

手册信息反馈: [service@loongson.cn](mailto:service@loongson.cn)

## 目 录

.....	1
1 概述.....	1
1.1 体系结构框图.....	1
1.2 芯片主要功能.....	2
1.2.1 GS232 CPU.....	2
1.2.2 DDR2.....	3
1.2.3 LCD Controller.....	3
1.2.4 USB2.0.....	3
1.2.5 AC97.....	3
1.2.6 GMAC.....	4
1.2.7 SPI.....	4
1.2.8 UART.....	4
1.2.9 I <sup>2</sup> C.....	4
1.2.10 PWM.....	5
1.2.11 CAN.....	5
1.2.12 RTC.....	5
1.2.13 GPIO.....	5
1.2.14 NAND.....	5
1.2.15 INT controller.....	5
1.2.16 Watchdog.....	5
1.2.17 功耗.....	6
1.2.18 其它.....	7
1.3 质量等级.....	7
2 芯片引脚定义.....	9
2.1 1B 引脚分布图.....	9
2.2 封装顶视图.....	16
2.3 系统相关引脚定义（6）.....	17
2.4 LCD 引脚定义（20）.....	17
2.5 PLL 引脚定义（4）.....	18
2.6 VR 引脚定义（6）.....	18
2.7 DDR2 引脚定义（71）.....	18
2.8 USB 引脚定义(11).....	20
2.9 EJTAG 引脚定义(6).....	20
2.10 GMAC0 引脚定义(15).....	20
2.11 GMAC1 引脚定义(4).....	21
2.12 AC97 引脚定义(5).....	21
2.13 SPI 引脚定义(7).....	21
2.14 UART 引脚定义(20).....	21
2.15 I <sup>2</sup> C 引脚定义(2).....	22
2.16 CAN 引脚定义(4).....	22
2.17 NAND 引脚定义(14).....	22

2.18	PWM 引脚定义(4).....	23
2.19	电源/地引脚(58).....	23
3	地址空间分配 .....	24
3.1	一级 AXI 交叉开关上模块的地址空间.....	24
3.2	AXI MUX 下各模块的地址空间.....	24
3.3	APB 各模块的地址空间分配.....	24
4	DDR2.....	26
4.1	DDR2 SDRAM 控制器特性 .....	26
4.2	DDR2 SDRAM 读协议.....	26
4.3	DDR2 SDRAM 写协议 .....	27
4.4	DDR2 SDRAM 参数设置顺序.....	27
4.5	DDR2 SDRAM 采样模式配置.....	28
4.6	DDR2 SDRAM PAD 驱动配置.....	28
4.7	DDR2 16 位工作模式配置.....	28
5	LCD .....	29
5.1	特性.....	29
5.1.1	数据格式 .....	29
5.2	寄存器.....	29
6	GMAC0.....	35
7	GMAC1.....	36
7.1	配置成 MAC 的连接和复用方式.....	36
7.2	GMAC1 外部信号复用和配置.....	36
8	USB HOST.....	38
8.1	总体概述.....	38
8.2	USB 主机控制器寄存器 .....	38
8.2.1	EHCI 相关寄存器.....	38
8.2.2	Capability 寄存器 .....	38
8.2.3	Operational 寄存器 .....	38
8.3	OHCI 相关寄存器.....	39
8.3.1	Operational 寄存器 .....	39
9	SPI0 .....	41
9.1	SPI 控制器结构.....	41
9.2	SPI 控制器寄存器.....	42
9.2.1	控制寄存器 (SPCR) .....	42
9.2.2	状态寄存器 (SPSR) .....	42
9.2.3	数据寄存器 (TxFIFO/RxFIFO) .....	43
9.2.4	外部寄存器 (SPER) .....	43
9.2.5	参数控制寄存器 (SFC_PARAM) .....	43
9.2.6	片选控制寄存器 (SFC_SOFTCS) .....	44
9.2.7	时序控制寄存器 (SFC_TIMING) .....	44
9.3	接口时序.....	44
	SPI 主控制器外部接口时序图 .....	44
	SPI Flash 访问时序图.....	45

9.4	SPI FLASH 控制器使用指南.....	46
	SPI 主控制器的读写操作.....	46
	硬件 SPI Flash 读.....	47
	混合访问 SPI Flash 和 SPI 主控制器.....	47
10	SPI1.....	49
10.1	SPI 主控制器结构.....	49
11	Conf and Interrupt.....	50
11.1	配置和中断控制器总体描述.....	50
11.2	中断控制器寄存器描述.....	51
12	DMA.....	53
12.1	DMA 控制器结构描述.....	53
12.2	DMA 控制器与 APB 设备的交互.....	53
12.3	DMA 控制器.....	53
	12.3.1 ORDER_ADDR_IN.....	53
	12.3.2 DMA_ORDER_ADDR.....	54
	12.3.3 DMA_SADDR.....	54
	12.3.4 DMA_DADDR.....	55
	12.3.5 DMA_LENGTH.....	55
	12.3.6 DMA_STEP_LENGTH.....	55
	12.3.7 DMA_STEP_TIMES.....	56
	12.3.8 DMA_CMD.....	56
13	UART.....	58
13.1	UART 控制器结构.....	58
13.2	UART 控制器寄存器.....	59
	13.2.1 数据寄存器 (DAT).....	60
	13.2.2 中断使能寄存器 (IER).....	60
	13.2.3 中断标识寄存器 (IIR).....	60
	13.2.4 FIFO 控制寄存器 (FCR).....	61
	13.2.5 线路控制寄存器 (LCR).....	61
	13.2.6 MODEM 控制寄存器 (MCR).....	62
	13.2.7 线路状态寄存器 (LSR).....	62
	13.2.8 MODEM 状态寄存器 (MSR).....	63
	13.2.9 分频锁存器.....	63
14	CAN.....	65
14.1	概述.....	65
14.2	CAN 控制器结构.....	65
14.3	标准模式.....	66
	14.3.1 标准模式地址表.....	66
	14.3.2 控制寄存器 (CR).....	67
	14.3.3 命令寄存器 (CMR).....	68
	14.3.4 状态寄存器 (SR).....	68
	14.3.5 中断寄存器 (IR).....	68
	14.3.6 验收代码寄存器 (ACR).....	69



14.3.7	验收屏蔽寄存器 (AMR) .....	69
14.3.8	发送缓冲区列表 .....	69
14.3.9	接收缓冲区列表 .....	70
14.4	扩展模式.....	70
14.4.1	扩展模式地址表 .....	70
14.4.2	模式寄存器 (MOD) .....	70
14.4.3	命令寄存器 (CMR) .....	71
14.4.4	状态寄存器 (SR) .....	71
14.4.5	中断寄存器 (IR) .....	72
14.4.6	中断使能寄存器 (IER) .....	72
14.4.7	仲裁丢失捕捉寄存器 (IER) .....	72
14.4.8	错误警报限制寄存器 (EMLR) .....	73
14.4.9	RX 错误计数寄存器 (RXERR) .....	74
14.4.10	TX 错误计数寄存器 (TXERR) .....	74
14.4.11	验收滤波器.....	74
14.4.12	RX 信息计数寄存器 (RMCR) .....	74
14.5	公共寄存器.....	74
14.5.1	总线定时寄存器 0 (BTR0) .....	74
14.5.2	总线定时寄存器 1 (BTR1) .....	75
14.5.3	输出控制寄存器 (OCR) .....	75
15	AC97.....	76
15.1	AC97 结构描述.....	76
15.2	AC97 控制器寄存器 .....	76
15.2.1	CSR 寄存器 .....	77
15.2.2	OCC 寄存器.....	77
15.2.3	ICC 寄存器 .....	77
15.2.4	(输入输出) 通道寄存器配置.....	78
15.2.5	Codec 寄存器访问命令 .....	78
15.2.6	中断状态寄存器/中断掩膜寄存器 .....	79
15.2.7	中断状态/清除寄存器 .....	79
15.2.8	OC 中断清除寄存器.....	79
15.2.9	IC 中断清除寄存器.....	80
15.2.10	CODEC WRITE 中断清除寄存器 .....	80
15.2.11	CODEC READ 中断清除寄存器 .....	80
16	I2C .....	81
16.1	概述.....	81
16.2	I2C 控制器结构.....	81
16.3	I <sup>2</sup> C 控制器寄存器说明 .....	82
16.3.1	分频锁存器低字节寄存器 (PRERlo) .....	82
16.3.2	分频锁存器高字节寄存器 (PRERhi) .....	82
16.3.3	控制寄存器 (CTR) .....	83
16.3.4	发送数据寄存器 (TXR) .....	83
16.3.5	接受数据寄存器 (RXR) .....	83

16.3.6	命令控制寄存器 (CR)	83
16.3.7	状态寄存器 (SR)	84
17	PWM	85
17.1	概述	85
17.2	PWM 寄存器说明	85
18	RTC	87
18.1	概述	87
18.2	寄存器描述	87
18.2.1	寄存器地址列表	87
18.2.2	SYS_TOYWRITE0	88
18.2.3	SYS_TOYWRITE1	88
18.2.4	SYS_TOYMATCH0/1/2	88
18.2.5	SYS_RTCCTRL	89
18.2.6	SYS_RTCMATCH0/1/2	90
19	NAND	91
19.1	NAND 控制器结构描述	91
19.2	NAND 控制器寄存器配置描述	91
19.2.1	NAND_CMD (地址: BFE7_8000)	91
19.2.2	ADDR_L (地址: BFE7_8004)	92
19.2.3	ADDR_H (地址: BFE7_8008)	92
19.2.4	NAND_TIMING (地址: BFE7_800C)	92
19.2.5	ID_L (地址: BFE7_8010)	92
19.2.6	STATUS & ID_H (地址: BFE7_8014)	92
19.2.7	NAND_PARAMETER (地址: BFE7_8018)	92
19.2.8	NAND_OP_NUM (地址: BFE7_801C)	93
19.2.9	CS_RDY_MAP (地址: BFE7_8020)	93
19.2.10	DMA_ADDRESS (地址: BFE7_8040)	93
19.3	NAND ADDR 说明	93
20	WATCHDOG	96
20.1	概述	96
20.2	WATCHDOG 寄存器描述	96
20.2.1	WDT_EN 地址: (0XBFE5_C060)	96
20.2.2	WDT_SET (地址: 0XBFE5_C068)	97
20.2.3	WDT_timer (地址: 0XBFE5_C064)	97
21	Clock Management	98
21.1	CLOCK 模块结构描述	98
21.2	CLOCK 配置描述	98
21.3	系统其它 CLOCK 描述	99
22	GPIO and MUX	100
22.1	GPIO 结构描述	100
22.2	GPIO 寄存器描述	103
22.3	MUX 寄存器描述	104
23	电气特性	106

23.1	电源域.....	106
23.2	系统复位.....	106
23.3	推荐的工作条件.....	107
23.4	绝对最大额定值.....	107
23.5	上电过程要求.....	107
23.6	参考时钟.....	108
23.7	GPIO DC 特性.....	108
24	热特性.....	109
24.1	焊接温度.....	109

## 图 目 录

图 1-1 1B 芯片结构图.....	2
图 10-1 SPI 主控制器结构 .....	42
图 10-2 SPI 主控制器时序图 .....	45
图 16-1 AC97 应用系统.....	76
图 21-1 看门狗的结构图 .....	96
图 25-1 焊接回流曲线 .....	109

## 表 目 录

表 18-1 四路控制器描述 .....	85
表 18-2 控制寄存器描述 .....	85
表 18-3 主计数器设置 .....	85
表 18-4 高脉冲计数器设置 .....	85
表 18-5 低脉冲计数器设置 .....	85
表 18-6 控制寄存器设置 .....	86
表 24-1 1B 电源域.....	106
表 24-2 1B 上电配置引脚汇总.....	106
表 24-3 推荐的工作条件 .....	107
表 24-4 绝对最大额定值 .....	107
表 24-5 单端参考时钟波形 .....	108
表 24-6 GPIO DC 特性 .....	108
表 25-1 回流焊接温度要求 .....	109

## 1 概述

龙芯 1B 芯片是基于 GS232 处理器核的片上系统，具有高性价比，可广泛应用于工业控制、家庭网关、信息家电、医疗器械和安全应用等领域。1B 采用 Wire Bond BGA256 封装。

1B 芯片具有以下关键特性：

- 集成一个 GS232 双发射龙芯处理器核，指令和数据 L1 Cache 各 8KB
- 集成一路 LCD 控制器，最大分辨率可支持到 1920\*1080@60Hz/16bit
- 集成 2 个 10M/100M 自适应 GMAC
- 集成 1 个 16/32 位 133MHz DDR2 控制器
- 集成 1 个 USB 2.0 接口，兼容 EHCI 和 OHCI
- 集成 1 个 8 位 NAND FLASH 控制器,最大支持 32GB
- 集成中断控制器，支持灵活的中断设置
- 集成 2 个 SPI 控制器，支持系统启动
- 集成 AC97 控制器
- 集成 1 个全功能串口、1 个四线串口和 10 个两线串口
- 集成 3 路 I2C 控制器，兼容 SMBUS
- 集成 2 个 CAN 总线控制器
- 集成 61 个 GPIO 端口
- 集成 1 个 RTC 接口
- 集成 4 个 PWM 控制器
- 集成看门狗电路

### 1.1 体系结构框图

1B 芯片内部顶层结构由 AXI XBAR 交叉开关互连，其中 GS232、DC、AXI\_MUX 作为主设备通过 3X3 交叉开关连接到系统；DC、AXI\_MUX 和 DDR2 作为从设备通过 3X3 交叉开关连接到系统。在 AXI\_MUX 内部实现了多个 AHB 和 APB 模块到顶层 AXI 交叉开关的连接，其中 DMA\_MUX、GMAC0、GMAC1、USB 被 AXI\_MUX 选择作为主设备访问交叉开关；AXI\_MUX(包括 confreg、SPI0、SPI1)、AXI2APB、GMAC0、GMAC1、USB 等作为从设备被来自 AXI\_MUX 的主设备访问。在 AXI2APB 内部实现了系统对内部 APB 接口设备的访问，这些设备包括 Watch Dog、RTC、PWM、I2C、CAN、NAND、UART 等。

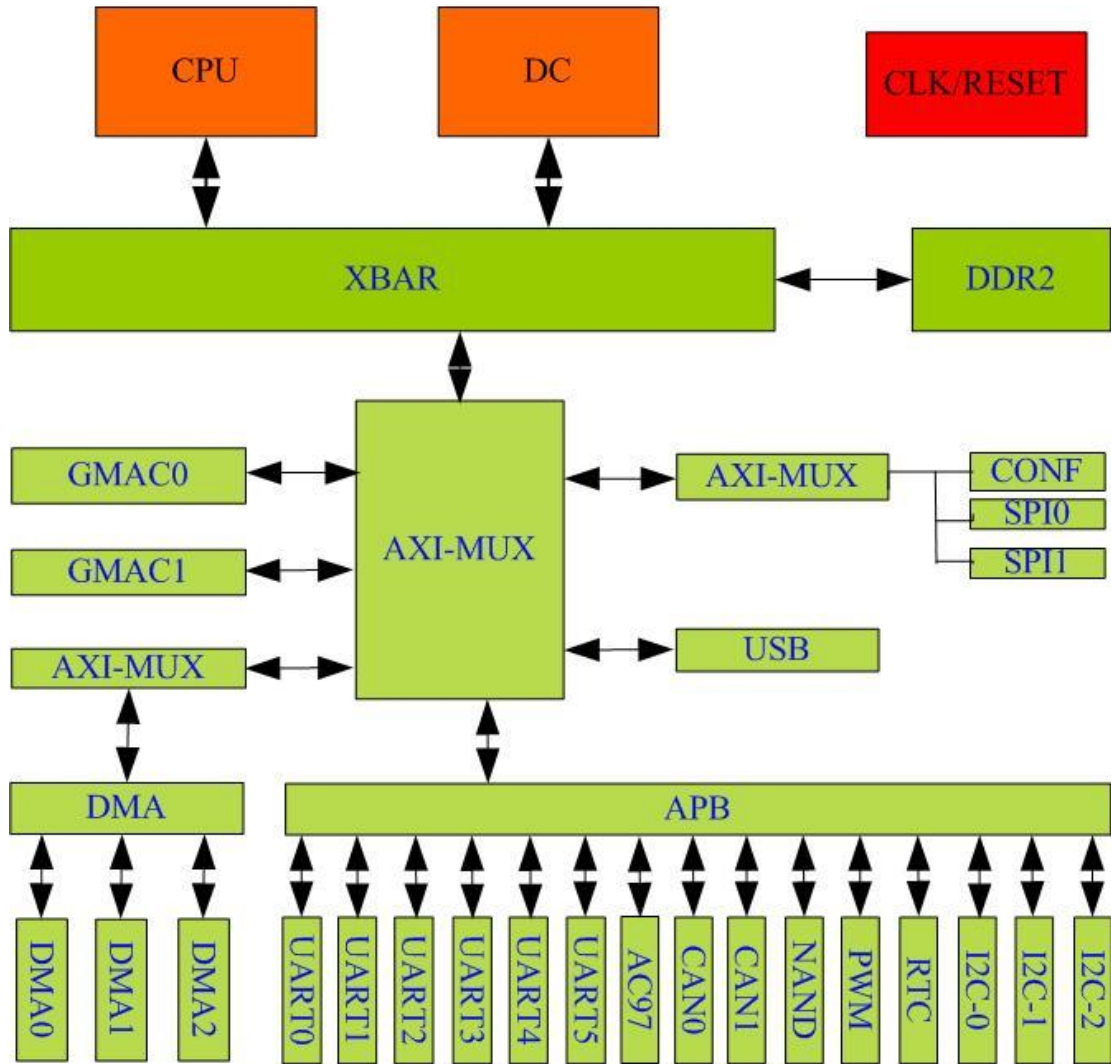


图 1-1 1B 芯片结构图

## 1.2 芯片主要功能

1B 芯片支持以下功能：

### 1.2.1 GS232 CPU

龙芯 232 核是一款实现 MIPS32 兼容且支持 EJTAG 调试的双发射处理器，通过采用转移预测、寄存器重命名、乱序发射、乱序执行的指令 CACHE、非阻塞的数据 CACHE、写合并收集等技术来提高流水线的效率。

- 双发射五级流水、乱序发射、乱序执行
- 8KB 指令 Cache+8KB 数据 Cache，4 路组相连，指令 CACHE 支持乱序预测
- 6 项 BRQ、16 项的 QUEUE
- 动态转移预测、地址返回栈
- 32 项 JTLB，4 项 ITLB、8 项 DTLB
- 两个定点 ALU 部件

- 支持非阻塞的 Cache 访问技术，4 项 load 队列、2 项 store 队列、3 项 miss 队列，最多容忍 5 条 store 指令 Cache 不命中和 4 条 load 指令 Cache 不命中
- 支持 cached store 指令的写合并和 uncached 写加速技术
- 支持 cache lock 技术和预取指令
- 支持流水线暂停模式
- 支持向量中断，可配置支持快速中断响应，最多 8 个时钟周期进入中断处理程序
- 支持 EJTAG 调试

### 1.2.2 DDR2

- 32 位 DDR2 控制器
- 遵守 DDR2 DDR 的行业标准（JESD79-2B）
- 一共含有 18 位的地址总线（即：15 位的行列地址总线和 3 位的逻辑 Bank 总线）
- 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 33-133MHZ 工作频率

### 1.2.3 LCD Controller

- 屏幕大小可达 1920\*1080
- 硬件光标
- 伽玛校正
- 最高像素时钟 172MHz
- 支持线性显示缓冲
- 上电序列控制
- 支持 16 位/24 位 LCD

### 1.2.4 USB2.0

- 1 个独立的 USB2.0 的 HOST ports 及 PHY
- 兼容 USB1.1 和 USB2.0
- 内部 EHCI 控制和实现高速传输可达 480Mbps
- 内部 OHCI 控制和实现全速和低速传输 12Mbps 和 1.5Mbps

### 1.2.5 AC97

- 支持 16, 18 和 20 位采样精度，支持可变速率
- 最高达 48KHz
- 2 频道立体声输出
- 支持麦克风输入



### 1.2.6 GMAC

- 两路 10/100Mbps 自适应以太网控制器
- 双网卡均兼容 IEEE 802.3
- 对外部 PHY 实现 MII 接口
- 半双工/全双工自适应
- 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

### 1.2.7 SPI

- 支持 2 路 SPI 接口
- SPI0 支持系统启动
- 极性和相位可编程的串行时钟
- 可在等待模式下对 SPI 进行控制

### 1.2.8 UART

- 集成 12 个串口
- 在寄存器与功能上兼容 NS16550A
- 全双工异步数据接收/发送
- 最高支持 115200bps
- 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 I<sup>2</sup>C

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

### 1.2.10 PWM

- 提供 4 路可配置 PWM 输出
- 数据宽度 32 位
- 定时器功能
- 计数器功能

### 1.2.11 CAN

- 支持 2 个独立 CAN 总线接口
- 每路 CAN 接口均支持 CAN2.0A/B 协议
- 支持 CAN 协议扩展

### 1.2.12 RTC

- 计时精确到 0.1 秒
- 可产生 3 个计时中断
- 支持定时开关机功能

### 1.2.13 GPIO

- 61 位 GPIO
- 支持位操作

### 1.2.14 NAND

- 支持最大单颗 NAND FLASH 为 32GB
- 共 4 个片选 CS
- 数据宽度 8bit
- 支持 SLC
- 支持页大小 2048Byte

### 1.2.15 INT controller

- 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
- 支持固定中断优先级

### 1.2.16 Watchdog

- 16 比特计数器及初始化寄存器
- 低功耗模式暂停功能

### 1.2.17 功耗

- 典型工作状态 0.3-0.5W

## 1.2.18 其它

- 测试访问口控制器 JTAG

## 1.3 质量等级

龙芯 1B 芯片有工业级和商业级两种，其主要特征如下：

配置	商业级	普通工业级	专用工业级
工作温度	0℃~70℃	-40℃~85℃	-40℃~85℃
是否筛选	—	√	√
是否质量一致性试验	—	√	√
是否老炼	—	—	√
质量一致性试验标准	—	GB 12750-2006 GB 4937-1995 GJB 548B-2005	GB 12750-2006 GB 4937-1995 GJB 548B-2005

龙芯 1B 芯片为验证是否符合国标以及其他标准的相关质量要求，依据国家标准，制定了器件专用的详细规范，并依据详细规范，进行了鉴定检验试验。鉴定检验试验中包含器件详细规范中规定的各种高温、高湿、老化寿命以及 ESD 等一系列可靠性试验。关键试验项目见下表，其余所有试验项目详见器件的详细规范。

分组	试验项目	方法	条件	LTPD (c=0) 或样品数 (允许失效数)
B5	a) 温度快速变化 b) 外部目检 c) 强加速稳态湿热 <sup>a</sup> d) 电测试	GB/T 4937 第 III 篇 1.1 GB/T 4589.1 4.3.1.1 GB/T 4937 第 III 篇 5C	10 次循环  130℃/85%RH, 24h 或 85℃/85%RH, 24h A2、A3 分组	10
C2	静电放电敏感度 (ESD)	ESDA/JEDEC JS-001-2012	2000V	18 (0)
C5 <sup>a</sup>	a) 温度快速变化 b) 外部目检 c) 强加速稳态湿热 <sup>a</sup>	GB/T 4937 第 III 篇 1.1 GB/T 4589.1 4.3.1.1 GB/T 4937.4-2012 (非偏置)	500 次循环  130℃/85%RH, 24h	10
C5a	盐雾	GB/T 4937 第 III 篇 8	24h	50
C7	a) 加速稳态湿热 b) 电测试	GB/T 4937 第 III 篇 5B	85℃/85%RH, 500h A2、A3 分组	15
C8	电耐久性 (寿命)	附录 A	T <sub>a</sub> =85℃, 1000h	10

<sup>a</sup> 工业级按照 130℃标准试验，商业级按照 85℃标准试验。

龙芯 1B 芯片和多数半导体器件一样，其失效率符合浴盆曲线模型。龙芯 1B 工业级芯片为了保证能够更长期、稳定、可靠地工作，并且能够适应更苛刻的环境温度要求，对芯片进行了可靠性筛选，以剔除早期失效的芯片。这种可靠性筛选是 100%的试验，通过筛

选的为符合工业级要求的芯片。

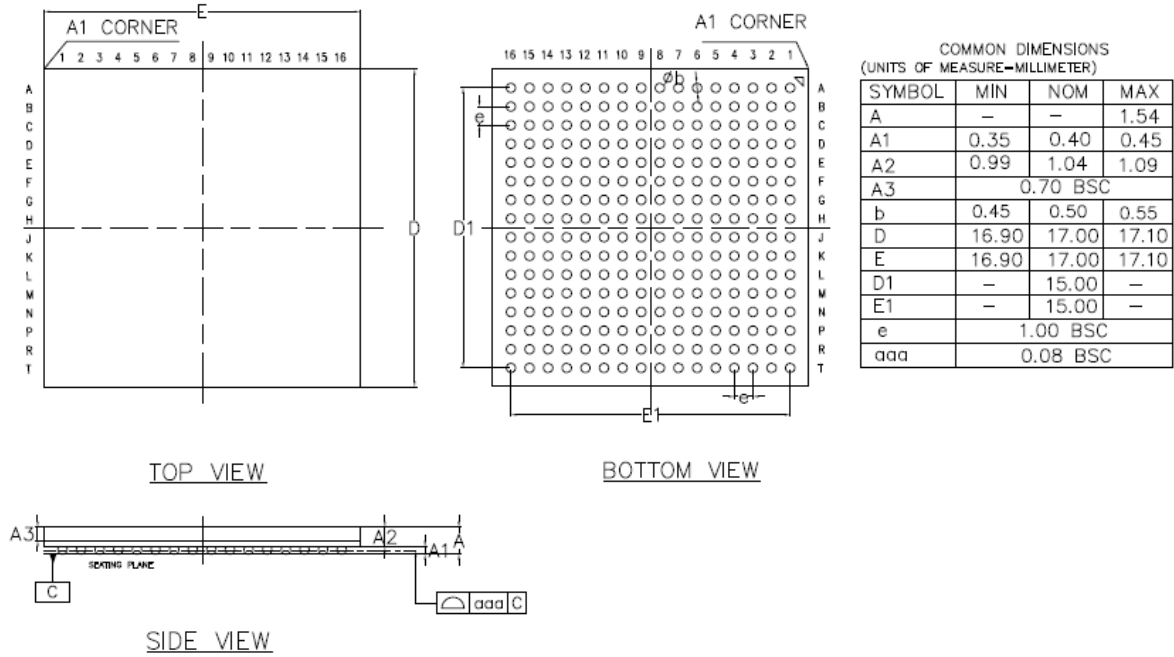
龙芯 1B 筛选试验主要内容如下：

筛选项目	方法和条件（概要）	要求
1、目检	标识清晰，无污渍，焊球无氧化，芯片完好	100%
2、稳定性烘培	100°C，24h	100%
3、温度快速变化	最高和最低储存温度下，10次循环	100%
4、编序列号		100%
5、终点电测试	三温，记录所有测试数据	100%
6、外部目检	标识清晰，无污渍，焊球无氧化，芯片完好	100%

## 2 芯片引脚定义

### 2.1 1B 引脚分布图

1B 采用 BGA256 封装形式，封装尺寸如下图所示：



芯片的引脚 PAD 的排布列常用信号延迟如下表所示：

表 2-1 芯片引脚 PAD 的排布列表

Location	Name	Delay (um)
K15	AC97_BIT_CLK	7630
J16	AC97_DATA_I	8471
J15	AC97_DATA_O	8426
K14	AC97_RESET	7351
K13	AC97_SYNC	6592
T12	CAN0_RX	9501
R12	CAN0_TX	8338
P12	CAN1_RX	7521
P13	CAN1_TX	8717
B01	DDR2_A00	12623
A01	DDR2_A01	13928
D02	DDR2_A02	9792
C02	DDR2_A03	11318
B02	DDR2_A04	11496
A02	DDR2_A05	12195

D03	DDR2_A06	9037
C03	DDR2_A07	9723
B03	DDR2_A08	10375
A03	DDR2_A09	11434
D04	DDR2_A10	8109
C04	DDR2_A11	8373
B04	DDR2_A12	9582
A04	DDR2_A13	9438
D05	DDR2_A14	6126
C07	DDR2_BA0	6741
B07	DDR2_BA1	7790
A07	DDR2_BA2	8917
C08	DDR2_CASN	6745
C05	DDR2_CKE0	8537
A05	DDR2_CKN0	9282
B05	DDR2_CKP0	9209
H01	DDR2_DQ00	10204
E01	DDR2_DQ01	12521
J02	DDR2_DQ02	8263
G02	DDR2_DQ03	9832
F01	DDR2_DQ04	11482
J01	DDR2_DQ05	9568
C01	DDR2_DQ06	13879
H02	DDR2_DQ07	10212
G03	DDR2_DQ08	8558
E02	DDR2_DQ09	10758
H04	DDR2_DQ10	6724
E03	DDR2_DQ11	10387
E04	DDR2_DQ12	10798
H03	DDR2_DQ13	6836
F04	DDR2_DQ14	6311
G04	DDR2_DQ15	3884
B10	DDR2_DQ16	8866
A12	DDR2_DQ17	12093
A09	DDR2_DQ18	9068
B11	DDR2_DQ19	10432
B12	DDR2_DQ20	10116
B09	DDR2_DQ21	7961
B13	DDR2_DQ22	10991
A10	DDR2_DQ23	9302
D10	DDR2_DQ24	6637

C12	DDR2_DQ25	9230
C09	DDR2_DQ26	6748
D13	DDR2_DQ27	9679
C13	DDR2_DQ28	9341
D09	DDR2_DQ29	7575
D12	DDR2_DQ30	7419
C10	DDR2_DQ31	6687
D01	DDR2_DQM0	12838
F03	DDR2_DQM1	8257
A13	DDR2_DQM2	11310
C11	DDR2_DQM3	6869
G01	DDR2_DQS0	10939
F02	DDR2_DQS1	8533
A11	DDR2_DQS2	10187
D11	DDR2_DQS3	7020
D06	DDR2_GATEI0	6628
D07	DDR2_GATEI1	3705
C06	DDR2_GATEO0	6888
B06	DDR2_GATEO1	8649
A06	DDR2_ODT0	8994
D08	DDR2_RASN	5651
A08	DDR2_SCSN0	8745
B08	DDR2_WEN	7745
L16	EJTAG_TCK	8933
L15	EJTAG_TDI	7868
K16	EJTAG_TDO	8685
L14	EJTAG_TMS	7823
L13	EJTAG_TRST	7524
N11	GMAC0_MDCK	6878
N12	GMAC0_MDIO	8206
R08	GMAC0_RX_CLK_I	6849
N09	GMAC0_RX_CTL_I	7076
P09	GMAC0_RX0	6973
N10	GMAC0_RX1	7360
P10	GMAC0_RX2	9231
P11	GMAC0_RX3	6844
T08	GMAC0_TX_CLK_I	8736
R11	GMAC0_TX_CLK_O	9410
T11	GMAC0_TX_CTL_O	7991
R10	GMAC0_TX0	8973
T10	GMAC0_TX1	9242



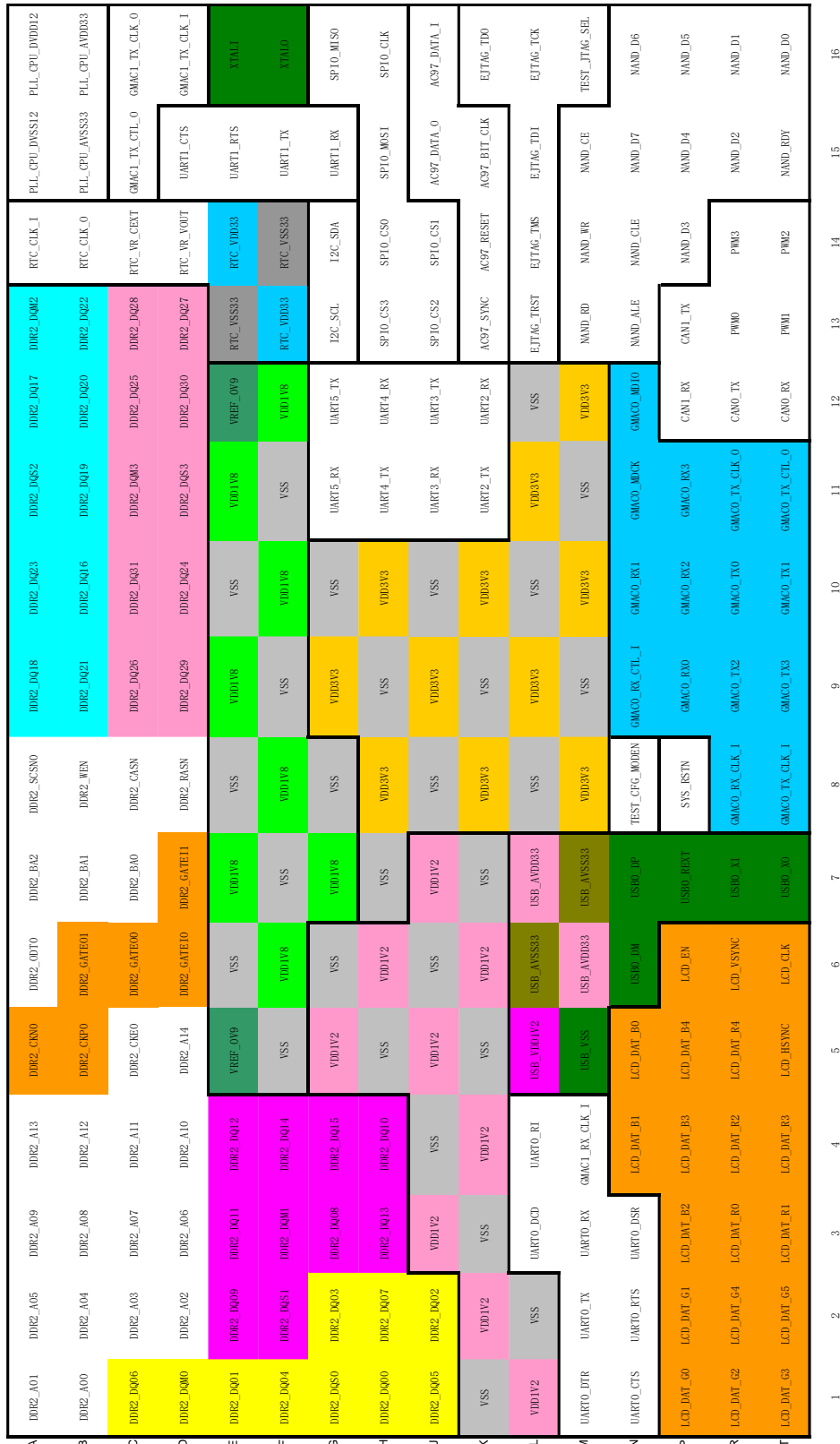
R09	GMAC0_TX2	9905
T09	GMAC0_TX3	10857
M04	GMAC1_RX_CLK_I	7499
D16	GMAC1_TX_CLK_I	10743
C16	GMAC1_TX_CLK_O	11938
C15	GMAC1_TX_CTL_O	10929
G13	I2C_SCL	9062
G14	I2C_SDA	7886
T06	LCD_CLK	9050
N05	LCD_DAT_B0	7837
N04	LCD_DAT_B1	9278
P03	LCD_DAT_B2	9723
P04	LCD_DAT_B3	9653
P05	LCD_DAT_B4	7609
P01	LCD_DAT_G0	11258
P02	LCD_DAT_G1	10460
R01	LCD_DAT_G2	13011
T01	LCD_DAT_G3	12938
R02	LCD_DAT_G4	11774
T02	LCD_DAT_G5	11960
R03	LCD_DAT_R0	10699
T03	LCD_DAT_R1	13006
R04	LCD_DAT_R2	9332
T04	LCD_DAT_R3	9489
R05	LCD_DAT_R4	8482
P06	LCD_EN	7889
T05	LCD_HSYNC	9617
R06	LCD_VSYNC	8085
N13	NAND_ALE	9117
M15	NAND_CE	9198
N14	NAND_CLE	8809
T16	NAND_D0	11717
R16	NAND_D1	11235
R15	NAND_D2	10371
P14	NAND_D3	9032
P15	NAND_D4	9769
P16	NAND_D5	10210
N16	NAND_D6	9839
N15	NAND_D7	8909
M13	NAND_RD	7573
T15	NAND_RDY	10896

M14	NAND_WR	7961
B16	PLL_CPU_AVDD33	17840
B15	PLL_CPU_AVSS33	18831
A16	PLL_CPU_DVDD12	12759
A15	PLL_CPU_DVSS12	12807
R13	PWM0	9034
T13	PWM1	9503
T14	PWM2	10205
R14	PWM3	9905
A14	RTC_CLK_I	11584
B14	RTC_CLK_O	10493
E14	RTC_VDD33	33449
F13	RTC_VDD33	
C14	RTC_VR_CEXT	12177
D14	RTC_VR_VOUT	11997
E13	RTC_VSS33	23636
F14	RTC_VSS33	
H16	SPI0_CLK	9620
H14	SPI0_CS0	7708
J14	SPI0_CS1	6530
J13	SPI0_CS2	6324
H13	SPI0_CS3	7239
G16	SPI0_MISO	9904
H15	SPI0_MOSI	8645
P08	SYS_RSTN	
N08	TEST_CFG_MODEN	
M16	TEST_JTAG_SEL	
N01	UART0_CTS	9660
L03	UART0_DCD	6806
N03	UART0_DSR	8056
M01	UART0_DTR	9219
L04	UART0_RI	6301
N02	UART0_RTS	8689
M03	UART0_RX	7353
M02	UART0_TX	8201
D15	UART1_CTS	10124
E15	UART1_RTS	9692
G15	UART1_RX	7806
F15	UART1_TX	8614
K12	UART2_RX	10114
K11	UART2_TX	7963

J11	UART3_RX	8783
J12	UART3_TX	7119
H12	UART4_RX	7246
H11	UART4_TX	6701
G11	UART5_RX	9284
G12	UART5_TX	6569
L07	USB_AVDD33	
M06	USB_AVDD33	
L06	USB_AVSS33	
M07	USB_AVSS33	
L05	USB_VDD1V2	
M05	USB_VSS	
N06	USB0_DM	
N07	USB0_DP	
P07	USB0_REXT	
R07	USB0_XI	
T07	USB0_XO	
G05	VDD1V2	
H06	VDD1V2	
J03	VDD1V2	
J05	VDD1V2	
J07	VDD1V2	
K02	VDD1V2	
K04	VDD1V2	
K06	VDD1V2	
L01	VDD1V2	
E07	VDD1V8	
E09	VDD1V8	
E11	VDD1V8	
F06	VDD1V8	
F08	VDD1V8	
F10	VDD1V8	
F12	VDD1V8	
G07	VDD1V8	
G09	VDD3V3	
H08	VDD3V3	
H10	VDD3V3	
J09	VDD3V3	
K08	VDD3V3	
K10	VDD3V3	
L09	VDD3V3	

L11	VDD3V3	
M08	VDD3V3	
M10	VDD3V3	
M12	VDD3V3	
E05	VREF_0V9	
E12	VREF_0V9	
E06	VSS	
E08	VSS	
E10	VSS	
F05	VSS	
F07	VSS	
F09	VSS	
F11	VSS	
G06	VSS	
G08	VSS	
G10	VSS	
H05	VSS	
H07	VSS	
H09	VSS	
J04	VSS	
J06	VSS	
J08	VSS	
J10	VSS	
K01	VSS	
K03	VSS	
K05	VSS	
K07	VSS	
K09	VSS	
L02	VSS	
L08	VSS	
L10	VSS	
L12	VSS	
M09	VSS	
M11	VSS	
E16	XTALI	10899
F16	XTALO	10414

2.2 封装顶视图



## 2.3 系统相关引脚定义（6）

表 2-1 系统时钟引脚定义

No.	信号名称	方向	上下拉	描述	电压域
1	XTALI	I		外部无源晶体时钟输入； 外部有源晶振输入（工业级产品应当使用有源晶振）	core
2	XTALO	O		外部无源晶体时钟回送； 外部有源晶振悬空连接	core
3	RTC_CKI	I		RTC 时钟晶体输入	RTC
4	RTC_CKO	O		RTC 时钟晶体回送	RTC
5	TEST_CFG_MODEN	I	PU	测试模块	core
6	SYS_RSTN	I	PU	系统复位	core

## 2.4 LCD 引脚定义（20）

表 2-2 LCD 引脚定义

No.	信号名称	方向	上下拉	描述	电压域
1	LCD_CLK	O	PU	LCD 时钟	core
2	LCD_VSYNC	O	PU	LCD 列同步	core
3	LCD_HSYNC	O	PU	LCD 行同步	core
4	LCD_EN	O	PU	LCD 可视使能信号	core
5	LCD_DAT_B0	O	PU	LCD 蓝色数据信号 0	core
6	LCD_DAT_B1	O	PU	LCD 蓝色数据信号 1	core
7	LCD_DAT_B2	O	PU	LCD 蓝色数据信号 2	core
8	LCD_DAT_B3	O	PU	LCD 蓝色数据信号 3	core
9	LCD_DAT_B4	O	PU	LCD 蓝色数据信号 4	core
10	LCD_DAT_G0	O	PU	LCD 绿色数据信号 0	core
11	LCD_DAT_G1	O	PU	LCD 绿色数据信号 1	core
12	LCD_DAT_G2	O	PU	LCD 绿色数据信号 2	core
13	LCD_DAT_G3	O	PU	LCD 绿色数据信号 3	core
14	LCD_DAT_G4	O	PU	LCD 绿色数据信号 4	core
15	LCD_DAT_G5	O	PU	LCD 绿色数据信号 5	core
16	LCD_DAT_R0	O	PU	LCD 红色数据信号 0	core
17	LCD_DAT_R1	O	PU	LCD 红色数据信号 1	core
18	LCD_DAT_R2	O	PU	LCD 红色数据信号 2	core
19	LCD_DAT_R3	O	PU	LCD 红色数据信号 3	core
20	LCD_DAT_R4	O	PU	LCD 红色数据信号 4	core

## 2.5 PLL 引脚定义 (4)

表 2-3 PLL 引脚定义

No.	信号名称	方向	上下拉	描述	电压域
1	PLL_DVDD12	I		1.2 伏数字电源	
2	PLL_DVSS12	I		1.2 伏数字地	
3	PLL_AVDD33	I		3.3 伏模拟电源	
4	PLL_AVSS33	I		3.3 伏模拟地	

## 2.6 VR 引脚定义 (6)

表 2-4 VR 引脚定义

No.	信号名称	方向	上下拉	描述	电压域
1	VR_VDDD-0	I		电源	VR_VDDD-0
2	VR_VDDA-0	I		电源	VR_VDDA-0
3	VR_VDDA-1	I		电源	VR_VDDA-1
4	VR_VDDD-1	I		电源	VR_VDDD-1
5	VR_VOUT	O		外接 10nf 电容	
6	VR_TOCAP	O		外接 4.7uf 电容	

## 2.7 DDR2 引脚定义 (71)

表 2-5 DDR 引脚定义

No.	信号名称	方向	上下拉	描述	电压域
1	DDR2_DQ00	B		外部存储数据总线第 0 位	
2	DDR2_DQ01	B		外部存储数据总线第 1 位	
3	DDR2_DQ02	B		外部存储数据总线第 2 位	
4	DDR2_DQ03	B		外部存储数据总线第 3 位	
5	DDR2_DQ04	B		外部存储数据总线第 4 位	
6	DDR2_DQ05	B		外部存储数据总线第 5 位	
7	DDR2_DQ06	B		外部存储数据总线第 6 位	
8	DDR2_DQ07	B		外部存储数据总线第 7 位	
9	DDR2_DQ08	B		外部存储数据总线第 8 位	
10	DDR2_DQ09	B		外部存储数据总线第 9 位	
11	DDR2_DQ10	B		外部存储数据总线第 10 位	
12	DDR2_DQ11	B		外部存储数据总线第 11 位	
13	DDR2_DQ12	B		外部存储数据总线第 12 位	
14	DDR2_DQ13	B		外部存储数据总线第 13 位	
15	DDR2_DQ14	B		外部存储数据总线第 14 位	
16	DDR2_DQ15	B		外部存储数据总线第 15 位	
17	DDR2_DQ16	B		外部存储数据总线第 16 位	
18	DDR2_DQ17	B		外部存储数据总线第 17 位	
19	DDR2_DQ18	B		外部存储数据总线第 18 位	
20	DDR2_DQ19	B		外部存储数据总线第 19 位	
21	DDR2_DQ20	B		外部存储数据总线第 20 位	
22	DDR2_DQ21	B		外部存储数据总线第 21 位	
23	DDR2_DQ22	B		外部存储数据总线第 22 位	

24	DDR2_DQ23	B		外部存储数据总线第 23 位	
25	DDR2_DQ24	B		外部存储数据总线第 24 位	
26	DDR2_DQ25	B		外部存储数据总线第 25 位	
27	DDR2_DQ26	B		外部存储数据总线第 26 位	
28	DDR2_DQ27	B		外部存储数据总线第 27 位	
29	DDR2_DQ28	B		外部存储数据总线第 28 位	
30	DDR2_DQ29	B		外部存储数据总线第 29 位	
31	DDR2_DQ30	B		外部存储数据总线第 30 位	
32	DDR2_DQ31	B		外部存储数据总线第 31 位	
33	DDR2_A00	O		外部存储地址总线第 0 位	
34	DDR2_A01	O		外部存储地址总线第 1 位	
35	DDR2_A02	O		外部存储地址总线第 2 位	
36	DDR2_A03	O		外部存储地址总线第 3 位	
37	DDR2_A04	O		外部存储地址总线第 4 位	
38	DDR2_A05	O		外部存储地址总线第 5 位	
39	DDR2_A06	O		外部存储地址总线第 6 位	
40	DDR2_A07	O		外部存储地址总线第 7 位	
41	DDR2_A08	O		外部存储地址总线第 8 位	
42	DDR2_A09	O		外部存储地址总线第 9 位	
43	DDR2_A10	O		外部存储地址总线第 10 位	
44	DDR2_A11	O		外部存储地址总线第 11 位	
45	DDR2_A12	O		外部存储地址总线第 12 位	
46	DDR2_A13	O		外部存储地址总线第 13 位	
47	DDR2_A14	O		外部存储地址总线第 14 位	
48	DDR2_DQS0	B		输入输出数据 strobe 信号	
49	DDR2_DQS1	B		输入输出数据 strobe 信号	
50	DDR2_DQS2	B		输入输出数据 strobe 信号	
51	DDR2_DQS3	B		输入输出数据 strobe 信号	
52	DDR2_DQM0	O		写数据屏蔽信号	
53	DDR2_DQM1	O		写数据屏蔽信号	
54	DDR2_DQM2	O		写数据屏蔽信号	
55	DDR2_DQM3	O		写数据屏蔽信号	
56	DDR2_CKp0	O		时钟信号	
57	DDR2_CKn0	O		时钟信号	
58	DDR2_CKE0	O		时钟有限信号	
59	DDR2_ODT0	O		ODT 信号	
60	DDR2_SCSn0	O		片选信号	
61	DDR2_BA0	O		bank 选择信号	
62	DDR2_BA1	O		bank 选择信号	
63	DDR2_BA2	O		bank 选择信号	
64	DDR2_RASn	O		行选择	
65	DDR2_CASn	O		列选择	
66	DDR2_WEn	O		写信号	
67	DDR2_GATEI0	I		GATE 信号, 与 GATEO0 通过 0Ω 电阻环回, 建议实现就近环回和到内存条远端环回两种选择	
68	DDR2_GATEI1	I		GATE 信号, 与 GATEO1 通过 0Ω 电阻环回, 建议实现就近环回和到内存条远端环回两种选择	



69	DDR2_GATE00	O		GATE 信号	
70	DDR2_GATE01	O		GATE 信号	
71	VREF_0V9	I		0.9V 参考电压	

## 2.8 USB 引脚定义(11)

表 2-6 USB 引脚定义

No.	信号名称	方向	上下拉	描述	电压域
1	USB_AVDD33	I		3.3 伏模拟电源	
2	USB_AVSS33	I		模拟地	
3	USB0_REXT	I		外部参考电阻 44.2ohm 对地	
4	USB_VSS33	I		模拟地	
5	USB_VDD33	I		3.3 伏模拟电源	
6	USB_DVDD	I		1.2 伏数字电源	
7	USB_DVSS	I		数字地	
8	USB0_XI	I		接地（或者接 12M 晶体）	
9	USB0_XO	I		外部 12Mhz 参考时钟输入 （或者接 12M 晶体）	
10	USB0_DP	B		USB 差分数据	
11	USB0_DM	B		USB 差分数据	

## 2.9 EJTAG 引脚定义(6)

表 2-7 JTAG 引脚定义

No.	信号名称	方向	上下拉	描述	电压域
1	EJTAG_TCK	I	PU	TAP 时钟(内置上拉)	core
2	EJTAG_TRST	I	PU	TAP 复位(内置上拉)	core
3	EJTAG_TDI	I	PU	TAP 数据输入(内置上拉)	core
4	EJTAG_TDO	O		TAP 数据输出	core
5	EJTAG_TMS	I	PU	TAP 工作模式(内置上拉)	core
6	TEST_JTAG_SEL	I	PU	JTAG/EJTAG 选择 0: JTAG 1: EJTAG	core

## 2.10 GMAC0 引脚定义(15)

表 2-8 GMAC 引脚定义

No.	信号名称	方向	上下拉	描述	电压域
1	GMAC0_TCKI	I		GMAC 传输时钟输入	core
2	GMAC0_TCKO	O		GMAC 传输时钟输出	core
3	GMAC0_TX0	O		GMAC 传输数据输出 0	core
4	GMAC0_TX1	O		GMAC 传输数据输出 1	core
5	GMAC0_TX2	O		GMAC 传输数据输出 2	core
6	GMAC0_TX3	O		GMAC 传输数据输出 3	core
7	GMAC0_TCTL	O		GMAC 传输控制	core
8	GMAC0_RCKI	I		GMAC 接收时钟输入	core
9	GMAC0_RX0	I		GMAC 接收数据输入 0	core
10	GMAC0_RX1	I		GMAC 接收数据输入 1	core
11	GMAC0_RX2	I		GMAC 接收数据输入 2	core
12	GMAC0_RX3	I		GMAC 接收数据输入 3	core

13	GMAC0_RCTL	I		GMAC 接受控制	core
14	GMAC0_MDC	O		读写 PHY 的时钟信号	core
15	GMAC0_MDIO	B		读写 PHY 的数据信号	core

注：GMAC 引脚电压为 3.3V

## 2.11 GMAC1 引脚定义(4)

表 2-9 GMAC 引脚定义

No.	信号名称	方向	上下拉	描述	电压域
1	GMAC1_TX_CLK_I	I		GMAC1 传输时钟输入	core
2	GMAC1_TX_CLK_O	O		GMAC1 传输时钟输出	core
3	GMAC1_RX_CLK_I	I		GMAC1 接收时钟输入	core
4	GMAC1_TX_CTL_O	O		GMAC1 传输控制	core

## 2.12 AC97 引脚定义(5)

表 2-10 AC97 引脚定义

No.	信号名称	方向	上下拉	描述	电压域
1	AC97_BIT_CLK	I		AC97 时钟输入	core
2	AC97_DATA_I	I	PU	AC97 数据输入	core
3	AC97_DATA_O	O	PU	AC97 数据输出	core
4	AC97_SYNC	O	PU	AC97 同步信号	core
5	AC97_RESET	O	PU	AC97 复位信号	core

## 2.13 SPI 引脚定义(7)

表 2-11 SPI 引脚定义

No.	信号名称	方向	上下拉	描述	电压域
1	SPI0_CLK	O		SPI0 时钟	core
2	SPI0_MISO	I		SPI0 主入从出数据	core
3	SPI0_MOSI	O		SPI0 主出从入数据	core
4	SPI0_CS0	O		SPI0 选通信号 0	core
5	SPI0_CS1	O	PU	SPI0 选通信号 1	core
6	SPI0_CS2	O	PU	SPI0 选通信号 2	core
7	SPI0_CS3	O	PU	SPI0 选通信号 3	core

## 2.14 UART 引脚定义(20)

表 2-12 UART 引脚定义

No.	信号名称	方向	上下拉	频率	描述	电压域
1	UART0_RX	I	PU	1MHz	UART0 发送数据	core
2	UART0_TX	O	PU	1MHz	UART0 接收数据	core
3	UART0_RTS	O	PU	1MHz	UART0 请求发送	core

4	UART0_CTS	I	PU	1MHz	UART0 允许发送	core
5	UART0_DSR	I	PU	1MHz	UART0 设备准备好	core
6	UART0_DTR	O	PU	1MHz	UART0 终端准备好	core
7	UART0_DCD	I	PU	1MHz	UART0 载波检测	core
8	UART0_RI	I	PU	1MHz	UART0 振铃提示	core
9	UART1_TX	O	PU	1MHz	UART1 发送数据	core
10	UART1_RX	I	PU	1MHz	UART1 接收数据	core
11	UART1_RTS	O	PU	1MHz	UART1 请求发送	core
12	UART1_CTS	I	PU	1MHz	UART1 允许发送	core
13	UART2_TX	O	PU	1MHz	UART2 发送数据	core
14	UART2_RX	I	PU	1MHz	UART2 接收数据	core
15	UART3_TX	O	PU	1MHz	UART3 发送数据	core
16	UART3_RX	I	PU	1MHz	UART3 接收数据	core
17	UART4_TX	O	PU	1MHz	UART4 发送数据	core
18	UART4_RX	I	PU	1MHz	UART4 接收数据	core
19	UART5_TX	O	PU	1MHz	UART5 发送数据	core
20	UART5_RX	I	PU	1MHz	UART5 接收数据	core

## 2.15 I2C 引脚定义(2)

表 2-13 I2C 引脚定义

No.	信号名称	方向	上下拉	描述	电压域
1	I2C_SCL	O		第一路 I2C 时钟	core
2	I2C_SDA	B		第一路 I2C 数据	core

## 2.16 CAN 引脚定义(4)

表 2-14 CAN 引脚定义

No.	信号名称	方向	上下拉	描述	电压域
1	CAN0_RX	I	PU	CAN0 数据输入	core
2	CAN0_TX	O	PU	CAN0 数据输出	core
3	CAN1_RX	I	PU	CAN1 数据输入	core
4	CAN1_TX	O	PU	CAN1 数据输出	core

## 2.17 NAND 引脚定义(14)

表 2-15 NAND 引脚定义

No.	信号名称	方向	上下拉	描述	电压域
1	NAND_CLE	O	PD	NAND 命令锁存	core
2	NAND_ALE	O	PD	NAND 地址锁存	core
3	NAND_RD	O	PD	NAND 读信号	core
4	NAND_WR	O	PD	NAND 写信号	core
5	NAND_CE	O	PD	NAND 片选信号	core

6	NAND_RDY	I	PD	NAND 忙信号	core
7	NAND_D0	O	PD	NAND 数据信号 0	core
8	NAND_D1	O	PD	NAND 数据信号 1	core
9	NAND_D2	O	PD	NAND 数据信号 2	core
10	NAND_D3	O	PD	NAND 数据信号 3	core
11	NAND_D4	O	PD	NAND 数据信号 4	core
12	NAND_D5	O	PD	NAND 数据信号 5	core
13	NAND_D6	O	PD	NAND 数据信号 6	core
14	NAND_D7	O	PD	NAND 数据信号 7	core

## 2.18 PWM 引脚定义(4)

表 2-18 PWM 引脚定义

No.	信号名称	方向	上下拉	描述	电压域
1	PWM0	O	PU	PWM0 波形输出	core
2	PWM1	O	PU	PWM1 波形输出	core
3	PWM2	O	PU	PWM2 波形输出	core
4	PWM3	O	PU	PWM3 波形输出	core

## 2.19 电源/地引脚(58)

表 2-19 电源地引脚

No.	信号名称	方向	描述	电压值	电压域	数目
1	VDD1V2		CORE 域	1.2v	CORE	10
2	VDD1V8		DDR2 电压域	1.8v	DDR2	8
3	VDD3V3		PAD 电压域	3.3v	PAD	11
4	RTC_VDD33		RTC 电源	3.3v	RTC	2
5	VSS		接地	0v	接地	27
6	RTC_VSS33		RTC 地	0v	RTC	2

### 3 地址空间分配

本章给出 1B 芯片各模块的地址空间分配。

#### 3.1 一级 AXI 交叉开关上模块的地址空间

表 3-1 AXI 各模块地址分配

地址空间	模块	说明
0x0000,0000 – 0x0fff,ffff	DDR	256MB
0x1000,0000 – 0x1c19,ffff		<b>RESERVED</b>
0x1c20,0000 – 0x1c2f,ffff	DC Slave	1MB
0x1c30,0000 – 0x1eff,ffff		RESERVED
0x1f00,0000 – 0x1fff,ffff	AXI MUX Slave	16MB
0x2000,0000 – 0x7fff,ffff		RESERVED

#### 3.2 AXI MUX 下各模块的地址空间

表 3-2 AXI MUX 各模块地址分配

地址空间	模块	说明
0xbf00,0000 – 0xbf7f,ffff	SPI0-memory	8MB
0xbf80,0000 – 0xbfbf,ffff	SPI1-memory	4MB
0xbfc0,0000 – 0xbfcf,ffff	SPI0	1MB
0xbfd0,0000 – 0xbfdf,ffff	CONFREG	1MB
0xbfe0,0000 – 0xbfe0,ffff	USB	64KB
0xbfe1,0000 – 0xbfe1,ffff	GMAC0	64KB
0xbfe2,0000 – 0xbfe2,ffff	GMAC1	64KB
0xbfe3,0000 – 0xbfe3,ffff		<b>RESERVED</b>
0xbfe4,0000 – 0xbfe7,ffff	APB-devices	256KB
0xbfe8,0000 – 0xbfeb,ffff	SPI0-IO	256KB
0xbfec,0000 – 0xbfef,ffff	SPI1-IO	256KB
0xbff0,0000 – 0xbfff,ffff		<b>RESERVED</b>

#### 3.3 APB 各模块的地址空间分配

表 3-3 APB 各模块地址分配

地址空间	模块	说明
0xbfe40000-0xbfe43fff	UART0	16KB
0xbfe44000-0xbfe47fff	UART1	16KB
0xbfe48000-0xbfe4bfff	UART2	16KB
0xbfe4c000-0xbfe4ffff	UART3	16KB

0xbfe50000-0xbfe53fff	CAN0	16KB
0xbfe54000-0xbfe57fff	CAN1	16KB
0xbfe58000-0xbfe5bfff	I2C-0	16KB
0xbfe5c000-0xbfe5ffff	PWM	16KB
0xbfe60000-0xbfe63fff		<b>RESERVED</b>
0xbfe64000-0xbfe67fff	RTC	16KB
0xbfe68000-0xbfe6bfff	I2C-1	16KB
0xbfe6c000-0xbfe6ffff	UART4	16KB
0xbfe70000-0xbfe73fff	I2C-2	16KB
0xbfe74000-0xbfe77fff	AC97	16KB
0xbfe78000-0xbfe7bfff	NAND	16KB
0xbfe7c000-0xbfe7ffff	UART5	16KB

## 4 DDR2

龙芯 1B 集成了内存控制器，兼容 DDR2 SDRAM 标准（JESD79-2B）。龙芯 1B 提供 JESD79-2B 兼容的内存读写操作。

### 4.1 DDR2 SDRAM 控制器特性

龙芯 1B 通过一个片选信号和 18 位的地址总线(15 位行/列地址和 3 位逻辑 Bank 地址)实现最大地址空间是 32G( $2^{35}$ )。

龙芯 1B 支持所有的与 JESD79-2B 兼容的内存颗粒。DDR2 控制器参数能被设置为支持指定的内存芯片类型。芯片选择信号（CS<sub>n</sub>）的最大数目是 1。行地址（RAS<sub>n</sub>）和列地址（CAS<sub>n</sub>）的最大带宽分别是 15 和 14。还有 3 位的逻辑 bank 信号（BANK<sub>n</sub>）。

CPU 内存的物理地址能被转换位行/列地址，见表 5-1。例如，1 个 CS<sub>n</sub> 信号，8 个 banks，12 位行地址和 12 位列地址。

表 5-1 DDR2 SDRAM 行/列地址转换

34	30 29	18 17	15 14	3 2	0
	RAS <sub>n</sub>	BANK <sub>n</sub>	CAS <sub>n</sub>	Byte	

内存控制器接收从处理器或外部设备发送的内存读写请求。无论是读还是写操作，内存控制器都处在 slave 状态。

内存控制器中实现了动态页管理功能。对于内存的一次存取，不需软件设计者的干预，控制器会在硬件电路上选择 Open Page/Close Page 策略。内存控制器特性包括：

- 全流水的命令和数据读写；
- 通过合并和重排序增加带宽；
- 通过丰富的寄存器读写端口修改基本的参数；
- 内置 Delay Compensation Circuit(DCC)，用来可靠的发送/接收数据；
- 1 位和 2 位错误检测，通过 ECC 进行 1 位的错误修正；
- 频率：133MHz-333MHz；

### 4.2 DDR2 SDRAM 读协议

图 5-1 中显示 DDR2 SDRAM 读协议，命令（CMD）包括 RAS<sub>n</sub>，CAS<sub>n</sub> 和 WE<sub>n</sub>。当一个读请求发生时，RAS<sub>n</sub>=1，CAS<sub>n</sub>=0，WE<sub>n</sub>=1。

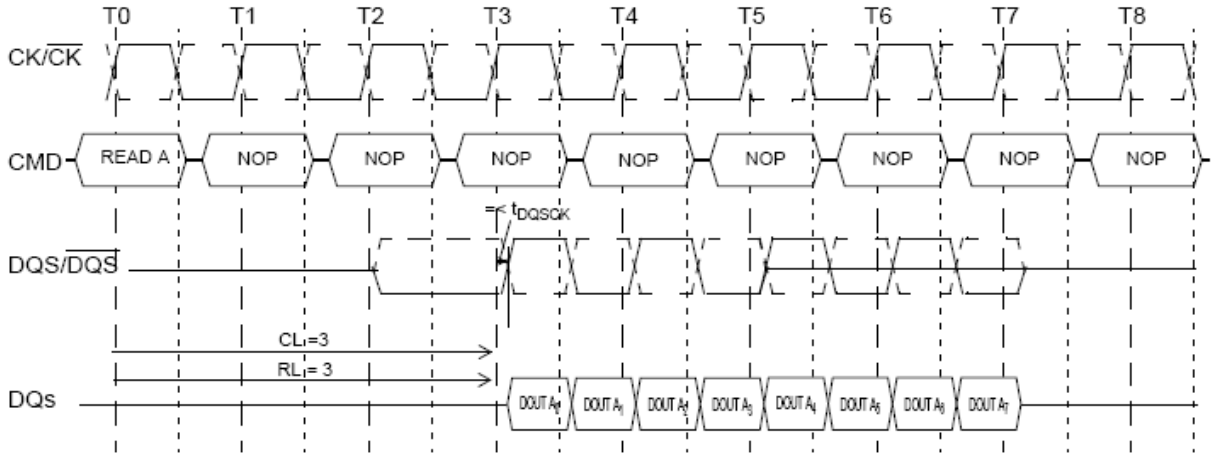


图 5-1 DDR2 SDRAM 读协议， Cas Latency = 3, Read Latency = 3, Burst Length = 8

### 4.3 DDR2 SDRAM 写协议

在图 5-2 中显示 DDR2 SDRAM 写协议，命令 (CMD) 包括 RAS<sub>n</sub>, CAS<sub>n</sub> 和 WE<sub>n</sub>。当写请求发生时，RAS<sub>n</sub>=1, CAS<sub>n</sub>=0, WE<sub>n</sub>=0。与读协议不同，DQM 用来识别需要被写的字节数。DQM 和 DQS 是同步的。

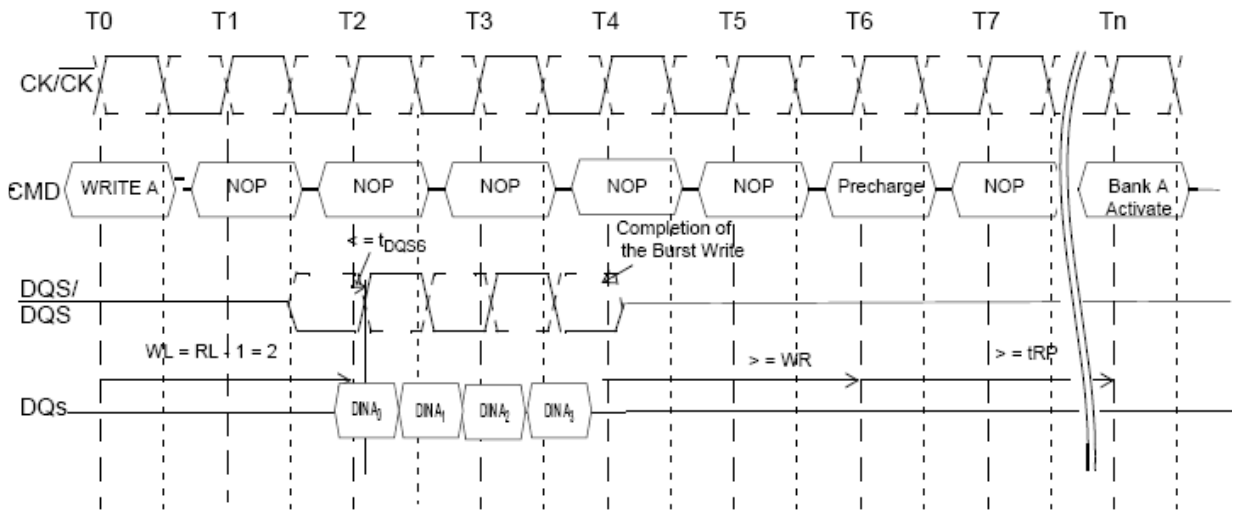


图 5-2 DDR2 SDRAM 写协议， Cas Latency = 3, Write Latency = Read Latency - 1 = 2, Burst Length = 4.

### 4.4 DDR2 SDRAM 参数设置顺序

为了在系统中支持不同的 DDR2 SDRAM 颗粒，DDR2 SDRAM 需要在加电复位后配置。JESD79-2B 标准定义了详细的配置操作和过程。DDR2 在内存初始化前是不可用的，内存初始化顺序为：

1. 系统复位期间，aresetn 信号设置为 0，所有的寄存器内容将清除为缺省值；
2. 系统复位释放，aresetn 信号设置为 1；



3. 向配置寄存器地址发 32 位写指令，配置所有 29 个配置寄存器。此时如果写 CTRL\_03，应将其中参数 START 设为 0。所有寄存器都必须正确配置才可以正常工作。
4. 向配置寄存器 CTRL\_03 中发 32 位写指令。此时应将参数 START 设为 1。结束后内存控制器将自动对内存发起初始化指令。

在系统主板初始化后，DDR2 SDRAM 控制器在内存使用前需要配置内存类型。特别的是需要将相应的配置参数写到对应于物理地址 0X0FFF FE00 的 29 个 64 位寄存器中。每个寄存器会包括一个、多个或部分的参数。

## 4.5 DDR2 SDRAM 采样模式配置

在龙芯 1B 的 DDR2 SDRAM 控制器中，并通过延迟补偿电路（使用 DLL）来采样返回 DQS 的数据。因为内存控制器和 SDRAM 模块间的数据返回路径有延迟，所以必须引进一组控制信号用来测量延迟。

DDR2\_GATE\_I[1:0] 和 DDR2\_GATE\_O[1:0] 的控制信号用于延迟测量。在 PCB 设计中，DDR2\_GATE\_I 和 DDR2\_GATE\_O 连接起来模拟 PCB 上的写延迟。这样，采样的精确性能够被保证。

## 4.6 DDR2 SDRAM PAD 驱动配置

表 4-2 DDR2 SDRAM PAD 驱动控制

寄存器地址	位	PAD 控制位	对应被控制的 PAD
0xbf010c8	[27:26]	DDR2_ssel[1:0]	DDR-CK/CONTRL/ADDR
	[29:28]	DDR2_ssel[3:2]	DQ[31:0]
	[31:30]	DDR2_ssel[5:4]	DQS[3:0]
0xbf010F8	[27:26]	DDR2_tsel[1:0]	DQ[31:0]
	[29:28]	DDR2_tsel[3:2]	DQS[3:0]
	[30]	DDR2_st[0]	All except CK

## 4.7 DDR2 16 位工作模式配置

DDR2 支持 16 位数据宽度的外部接口，配置工作如下：

- 1) 初始化 DDR 控制器，使用与 32 位模式相同的参数；
- 2) DISABLE\_DDR\_CONFSPACE 位置 1，关闭 DDR 控制器配置空间；
- 3) DDR32TO16EN 位置 1，使能 DDR16 位模式；
- 4) DDR2 16 位数据宽度模式正常使用。

表 4-3 DDR2 SDRAM 16 位数据宽度配置

寄存器地址	位	配置描述
0xBFD0_0424	[20]	DISABLE_DDR_CONFSPACE
0xBFD0_0424	[16]	DDR32TO16EN

## 5 LCD

本章给出 1B 芯片内 LCD 控制器 (Display Controller) 的详细描述和配置。Display Controller 作为一个整体的模块, 该模块读取指针数据和图像数据, 通过对这些数据进行格式转换、颜色抖动、gamma 调整等步骤产生最终的数据输出, 同时为两个显示处理单元产生同步信号和数据使能信号, 最后将最终处理后的图像数据和同步信号发往显示接口。

### 5.1 特性

- 支持格式转换
- 最大显示支持到 1920×1080@60Hz
- 同步信号可编程
- Gamma 调整查找表
- VBLANK 同步

#### 5.1.1 数据格式

Display Controller 支持以下数据格式:

R4G4B4           -> 12 bits per pixel  
 R5G5B5           -> 15 bits per pixel  
 R5G6B5           -> 16 bits per pixel  
 R8G8B8           -> 24 bits per pixel

### 5.2 寄存器

寄存器列表:

寄存器名	配置地址
Frame Buffer configuration	{15'h92,5'h0}
Frame Buffer Address_0	{15'h93,5'h0}
Frame Buffer Address_1	{15'hAC,5'h0}
Frame Buffer Stride	{15'h94,5'h0}
Frame Buffer Origin	{15'h95,5'h0}
Display Ditheronfiguration	{15'h9B,5'h0}
Display Dither Table(low)	{15'h9C,5'h0}
Display Dither Table(high)	{15'h9D,5'h0}
Pane Configuration	{15'h9E,5'h0}
Panel Timing	{15'h9F,5'h0}
HDisplay	{15'hA0,5'h0}

Hsync	{15'hA1,5'h0}
VDisplay	{15'hA4,5'h0}
VSync	{15'hA5,5'h0}
Cursor Configuration	{16'h152,4'h0}
Cursor Address	{16'h153,4'h0}
Cursor Location	{16'h154,4'h0}
Cursor Background	{16'h155,4'h0}
Cursor Foreground	{16'h156,4'h0}
Gamma Index	{15'hA7,5'h0}
Gamma Data	{15'hA8,5'h0}

Frame Buffer Configuration	bit	描述	初始值
Reset	20	写 0 reset	0
Gamma enable	12	写 1 使能	0
Switch Panel	9	置 1 时，表示该显示单元的输出使用另外一个显示单元的输出，即如果对 0 号显示单元配置该位时表示 0 号单元的输出和输出控制信号复制于 1 号显示单元的输出，同理如果对 1 号显示单元配置该位表示 1 号显示单元的输出和输出控制信号复制于 0 号显示单元。	0
Output enable	8	写 1 使能输出，写 0 则不输出显示数据	0
Format	[2:0]	0 none 1 R4G4B4 2 R5G5B5 3 R5G6B5 4 R8G8B8	0

Frame buffer address_0	Bit	描述	初始值
Frame buffer address_0	[31:0]	内存中图像数据首地址	32'h0000_0000

Frame buffer address_1	Bit	描述	初始值
Frame buffer address_1	[31:0]	对于需要支持双 frame buffer 显示的情况，此时该寄存器可配置第二块 Frame Buffer 的地址，DC 运行时第一帧先从 Frame Buffer_0 取数据，第二帧从 Frame Buffer_1 取数据，第三帧再从 Frame Buffer_0 取数据，依此循环。对于不需要双 frame buffer 的情况，可将此 Frame Buffer Address_1 配置成和 Frame Buffer_0 一样的地址即可	32'h0000_0000

Frame buffer stride	Bit	描述	初始值
---------------------	-----	----	-----

Frame buffer stride	[31:0]	显示屏一行的字节数	32'h0000_0000
---------------------	--------	-----------	---------------

Frame buffer origin	Bit	描述	初始值
Frame buffer origin	[31:0]	显示屏左侧原有字节数，一般配 0 即可	32'h0000_0000

Display Dither Configuration	Bit	描述	初始值
Enable	31	置 1 使能 dither 功能	0
RedSize	[19:16]	红色域宽度	4'b0000
GreenSize	[11:8]	绿色域宽度	4'b0000
BlueSize	[3:0]	蓝色域宽度	4'b0000

Display Dither Table	Bit	描述	初始值
Display Dither Table	[63:0]	该寄存器有 64 位，而 Display Controller 的寄存器都是 32 位宽，所以实际上该寄存器为两个 32 位的寄存器。分为 Display Dither Table(low)和 Display Dither Table(high)。这两个寄存器以像素点的 X 和 Y 坐标为索引，配置作为比较的数值。凡进入 Dither 处理模块的图像数据都会在 Display Dither Table 寄存器中被相应的索引到一个比较值，若输入数据的值的后四位大于该比较值则进行颜色增强。	64'h0000_0000
Display Dither Table (low)	[31:0]		
Y0_X0	[3:0]	坐标 (0, 0) 处的比较值	4'b0000
Y0_X1	[7:4]	坐标 (1, 0) 处的比较值	4'b0000
Y0_X2	[11:8]	坐标 (2, 0) 处的比较值	4'b0000
Y0_X3	[15:12]	坐标 (3, 0) 处的比较值	4'b0000
Y1_X0	[19:16]	坐标 (0, 1) 处的比较值	4'b0000
Y1_X1	[23:20]	坐标 (1, 1) 处的比较值	4'b0000
Y1_X2	[27:24]	坐标 (2, 1) 处的比较值	4'b0000
Y1_X3	[31:28]	坐标 (3, 1) 处的比较值	4'b0000
Display Dither Table (high)	[31:0]		
Y2_X0	[3:0]	坐标 (0, 2) 处的比较值	4'b0000
Y2_X1	[7:4]	坐标 (1, 2) 处的比较值	4'b0000
Y2_X2	[11:8]	坐标 (2, 2) 处的比较值	4'b0000
Y2_X3	[15:12]	坐标 (3, 2) 处的比较值	4'b0000
Y3_X0	[19:16]	坐标 (0, 3) 处的比较值	4'b0000
Y3_X1	[23:20]	坐标 (1, 3) 处的比较值	4'b0000

Y3_X2	[27:24]	坐标 (2, 3) 处的比较值	4'b0000
Y3_X3	[31:28]	坐标 (3, 3) 处的比较值	4'b0000

Panel configuration	Bit	描述	初始值
ClockPolarity	9	时钟极性, 置 1 将时钟反向	0
Clock	8	时钟使能, 置 1 使能时钟	1
DE_Polarity	1	数据使能极性, 置 1 取反, 一般设 0	0
DE	0	数据使能, 置 1 使能数据输出	1

HDisplay	Bit	描述	初始值
Total	[27:16]	显示屏一行的总体像素数 (包括非显示区)	12'b0
DisplayEnd	[11:0]	显示屏一行中显示区的像素数	12'b0

HSync	Bit	描述	初始值
Polarity	31	HSync 信号的极性, 置 1 取反, 一般设 0	0
Pulse	30	HSync 信号使能, 置 1 只能 HSync 信号输出	1
End	[27:16]	HSync 信号结束的像素数	12'b0
Start	[11:0]	HSync 信号开始的像素数	12'b0

VDisplay	Bit	描述	初始值
Total	[26:16]	显示屏总体的行数 (包括消隐区)	11'b0
DisplayEnd	[10:0]	显示屏中显示区的行数	11'b0

VSync	Bit	描述	初始值
Polarity	31	VSync 信号的极性, 置 1 取反, 一般设 0	0
Pulse	30	VSync 信号使能, 置 1 只能 VSync 信号输出	1
End	[27:16]	VSync 信号结束的行数	12'b0
Start	[11:0]	VSync 信号开始的行数	12'b0

Cursor Configuration	Bit	描述	初始值
HotSpotX	[20:16]	指针的“焦点”(作用点)的横坐标(在指针 32*32 的图案中的横坐标)	5'b0
HotSpotY	[12:8]	指针的“焦点”(作用点)的纵坐标(在指针 32*32 的图案中的横坐标)	5'b0
Display	4	指示指针存在于哪个显示单元中, 0 表示在 0 号显示单元中, 1 表示指针在 1 号显示单元中	0
Format	[1:0]	0 disabled 1 masked 2 A8R8G8B8	2'b0

Cursor Address	Bit	描述	初始值
----------------	-----	----	-----

Cursor Address	[31: 0]	指针数据在内存中的基地址	32'b0
----------------	---------	--------------	-------

Cursor Location	Bit	描述	初始值
Y	[26:16]	指针的焦点在整个显示区的纵坐标	11'b0
X	[10:0]	指针的焦点在整个显示区的横坐标	11'b0
Cursor Background	Bit	描述	初始值
Red	[23:16]	指针单色模式下背景色的红色域	8'b0
Green	[15:8]	指针单色模式下背景色的绿色域	8'b0
Blue	[7:0]	指针单色模式下背景色的蓝色域	8'b0

Cursor Foreground	Bit	描述	初始值
Red	[23:16]	指针单色模式下前景色的红色域	8'b0
Green	[15:8]	指针单色模式下前景色的绿色域	8'b0
Blue	[7:0]	指针单色模式下前景色的蓝色域	8'b0

Gamma Index	Bit	描述	初始值
Index	[7:0]	表示从 0-255 颜色值之间的哪一项开始进行 Gamma 调整，一般设 0。只需配一次，此后该值硬件会自增。	8'b0

Gamma Data	Bit	描述	初始值
Red	[23:16]	Gamma 调整的红色域，将 Gamma Index 指示的值调整为当前域的值	8'b0
Green	[15:8]	Gamma 调整的绿色域，将 Gamma Index 指示的值调整为当前域的值	8'b0
Blue	[7:0]	Gamma 调整的蓝色域，将 Gamma Index 指示的值调整为当前域的值	8'b0

1B 芯片 LCD 正常工作缺省是 16 位模式，同时也支持 RGB444/555/565/888 位模式。芯片外部只有 16 个 LCD 数据 PAD，如果需要复用连接外部 24 位数据线的 LCD，内部数据的传输会变化。

当工作在 24 位模式，LCD\_BLUE0/LCD\_RED0 可以不使用（显示效果区别不明显，可以节省一个两线 UART，下表最后一列粉色部分）。

配置如下所示，

1B-PAD	RGB 444	RGB 555	RGB 565	RGB 888
LCD_DAT_B0	Nouse, USE as GPIO	LCD_BLUE0	LCD_BLUE0	LCD_BLUE3
LCD_DAT_B1	LCD_BLUE0	LCD_BLUE1	LCD_BLUE1	LCD_BLUE4
LCD_DAT_B2	LCD_BLUE1	LCD_BLUE2	LCD_BLUE2	LCD_BLUE5
LCD_DAT_B3	LCD_BLUE2	LCD_BLUE3	LCD_BLUE3	LCD_BLUE6
LCD_DAT_B4	LCD_BLUE3	LCD_BLUE4	LCD_BLUE4	LCD_BLUE7
LCD_DAT_G0	Nouse, USE as GPIO		LCD_GREEN0	LCD_GREEN2
LCD_DAT_G1	Nouse, USE as GPIO	LCD_GREEN0	LCD_GREEN1	LCD_GREEN3

LCD_DAT_G2	LCD_GREEN0	LCD_GREEN1	LCD_GREEN2	LCD_GREEN4
LCD_DAT_G3	LCD_GREEN1	LCD_GREEN2	LCD_GREEN3	LCD_GREEN5
LCD_DAT_G4	LCD_GREEN2	LCD_GREEN3	LCD_GREEN4	LCD_GREEN6
LCD_DAT_G5	LCD_GREEN3	LCD_GREEN4	LCD_GREEN5	LCD_GREEN7
LCD_DAT_R0	Nouse, USE as GPIO	LCD_RED0	LCD_RED0	LCD_RED3
LCD_DAT_R1	LCD_RED0	LCD_RED1	LCD_RED1	LCD_RED4
LCD_DAT_R2	LCD_RED1	LCD_RED2	LCD_RED2	LCD_RED5
LCD_DAT_R3	LCD_RED2	LCD_RED3	LCD_RED3	LCD_RED6
LCD_DAT_R4	LCD_RED3	LCD_RED4	LCD_RED4	LCD_RED7
URT0_RX	UART0 发送数据			LCD_BLUE0
URT0_TX	UART0 接收数据			LCD_RED0
URT0_RTS	UART0 请求发送			LCD_BLUE1
URT0_CTS	UART0 允许发送			LCD_BLUE2
URT0_DSR	UART0 数据设备准备好			LCD_GREEN0
URT0_DTR	UART0 数据终端准备好			LCD_GREEN1
URT0_DCD	UART0 载波检测			LCD_RED1
URT0_RI	UART0 振铃提示			LCD_RED2

## 6 GMAC0

**注意：千兆模式 RGMII 接口有不稳定现象，不建议使用。**

GMAC0 控制器可以通过配置成百兆模式(MII)或千兆模式(RGMII)。如果外部连接百兆 PHY，需要复用 PWM0, PWM1 和 GMAC0\_TX\_CLK\_O 三个 PAD，配置如下表：

PAD	MAC 信号	配置位	复位值
PWM0	MAC_0_COL	GMA0_USE_PWM01	1'b0
PWM1	MAC_0_CRD	GMA0_USE_PWM01	1'b0
GMAC0_TX_CLK_O	MAC_0_RX_ERR	GMA0_USE_TX_CLK	1'b1

寄存器地址：0XBFD0\_0424

配置位	Bit 位	描述
GMA0_USE_PWM01	0	1: 百兆模式 MAC_COL/MAC_CRD 分别复用 PWM0/1 0: 千兆模式
GMA0_USE_TX_CLK	2	1: Mii_0_RX_ERR 输入信号复用 GMAC0_TX_CLK_O 0: 千兆模式

- a) 百兆模式下，如果外部 PHY 不提供 RX\_ERR 信号，GMAC0\_TX\_CLK\_O 需接地，不能悬空；
- b) 百兆模式下，如果外部 PHY 提供 RX\_ERR 信号，GMAC0\_TX\_CLK\_O 与 RX\_ERR 连接。



## 7 GMAC1

### 7.1 配置成 MAC 的连接和复用方式

**注意：**千兆模式 RGMII 接口有不稳定现象，不建议使用。

GMAC1 控制器可以通过配置成百兆模式(MII)或千兆模式(RGMII)。如果外部连接百兆 PHY，需要复用 PWM2, PWM3 和 GMAC1\_TX\_CLK\_O 三个 PAD，配置如下表：

PAD	MAC 信号	配置位	复位值
PWM2	MAC_1_COL	GMA1_USE_PWM23	1'b0
PWM3	MAC_1_CRD	GMA1_USE_PWM23	1'b0
GMAC1_TX_CLK_O	MAC_1_RX_ERR	GMA1_USE_TX_CLK	1'b1

寄存器地址：0XBFD0\_0424

配置位	Bit 位	描述
GMA1_USE_PWM23	1	1: 百兆模式 MAC_COL/MAC_CRD 分别复用 PWM2/3 0: 千兆模式
GMA1_USE_TX_CLK	3	1: MII_1_RX_ERR 输入信号复用 GMAC1_TX_CLK_O 0: 千兆模式

- c) 百兆模式下，如果外部 PHY 不提供 RX\_ERR 信号，GMAC1\_TX\_CLK\_O 需接地，不能悬空；
- d) 百兆模式下，如果外部 PHY 提供 RX\_ERR 信号，GMAC1\_TX\_CLK\_O 与 RX\_ERR 连接。

### 7.2 GMAC1 外部信号复用和配置

在 1B 芯片里，GMAC1 的大多数信号通过复用来实现对外部的连接。

PAD	方向	描述	复用
GMAC1_MDCK	O	GMAC1 读写 PHY 的时钟信号	UART0_DCD
GMAC1_MDIO	I/O	GMAC1 读写 PHY 的数据信号	UART0_RI
GMAC1_RX_CTL	I	GMAC1 接收数据控制信号	UART0_RX
GMAC1_RX0	I	GMAC1 接收数据输入 0	UART0_TX
GMAC1_RX1	I	GMAC1 接收数据输入 1	UART0_RTS
GMAC1_RX2	I	GMAC1 接收数据输入 2	UART0_CTS
GMAC1_RX3	I	GMAC1 接收数据输入 3	UART0_DSR
GMAC1_TX0	O	GMAC1 传输数据输出 0	UART1_RX
GMAC1_TX1	O	GMAC1 传输数据输出 1	UART1_TX
GMAC1_TX2	O	GMAC1 传输数据输出 2	UART1_RTS
GMAC1_TX3	O	GMAC1 传输数据输出 3	UART1_CTS
GMAC1_TX_CLK_I	I	GMAC1 传输时钟输入	无复用
GMAC1_TX_CLK_O	O	GMAC1 传输时钟输出	
GMAC1_RX_CLK_I	I	GMAC1 接收时钟输入	
GMAC1_TX_CTL_O	O	GMAC1 输出数据控制信号	

从上表可以看出，GMAC1 的多数 PAD 通过复用 UART0 和 UART1 的信号实现。

这些复用的控制和配置如下：

GMAC1 复用 UART0 部分控制信号：GMAC1\_USE\_UART0；

GMAC1 复用 UART1 部分控制信号：GMAC1\_USE\_UART1；

参数名称	BIT 位	缺省值	描述
GPIO_MUX 基地址：0XBFD0_0420			
GMAC1_USE_UART0	3	0	UART0 的信号用来复用给 GMAC1 使用
GMAC1_USE_UART1	4	0	UART1 的信号用来复用给 GMAC1 使用

## 8 USB HOST

### 8.1 总体概述

1B 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

GPIO\_MUX\_CTRL1[31] (基地址 0XBFD0\_0424) 是 USB 复位使能信号, USB 工作前需要写 1

### 8.2 USB 主机控制器寄存器

#### 8.2.1 EHCI 相关寄存器

EHCI 的相关寄存器包括 Capability 寄存器、Operational 寄存器和, EHCI 实现相关寄存器。1B 的 USB 主机控制器兼容 EHCI Rev 1.0 协议, Capability 寄存器和 Operational 寄存器的详细信息参照 Enhanced Host Controller Interface Rev 1.0 Specification。

#### 8.2.2 Capability 寄存器

名称	地址	宽度	访问	说明
HCCAPBASE	0xbfe00000	32	RO	默认值为 32'h01000010
HCSPARAMS	0xbfe00004	32	RO	默认值为 32'h00001116
HCCPARAMS	0xbfe00008	32	RO	默认值为 32'h0000A010

(注: USBBase 固定为 EHCI slave 的起始地址 0xbfe00000)

#### 8.2.3 Operational 寄存器

名称	地址	宽度	访问	说明
USBCMD	0xbfe00010	32	R/W 、 RO	USB 主机控制器的命令寄存器
USBSTS	0xbfe00014	32	R/W 、 RO	USB 主机控制器的状态寄存器
USBINTR	0xbfe00018	32	R/W	USB 主机控制器的中断设置寄存器
FRINDEX	0xbfe0001c	32	R/W	USB 主机控制器的帧索引寄存器
CTRLDSSEGMENT	0xbfe00020	32	R/W	存放 EHCI 控制数据结构的地址
PERIODICLISTBASE	0xbfe00024	32	R/W	存放周期数据帧表的起始地址
ASYNCLISTADDR	0xbfe00028	32	R/W	存放下一个要执行的异步队

				列的起始地址
CONFIGFLAG	0xbfe00050	32	R/W	配置模式寄存器
PORTSC 1	0xbfe00054	32	R/W 、 RO	端口 1 状态和控制寄存器
PORTSC 2	0xbfe00058	32	R/W 、 RO	端口 2 状态和控制寄存器

(注: USBOPBase 固定为 EHCI slave 的起始地址+`h10)

## 8.3 OHCI 相关寄存器

OHCI 的相关寄存器包括 Operational 寄存器和 OHCI 实现相关寄存器。1B 的 USB 主机控制器兼容 OHCI Rev 1.0 协议, Operational 寄存器的详细信息参照 Open Host Controller Interface Rev 1.0 Specification。

### 8.3.1 Operational 寄存器

名称	地址	宽度	访问	说明
HcRevision	0xbfe08000	32	-	控制和状态
HcControl	0xbfe08004	32	-	
HcCommonStatus	0xbfe08008	32	-	
HcInterruptStatus	0xbfe0800C	32	-	
HcInterruptEnable	0xbfe08010	32	-	
HcInterruptDisable	0xbfe08014	32	--	
HcHCCA	0xbfe08018	32	-	内存指针
HcPeriodCuttentED	0xbfe0801C	32	-	
HcControlHeadED	0xbfe08020	32	-	
HcControlCurrentED	0xbfe08024	32	-	
HcBulkHeadED	0xbfe08028	32	-	
HcBulkCurrentED	0xbfe0802C	32	-	
HcDoneHead	0xbfe08030	32	-	
HcRmInterval	0xbfe08034	32	-	
HcFmRemaining	0xbfe08038	32	-	
HcFmNumber	0xbfe0803C	32	-	
HcPeriodicStart	0xbfe08040	32	-	
HcLSThreshold	0xbfe08044	32	-	根集线器
HcRhDescriptorA	0xbfe08048	32	-	

HcRhDescriptorB	0xbf0804C	32	-	
HcRhStatus	0xbf08050	32	-	
HcRhPortStatus1	0xbf08054	32	-	
HcRhPortStatus2	0xbf08058	32	-	

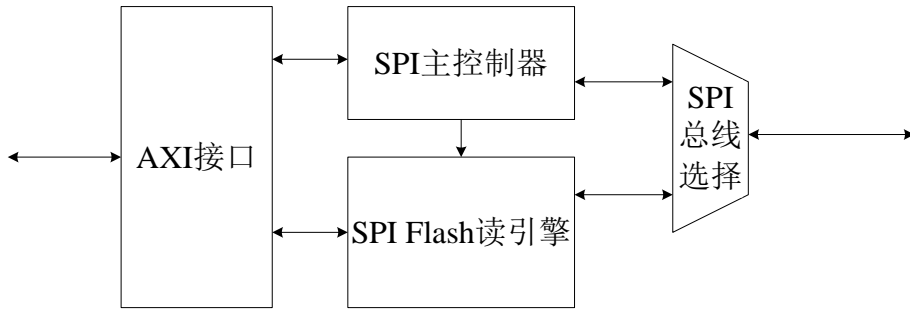
## 9 SPI0

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

1B 集成的 SPI 控制器仅可作为主控端，所连接的是从设备。其结构如下图所示，由一个 SPI 主控制器和 SPI Flash 读引擎组成。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 0xbfc00000，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。SPI0 的 IO 寄存器的基地址 0xbf80000，外部存储地址空间是 0xbf00,0000-0xbf7f,ffff 共 8MB。

本模块结构如下图所示，由 AXI 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，AXI 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)。



下图是 SPI 主控制器的结构，系统寄存器包括控制寄存器，状态寄存器和外部寄存器，分频器生成 SPI 总线工作的时钟信号，由于数据读、写缓冲器 (FIFO) 允许 SPI 同时进行串行发送和接收数据。

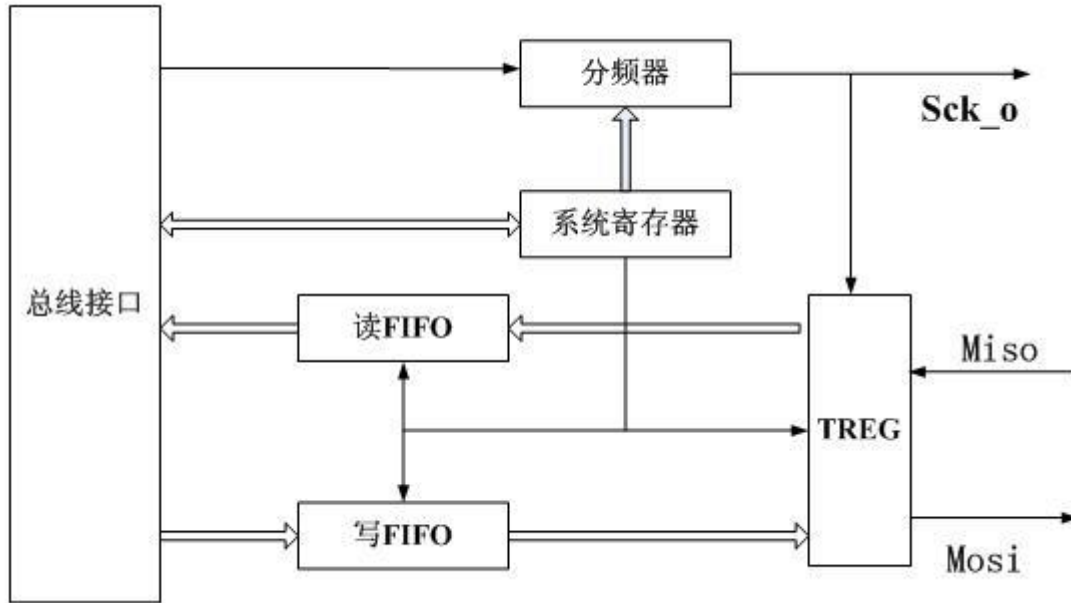


图 10-1 SPI 主控制器结构

## 9.2 SPI 控制器寄存器

### 9.2.1 控制寄存器（SPCR）

中文名： 控制寄存器  
寄存器位宽： [7: 0]  
偏移量： 0x00  
复位值： 0x10

位域	位域名称	位宽	访问	描述
7	Spie	1	RW	中断输出使能信号 高有效
6	spe	1	RW	系统工作使能信号高有效
5	Reserved	1	RW	保留
4	mstr	1	RW	master 模式选择位，此位一直保持 1
3	cpol	1	RW	时钟极性位
2	cpha	1	RW	时钟相位位 1 则相位相反，为 0 则相同
1:0	spr	2	RW	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）

中文名： 状态寄存器  
寄存器位宽： [7: 0]  
偏移量： 0x01  
复位值： 0x05

位域	位域名称	位宽	访问	描述
7	spif	1	RW	中断标志位 1 表示有中断申请，写 1 则清零
6	wcol	1	RW	写寄存器溢出标志位 为 1 表示已经溢出,写 1 则清零

5:4	Reserved	2	RW	保留
3	wffull	1	RW	写寄存器满标志 1 表示已经满
2	wfempty	1	RW	写寄存器空标志 1 表示空
1	rffull	1	RW	读寄存器满标志 1 表示已经满
0	rfempty	1	RW	读寄存器空标志 1 表示空

### 9.2.3 数据寄存器 (TxFIFO/RxFIFO)

中文名： 数据传输寄存器  
寄存器位宽： [7: 0]  
偏移量： 0x02  
复位值： 0x00

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

### 9.2.4 外部寄存器 (SPER)

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

位域	位域名称	位宽	访问	描述
7:6	icnt	2	RW	在传输完多少个字节后送出中断申请信号 00 - 1 字节    01 - 2 字节 10 - 3 字节    11 - 3 字节
5:3	Reserved	3	RW	保留
2	mode	1	RW	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	2	RW	与 Spr 一起设定分频的比率

分频系数（分频的源时钟频率是 DDR\_CLK 的一半，参考第 22 章）：

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)

中文名： SPI Flash 参数控制寄存器  
寄存器位宽： [7: 0]  
偏移量： 0x04  
复位值： 0x21

位域	位域名称	位宽	访问	描述
7:4	clk_div	4	RW	时钟分频数选择（分频系数与 {spre,spr} 组合相同）
3	dual_io	1	RW	使用双 I/O 模式，优先级高于快速读模式



2	fast_read	1	RW	使用快速读模式
1	burst_en	1	RW	spl flash 支持连续地址读模式
0	memory_en	1	RW	spl 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)

中文名: SPI Flash 片选控制寄存器  
寄存器位宽: [7: 0]  
偏移量: 0x05  
复位值: 0x00

位域	位域名称	位宽	访问	描述
7:4	csn	4	RW	csn 引脚输出值
3:0	cscn	4	RW	为 1 时对应位的 cs 线由 7:4 位控制

### 9.2.7 时序控制寄存器 (SFC\_TIMING)

中文名: SPI Flash 时序控制寄存器  
寄存器位宽: [7: 0]  
偏移量: 0x06  
复位值: 0x03

位域	位域名称	位宽	访问	描述
7:2	Reserved	6	RW	保留
1:0	tCSH	2	RW	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

### SPI 主控制器外部接口时序图

如图 10-2 所示，SPI 主控制器发送数据时，数据提前半拍放在 MOSI 引线上，接着从设备端用时钟边沿锁存数据。根据时钟极性 (CPOL) 和时钟相位 (CPHA) 的设置，有 4 种可能的时序关系。

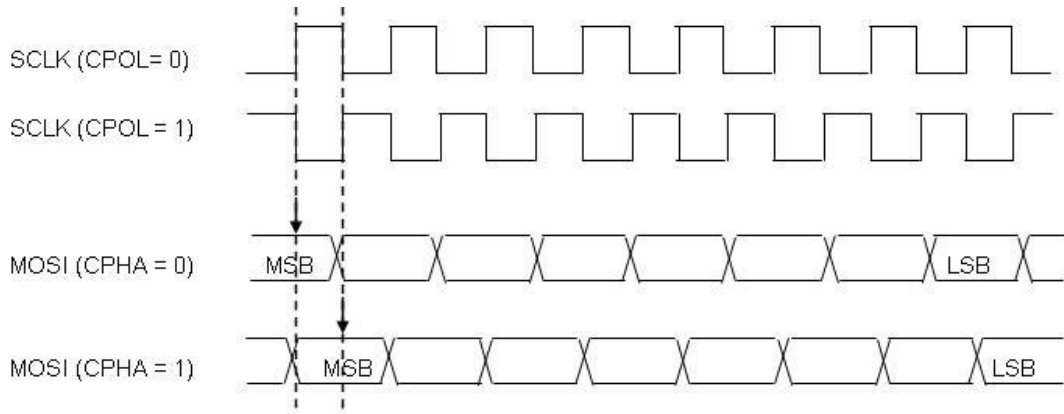
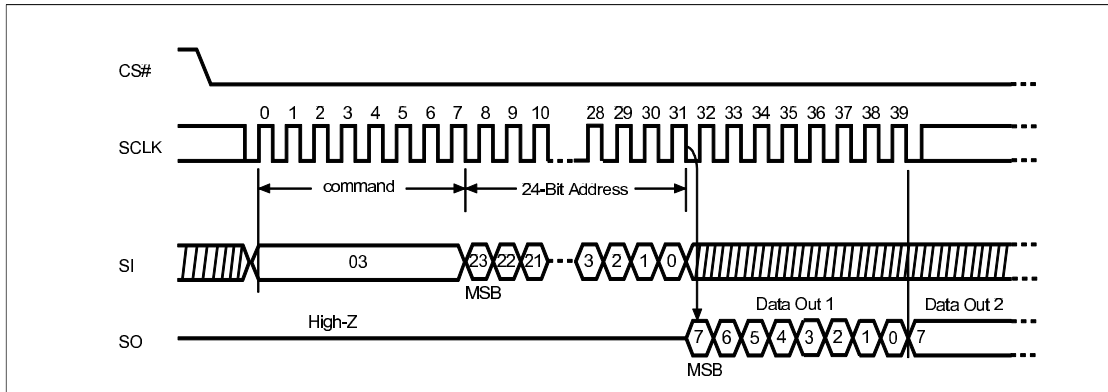


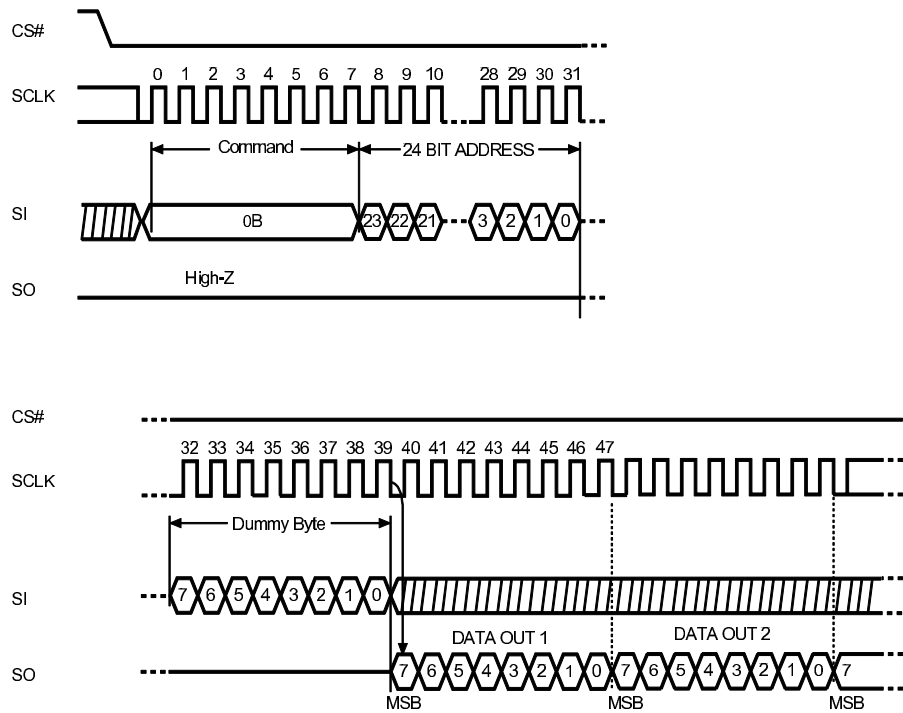
图 10-2 SPI 主控制器时序图

### SPI Flash 访问时序图

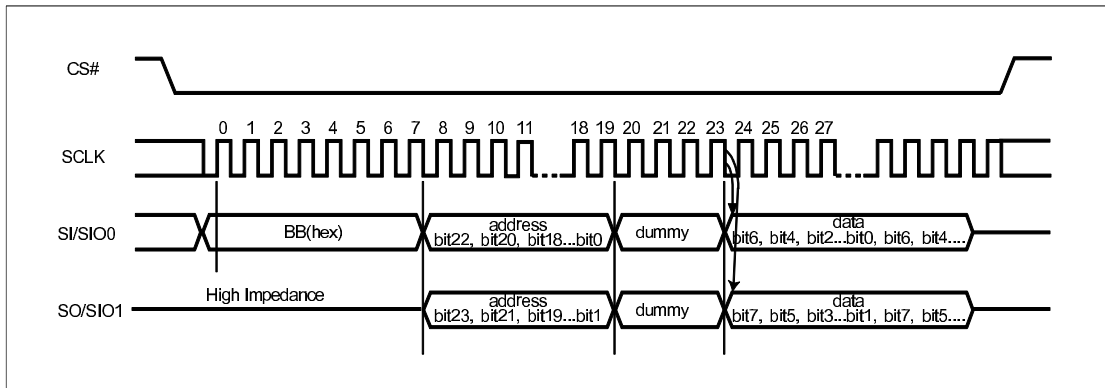
- 标准读模式



- 快速读模式



● 双 I/O 模式



在所有模式下，若没有使能连续地址读，则 CS 将在传输完一个字节数据后拉高。

## 9.4 SPI Flash 控制器使用指南

### 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

## 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。

## 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 SPI1

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

### 10.1 SPI 主控制器结构

SPI1 和 SPI0 的实现一样，系统启动地址不会映射到 SPI1 控制器，所以 SPI1 不支持系统启动。SPI1 的外部存储地址空间是 0xbf80,0000–0xbfbf,fff 共 4MB。所有结构和配置相关请参考第 10 章信息。

## 11 Conf and Interrupt

### 11.1 配置和中断控制器总体描述

1B 芯片内置简单、灵活的中断控制器。1B 芯片的中断控制器除了管理 GPIO 输入的中断信号外，中断控制器还处理内部事件引起的中断。所有的中断寄存器的位域安排相同，一个中断源对应其中一位。中断控制器共四个中断输出连接 CPU 模块，分别对应 INT0, INT1, INT2, INT3。芯片支持 64 个内部中断和 64 个 GPIO 的中断；其中 INT0 和 INT1 分别对应于 64 个内部中断的前后 32 位，INT2 和 INT3 对应于 62 个外部 GPIO 中断。具体如下表所示：

	INT0	INT1	INT2	INT3
31	保留	保留	保留	保留
30	UART5	保留	GPIO30	保留
29	UART4	保留	GPIO29	GPIO61
28	TOY_TICK	保留	GPIO28	GPIO60
27	RTC_TICK	保留	GPIO27	GPIO59
26	TOY_INT2	保留	GPIO26	GPIO58
25	TOY_INT1	保留	GPIO25	GPIO57
24	TOY_INT0	保留	GPIO24	GPIO56
23	RTC_INT2	保留	GPIO23	GPIO55
22	RTC_INT1	保留	GPIO22	GPIO54
21	RTC_INT0	保留	GPIO21	GPIO53
20	PWM3	保留	GPIO20	GPIO52
19	PWM2	保留	GPIO19	GPIO51
18	PWM1	保留	GPIO18	GPIO50
17	PWM0	保留	GPIO17	GPIO49
16	保留	保留	GPIO16	GPIO48
15	DMA2	保留	GPIO15	GPIO47
14	DMA1	保留	GPIO14	GPIO46
13	DMA0	保留	GPIO13	GPIO45
12	保留	保留	GPIO12	GPIO44
11	保留	保留	GPIO11	GPIO43
10	AC97	保留	GPIO10	GPIO42
9	SPI1	保留	GPIO09	GPIO41
8	SPI0	保留	GPIO08	GPIO40
7	CAN1	保留	GPIO07	GPIO39
6	CAN0	保留	GPIO06	GPIO38
5	UART3	保留	GPIO05	GPIO37
4	UART2	保留	GPIO04	GPIO36
3	UART1	Gmac1	GPIO03	GPIO35
2	URAT0	Gmac0	GPIO02	GPIO34

1	保留	Ohci	GPIO01	GPIO33
0	保留	Ehci	GPIO00	GPIO32

1B 互联基于 XBAR, XBAR 对各主设备的窗口进行配置, 每个主设备配置 8 各窗口, 每个窗口的寄存器包括 BASE, MASK, MMAP 组成。

## 11.2 中断控制器寄存器描述

中断的使用首先要设置中断使能寄存器中相应的位来使能该中断, 系统复位时默认不使能中断。然后设置中断触发类型寄存器、中断极性控制寄存器和中断输出控制寄存器相应的属性。最后当发生中断时, 通过中断状态寄存器查看相应的中断源。

中断触发方式分为电平触发与边沿触发两种, 电平触发方式时, 中断控制器内部不寄存外部中断, 此时对中断处理的响应完成后只需要清除对应设备上的中断就可以清除对 CPU 的相应中断。例如, 上行网口向 CPU 发出接收包中断, 网络驱动处理中断后, 只要清除上行网口内部的中断寄存器中的中断状态, 就可以清除 CPU 中断控制器的中断状态, 而不需要通过对应的 INT\_CLR 对 CPU 进行清中断。但是在边沿触发的方式下, 中断控制器会寄存外部中断, 此时软件处理中断时, 需要通过写对应的 INT\_CLR, 清除 CPU 中断控制器内部的对应中断状态。另外, 在边沿触发的情况下, 用户可以通过写 INT\_SET 位强置中断控制器的对应中断状态。

偏移地址	位	寄存器	描述	读写特性
0xbf01040	32	INTISR0	中断控制状态寄存器 0	RO
0xbf01044	32	INTIEN0	中断控制使能寄存器 0	R/W
0xbf01048	32	INTSET0	中断置位寄存器 0	R/W
0xbf0104c	32	INTCLR0	中断清空寄存器 0	R/W
0xbf01050	32	INTPOL0	高电平触发中断使能寄存器 0	R/W
0xbf01054	32	INTEDGE0	边沿触发中断使能寄存器 0	R/W
0xbf01058	32	INTISR1	中断控制状态寄存器 1	RO
0xbf0105c	32	INTIEN1	中断控制使能寄存器 1	R/W
0xbf01060	32	INTSET1	中断置位寄存器 1	R/W
0xbf01064	32	INTCLR1	中断清空寄存器 1	R/W
0xbf01068	32	INTPOL1	高电平触发中断使能寄存器 1	R/W
0xbf0106c	32	INTEDGE1	边沿触发中断使能寄存器 1	R/W
0xbf01070	32	INTISR2	中断控制状态寄存器 2	RO
0xbf01074	32	INTIEN2	中断控制使能寄存器 2	R/W
0xbf01078	32	INTSET2	中断置位寄存器 2	R/W
0xbf0107c	32	INTCLR2	中断清空寄存器 2	R/W
0xbf01080	32	INTPOL2	高电平触发中断使能寄存器 2	R/W
0xbf01084	32	INTEDGE2	边沿触发中断使能寄存器 2	R/W
0xbf01088	32	INTISR3	中断控制状态寄存器 3	RO
0xbf0108c	32	INTIEN3	中断控制使能寄存器 3	R/W
0xbf01090	32	INTSET3	中断置位寄存器 3	R/W
0xbf01094	32	INTCLR3	中断清空寄存器 3	R/W
0xbf01098	32	INTPOL3	高电平触发中断使能寄存器 3	R/W
0xbf0109c	32	INTEDGE3	边沿触发中断使能寄存器 3	R/W
0xbf010c0	32	GPIOCFG0	GPIO 配置寄存器 0	R/W
0xbf010c4	32	GPIOCFG1	GPIO 配置寄存器 1	R/W
0xbf010d0	32	GPIOOE0	GPIO 配置寄存器输入使能 0	R/W
0xbf010d4	32	GPIOOE1	GPIO 配置寄存器输入使能 1	R/W
0xbf010e0	32	GPIOIN0	GPIO 配置寄存器输入寄存器 0	R/W
0xbf010e4	32	GPIOIN1	GPIO 配置寄存器输入寄存器 1	R/W
0xbf010f0	32	GPIOOUT0	GPIO 配置寄存器输出寄存器 0	R/W
0xbf010f4	32	GPIOOUT1	GPIO 配置寄存器输出寄存器 1	R/W
0xbf01160	32	ORDER_REG_ADDR	DMA 模块控制寄存器位	



XBAR 上各主设备地址窗口的配置如下表所示：

偏移地址	位	寄存器	描述	读写特性
0xbf000000	64	CPU_WIN0_BASE	Cpu 配置窗口 0 基地址	R/W
0xbf000008	64	CPU_WIN1_BASE	Cpu 配置窗口 1 基地址	R/W
0xbf000010	64	CPU_WIN2_BASE	Cpu 配置窗口 2 基地址	R/W
0xbf000018	64	CPU_WIN3_BASE	Cpu 配置窗口 3 基地址	R/W
0xbf000020	64	CPU_WIN4_BASE	Cpu 配置窗口 4 基地址	R/W
0xbf000028	64	CPU_WIN5_BASE	Cpu 配置窗口 5 基地址	R/W
0xbf000030	64	CPU_WIN6_BASE	Cpu 配置窗口 6 基地址	R/W
0xbf000038	64	CPU_WIN7_BASE	Cpu 配置窗口 7 基地址	R/W
0xbf000040	64	CPU_WIN0_MASK	Cpu 配置窗口 0 掩码地址	R/W
0xbf000048	64	CPU_WIN1_MASK	Cpu 配置窗口 1 掩码地址	R/W
0xbf000050	64	CPU_WIN2_MASK	Cpu 配置窗口 2 掩码地址	R/W
0xbf000058	64	CPU_WIN3_MASK	Cpu 配置窗口 3 掩码地址	R/W
0xbf000060	64	CPU_WIN4_MASK	Cpu 配置窗口 4 掩码地址	R/W
0xbf000068	64	CPU_WIN5_MASK	Cpu 配置窗口 5 掩码地址	R/W
0xbf000070	64	CPU_WIN6_MASK	Cpu 配置窗口 6 掩码地址	R/W
0xbf000078	64	CPU_WIN7_MASK	Cpu 配置窗口 7 掩码地址	R/W
0xbf000080	64	CPU_WIN0_MMAP	Cpu 配置窗口 0 映射地址	R/W
0xbf000088	64	CPU_WIN1_MMAP	Cpu 配置窗口 1 映射地址	R/W
0xbf000090	64	CPU_WIN2_MMAP	Cpu 配置窗口 2 映射地址	R/W
0xbf000098	64	CPU_WIN3_MMAP	Cpu 配置窗口 3 映射地址	R/W
0xbf0000a0	64	CPU_WIN4_MMAP	Cpu 配置窗口 4 映射地址	R/W
0xbf0000a8	64	CPU_WIN5_MMAP	Cpu 配置窗口 5 映射地址	R/W
0xbf0000b0	64	CPU_WIN6_MMAP	Cpu 配置窗口 6 映射地址	R/W
0xbf0000b8	64	CPU_WIN7_MMAP	Cpu 配置窗口 7 映射地址	R/W

## 12 DMA

### 12.1 DMA 控制器结构描述

DMA 来进行 DDR2 与设备间数据搬移工作，提高系统数据传输的效率。本章介绍的 DMA 是专用 DMA，DMA 共有三路，分别对应 NAND、AC97 播放、AC97 录音的数据传输。

DMA 传送数据的过程由三个阶段组成：

- a) 传送前的预处理：由 CPU 完成以下步骤：配置 DMA 描述符相关的寄存器。
- b) 数据传送：在 DMA 控制器的控制下自动完成。
- c) 传送结束处理：发送中断请求。

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

CPU 通过配置 DMA 寄存器，将来自于 DDR2 或设备的数据保存在缓存中，将缓存中的数据写入要对应的内存或设备中去，最后发送 DMA 传输结束信号。在 DMA 传输过程中，CPU 可以随时监听 DMA 的工作状态。

### 12.2 DMA 控制器与 APB 设备的交互

在 1B 中，使用 DMA 的 APB 设备包括 NAND，AC97，每个设备都有单独的 DMA 控制器。AC97 的写通道是双通道，且 AC97 写使能的判断条件的 DMA\_DADDR[31]=1，DMA\_DADDR[29:28]域为 AC97 写模式选择域，判断 AC97 写操作是字节、半字或者字操作，与 AC97 的写模式配置一致。所以，如果是写 AC97 的写操作，需要在配置 DMA 描述符时，将 DAM\_DADDR[31]配置为 1，将 DMA\_DADDR[29:28]配置为需要的 AC97 写模式。

### 12.3 DMA 控制器

#### 12.3.1 ORDER\_ADDR\_IN

中文名： 该寄存器广播到三路 DMA，被选中的 DMA 根据寄存器的配置开始工作  
 寄存器位宽： [31: 0]  
 地址： 0xbf01160  
 复位值： 0x00000000

位域	位域名称	位宽	访问	描述
31:6	Ask_addr	26	R/W	被选中 DMA 第一个描述符地址的高 26 位，低 6 位为 0；相当于 26 为的 Ask_addr 左移 6 位。
5	保留	1		
4	dma_stop	1	R/W	用户请求停止 DMA 操作； 完成当前数据读写操作后，停止操作
3	dma_start	1	R/W	可以开始读描述符链的第一个 DMA 描述符； 当第一个描述符相关的寄存器读回后，该位清零
2	Ask_valid	1	R/W	用户请求将当前 DMA 操作的相关信息写回到指定的内存地址； 当用户写回 DMA 操作相关信息后，该位清零。
1:0	Dev_num	2	R	2'b00 nand flash 2'b01 AC97 read device 2'b10 AC97 write device

说明：

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

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

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

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

### 12.3.2 DMA\_ORDER\_ADDR

中文名： 下一个描述符地址寄存器

寄存器位宽： [31: 0]

基地址： Ask\_addr<<6

偏移地址： 0x0

复位值： 0x00000000

位域	位域名称	位宽	访问	描述
31:1	dma_order_addr	31	R/W	存储器内部下一个描述符地址寄存器
0	Dma_order_en	1	R/W	描述符是否有效信号

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

### 12.3.3 DMA\_SADDR

中文名： 内存地址寄存器

寄存器位宽： [31: 0]

基地址: Ask\_addr<<6  
偏移地址: 0x4  
复位值: 0x00000000

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

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

### 12.3.4 DMA\_DADDR

中文名: 设备地址寄存器  
寄存器位宽: [31: 0]  
基地址: Ask\_addr<<6  
偏移地址: 0x8  
复位值: 0x00000000

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

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

### 12.3.5 DMA\_LENGTH

中文名: 长度寄存器  
寄存器位宽: [31: 0]  
基地址: Ask\_addr<<6  
偏移地址: 0xc  
复位值: 0x00000000

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

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

### 12.3.6 DMA\_STEP\_LENGTH

中文名: 间隔长度寄存器  
寄存器位宽: [31: 0]

基地址: Ask\_addr<<6  
偏移地址: 0x10  
复位值: 0x00000000

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

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

### 12.3.7 DMA\_STEP\_TIMES

中文名: 循环次数寄存器  
寄存器位宽: [31: 0]  
基地址: Ask\_addr<<6  
偏移地址: 0x14  
复位值: 0x00000000

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

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

### 12.3.8 DMA\_CMD

中文名: 控制寄存器  
寄存器位宽: [31: 0]  
基地址: Ask\_addr<<6  
偏移地址: 0x18  
复位值: 0x00000000

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

说明: dma\_single\_trans\_over=1 指一次 DMA 操作执行结束, 此时 length=0 且 step\_times=1, 开始取下个 DMA 操作的描述符。下个 DMA 操作的描述符地址保存在 DMA\_ORDER\_ADDR 寄存器中, 如果 DMA\_ORDER\_ADDR 寄存器中 dma\_order\_en=0, 则 dma\_trans\_over=1, 整个 dma 操作结束, 没有新的描述符要读; 如果 dma\_order\_en=1,

则 dma\_trans\_over 置为 0，开始读下个 dma 描述符。dma\_int 为 DMA 的中断，如果没有中断屏蔽，在一次配置的 DMA 操作结束后发生中断。CPU 处理完中断后可以直接将其置低，也可以等到 DMA 进行下次传输时自动置低。dma\_int\_mask 为对应 dma\_int 的中断屏蔽。dma\_read\_state 说明了 DMA 当前的读状态。dma\_write\_state 说明了 DMA 当前的写状态。

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

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

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

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

## 13 UART

1B集成了 12 个UART核，通过APB总线与总线桥通信。UART控制器提供与MODEM或其他外部设备串行通信的功能，例如与另外一台计算机，以RS232 为标准使用串行线路进行通信。该控制器在设计上能很好地兼容国际工业标准半导体设备 16550A。

### 13.1 UART 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）、访问寄存器模块（Register Access Control），这些模块之间的关系见下图所示。主要模块功能及特征描述如下：

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

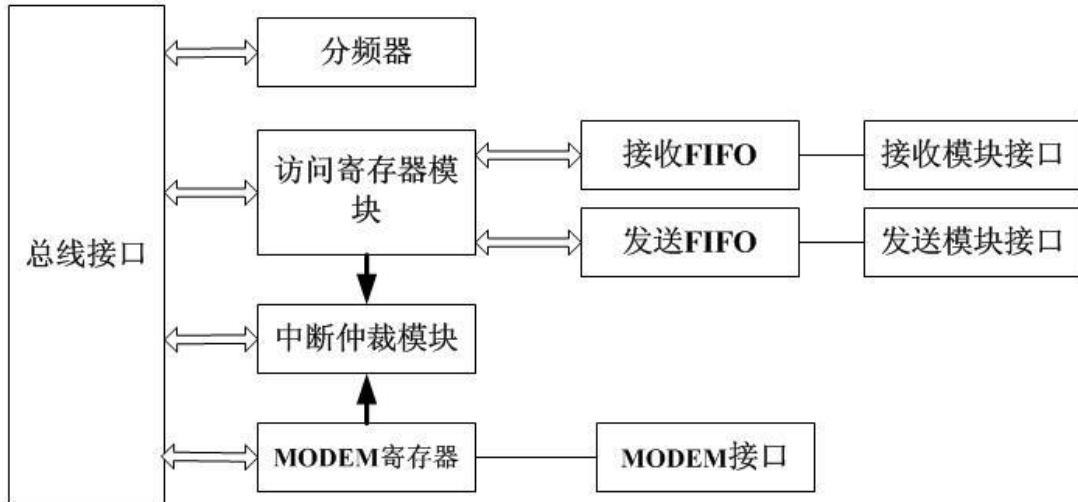


图 14-1 UART 控制器结构

### 13.2 UART 控制器寄存器

1B 内共 12 个并行工作的 UART 接口，其功能寄存器完全一样，只是访问基址不一样。

UART0 寄存器物理地址基址为 0xbfE40000。

UART0\_1 寄存器物理地址基址为 0xbfE41000。

UART0\_2 寄存器物理地址基址为 0xbfE42000。

UART0\_3 寄存器物理地址基址为 0xbfE43000。

UART1 寄存器物理地址基址为 0xbfE44000。

UART1\_1 寄存器物理地址基址为 0xbfE45000。

UART1\_2 寄存器物理地址基址为 0xbfE46000。

UART1\_3 寄存器物理地址基址为 0xbfE47000。

UART2 寄存器物理地址基址为 0xbfE48000。

UART3 寄存器物理地址基址为 0xbfE4c000。

UART4 寄存器物理地址基址为 0xbfE6c000。

UART5 寄存器物理地址基址为 0xbfE7c000。

UART0 和 UART1 都实现了一分四功能，UART0 有 8 个 PAD；UART1 只有 4 个 PAD，同时利用 CAN0 和 CAN1 的 4 个 PAD。所以在 CAN0/CAN1 不用的时候，1B 最多可以提供出来 12 个两线 UART。

0XBFE7\_8038:uart\_split

位域	位域名称	访问	描述
1	Uart1_split	R/W	UART1 被分成四个独立两线 UART
0	Uart0_split	R/W	UART0 被分成四个独立两线 UART

0XBFD0\_0424:uart1\_use\_can

5	UART1_3_USE_CAN1	UART1_3 利用 CAN1 实现
4	UART1_2_USE_CAN0	UART1_2 利用 CAN0 实现



### 13.2.1 数据寄存器 (DAT)

中文名： 数据传输寄存器  
寄存器位宽： [7: 0]  
偏移量： 0x00  
复位值： 0x00

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

### 13.2.2 中断使能寄存器 (IER)

中文名： 中断使能寄存器  
寄存器位宽： [7: 0]  
偏移量： 0x01  
复位值： 0x00

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

### 13.2.3 中断标识寄存器 (IIR)

中文名： 中断源寄存器  
寄存器位宽： [7: 0]  
偏移量： 0x02  
复位值： 0xc1

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

中断控制功能表

Bit 3	Bit 2	Bit 1	优先级	中断类型	中断源	中断复位控制
0	1	1	1 <sup>st</sup>	接收线路状态	奇偶、溢出或帧错误，或打断中断	读 LSR
0	1	0	2 <sup>nd</sup>	接收到有效数据	FIFO 的字符个数达到 trigger 的水平	FIFO 的字符个数低于 trigger 的值
1	1	0	2 <sup>nd</sup>	接收超时	在 FIFO 至少有一个字符，但在 4 个字符时间内没有任何操作，包括读和写操作	读接收 FIFO

0	0	1	3 <sup>rd</sup>	传输保存寄存器为空	传输保存寄存器为空	写数据到 THR 或者多 IIR
0	0	0	4 <sup>th</sup>	Modem 状态	CTS, DSR, RI or DCD.	读 MSR

### 13.2.4 FIFO 控制寄存器 (FCR)

中文名: FIFO 控制寄存器, FIFO 深度 16

寄存器位宽: [7: 0]

偏移量: 0x02

复位值: 0xc0

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

### 13.2.5 线路控制寄存器 (LCR)

中文名: 线路控制寄存器

寄存器位宽: [7: 0]

偏移量: 0x03

复位值: 0x03

位域	位域名称	位宽	访问	描述
7	dlab	1	RW	分频锁存器访问位 '1' - 访问操作分频锁存器 '0' - 访问操作正常寄存器
6	bcb	1	RW	打断控制位 '1' - 此时串口的输出被置为 0(打断状态). '0' - 正常操作
5	spb	1	RW	指定奇偶校验位 '0' - 不用指定奇偶校验位 '1' - 如果 LCR[4]位是 1 则传输和检查奇偶校验位为 0。如果 LCR[4]位是 0 则传输和检查奇偶校验位为 1。
4	eps	1	RW	奇偶校验位选择 '0' - 在每个字符中有奇数个 1 (包括数据和奇偶校验位) '1' - 在每个字符中有偶数个 1
3	pe	1	RW	奇偶校验位使能 '0' - 没有奇偶校验位 '1' - 在输出时生成奇偶校验位, 输入则判断奇偶校验位
2	sb	1	RW	定义生成停止位的位数

				‘0’ – 1 个停止位 ‘1’ – 在 5 位字符长度时是 1.5 个停止位，其他长度是 2 个停止位
1:0	bec	2	RW	设定每个字符的位数 ‘00’ – 5 位    ‘01’ – 6 位 ‘10’ – 7 位    ‘11’ – 8 位

### 13.2.6 MODEM 控制寄存器 (MCR)

中文名: Modem 控制寄存器

寄存器位宽: [7: 0]

偏移量: 0x04

复位值: 0x00

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

### 13.2.7 线路状态寄存器 (LSR)

中文名: 线路状态寄存器

寄存器位宽: [7: 0]

偏移量: 0x05

复位值: 0x00

位域	位域名称	位宽	访问	描述
7	ERROR	1	R	错误表示位 ‘1’ – 至少有奇偶校验位错误，帧错误或打断中断的一个。 ‘0’ – 没有错误
6	TE	1	R	传输为空表示位 ‘1’ – 传输 FIFO 和传输移位寄存器都为空。给传输 FIFO 写数据时清零 ‘0’ – 有数据
5	TFE	1	R	传输 FIFO 位空表示位 ‘1’ – 当前传输 FIFO 为空，给传输 FIFO 写数据

位域	位域名称	位宽	访问	描述
				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。

### 13.2.8 MODEM 状态寄存器 (MSR)

中文名: Modem 状态寄存器  
寄存器位宽: [7: 0]  
偏移量: 0x06  
复位值: 0x00

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

### 13.2.9 分频锁存器

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

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

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

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

模块中被分频时钟 `clock_a` 的频率是 `DDR_clk` 频率的的一半（`DDR_clk` 配置见 22 章）；假设分频锁存器的值为 `prescale`，波特率为 `clock_baud`（波特率根据用户需要和外部 UART 连接特性确定），则应满足如下关系：

$$\text{Prcescale} = \text{clock\_a}/(16*\text{clock\_baud})$$

$$\text{或者 Prcescale} = \text{DDR\_clk}/(32*\text{clock\_baud})$$

## 14 CAN

### 14.1 概述

1B集成了两路CAN接口控制器。CAN 总线是由发送数据线TX和接收数据线RX构成的串行总线，可发送和接收数据。器件与器件之间进行双向传送，最高传送速率 1Mbps。

本芯片集成了两个 CAN 总线控制器，引脚复用 GPIO[41:38]，对应关系如下表。

CAN0_RX	GPIO38
CAN0_TX	GPIO39
CAN1_RX	GPIO40
CAN1_TX	GPIO41

两个 CAN 总线控制器的中断连接到中断控制的第一组寄存器中，其中 can0 的中断对应 bit6，can1 的中断对应 bit7。参考第 12 章的说明。

CAN0 总线控制器的寄存器基地址为 0xbf50000 开始的 16KB;

CAN1 总线控制器的寄存器基地址为 0xbf54000 开始的 16KB

### 14.2 CAN 控制器结构

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

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

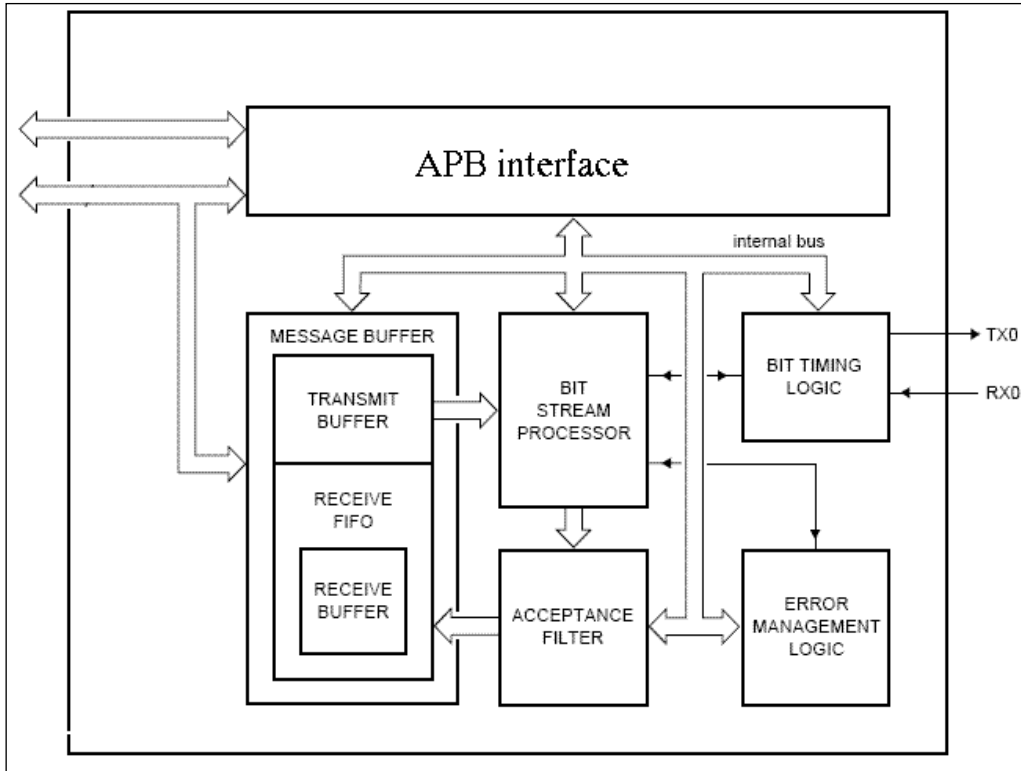


图 15-1 CAN 主控制器结构

本模块包括两个功能完全相同，能够同时工作的 CAN 模块；他们寄存器完全一样，所不同的是他们的物理地址基址。他们的地址基址分别为：

CAN #0 的物理地址基址为 0xbf004400；

CAN #1 的物理地址基址为 0xbf004300。

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

## 14.3 标准模式

### 14.3.1 标准模式地址表

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

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

CAN 地址	段	工作模式		复位模式	
		读	写	读	写
0	控制	控制	控制	控制	控制
1		FF	命令	FF	命令
2		状态	—	状态	—
3		FF	—	中断	—

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

### 14.3.2 控制寄存器 (CR)

中文名： 控制寄存器

寄存器位宽： [7: 0]

偏移量： 0x00

复位值： 0x01

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

复位请求位被设为 0 后控制器将会等待

- a) 一个总线空闲信号（11 个弱势位），如果前一次复位请求是硬件复位或 CPU 初始复位。
- b) 128 个总线空闲，如果前一次复位请求是 CAN 控制器在重新进入总线开启模式前初始化总线造成的。

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



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

### 14.3.3 命令寄存器（CMR）

中文名： 命令寄存器

寄存器位宽： [7: 0]

偏移量： 0x01

复位值： 0x00

命令寄存器对微控制器来说是只写存储器如果去读这个地址返回值是 1111 1111

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

### 14.3.4 状态寄存器（SR）

中文名： 状态寄存器

寄存器位宽： [7: 0]

偏移量： 0x02

复位值： 0x00

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

### 14.3.5 中断寄存器（IR）

中文名： 中断寄存器

寄存器位宽： [7: 0]

偏移量： 0x03

复位值： 0x00

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

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

### 14.3.6 验收代码寄存器 (ACR)

中文名： 验收代码寄存器  
寄存器位宽： [7: 0]  
偏移量： 0x04  
复位值： 0x00

在复位情况下，该寄存器是可以读写的。

位域	位域名称	位宽	访问	描述
7:0	AC	8	RW	ID 验收代码

### 14.3.7 验收屏蔽寄存器 (AMR)

中文名： 验收屏蔽寄存器  
寄存器位宽： [7: 0]  
偏移量： 0x05  
复位值： 0x00

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

位域	位域名称	位宽	访问	描述
7:0	AM	8	RW	ID 屏蔽位

### 14.3.8 发送缓冲区列表

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

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

### 14.3.9 接收缓冲区列表

接收缓冲区的配置和发送缓冲区的一样，只是地址变为 20—29。

## 14.4 扩展模式

### 14.4.1 扩展模式地址表

CAN 地址	工作模式		复位模式	
	读	写	读	写
0	控制	控制	控制	控制
1	0	命令	0	命令
2	状态	—	状态	—
3	中断	—	中断	—
4	中断使能	中断使能	中断使能	中断使能
5	—	—	验收屏蔽	验收屏蔽
6	总线定时 0	—	总线定时 0	总线定时 0
7	总线定时 1	—	总线定时 1	总线定时 1
8	保留	保留	保留	保留
9	保留	保留	保留	保留
10	保留	保留	保留	保留
11	仲裁丢失捕捉	—	仲裁丢失捕捉	—
12	错误代码捕捉	—	错误代码捕捉	—
13	错误警报限制	—	错误警报限制	—
14	RX 错误计数器	—	RX 错误计数器	—
15	TX 错误计数器	—	TX 错误计数器	—
16	RX 帧信息	TX 帧信息	验收代码 0	验收代码 0
17	RX 识别码 1	TX 识别码 1	验收代码 1	验收代码 1
18	RX 识别码 2	TX 识别码 2	验收代码 2	验收代码 2
19	RX 识别码 3	TX 识别码 3	验收代码 3	验收代码 3
20	RX 识别码 4	TX 识别码 4	验收屏蔽 0	验收屏蔽 0
21	RX 数据 1	TX 数据 1	验收屏蔽 1	验收屏蔽 1
22	RX 数据 2	TX 数据 2	验收屏蔽 2	验收屏蔽 2
23	RX 数据 3	TX 数据 3	验收屏蔽 3	验收屏蔽 3
24	RX 数据 4	TX 数据 4	—	—
25	RX 数据 5	TX 数据 5	—	—
26	RX 数据 6	TX 数据 6	—	—
27	RX 数据 7	TX 数据 7	—	—
28	RX 数据 8	TX 数据 8	—	—
29	RX 信息计数器	—	RX 信息计数器	—

### 14.4.2 模式寄存器（MOD）

中文名： 模式寄存器  
寄存器位宽： [7: 0]  
偏移量： 0x00  
复位值： 0x01

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

复位请求位被设为 0 后控制器将会等待

- a) 一个总线空闲信号（11 个弱势位），如果前一次复位请求是硬件复位或 CPU 初始复位。
- b) 128 个总线空闲，如果前一次复位请求是 CAN 控制器在重新进入总线开启模式前初始化总线造成的。

位域	位域名称	位宽	访问	描述
7: 5	Reserve	3	—	保留
4	SM	1	RW	睡眠模式
3	AFM	1	RW	单/双滤波模式
2	STM	1	RW	正常工作模式
1	LOM	1	RW	只听模式
0	RM	1	RW	复位模式

### 14.4.3 命令寄存器（CMR）

中文名： 命令寄存器  
寄存器位宽： [7: 0]  
偏移量： 0x01  
复位值： 0x00

命令寄存器对微控制器来说是只写存储器如果去读这个地址返回值是 1111 1111

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

### 14.4.4 状态寄存器（SR）

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

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

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

#### 14.4.5 中断寄存器 (IR)

中文名： 中断寄存器

寄存器位宽： [7: 0]

偏移量： 0x03

复位值： 0x00

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

#### 14.4.6 中断使能寄存器 (IER)

中文名： 中断使能寄存器

寄存器位宽： [7: 0]

偏移量： 0x04

复位值： 0x00

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

#### 14.4.7 仲裁丢失捕捉寄存器 (IER)

中文名： 仲裁丢失捕捉寄存器

寄存器位宽： [7: 0]

偏移量： 0xB

复位值： 0x00

位域	位域名称	位宽	访问	描述
7: 5	—	3	R	保留
4	BITNO4	1	R	第四位

3	BITNO3	1	R	第三位
2	BITNO2	1	R	第二位
1	BITNO1	1	R	第一位
0	BITNO0	1	R	第零位

位					十进制值	功能
ALC. 4	ALC. 3	ALC. 2	ALC. 1	ALC. 0		
0	0	0	0	0	0	仲裁丢失在识别码的bit1
0	0	0	0	1	1	仲裁丢失在识别码的bit2
0	0	0	1	0	2	仲裁丢失在识别码的bit3
0	0	0	1	1	3	仲裁丢失在识别码的bit4
0	0	1	0	0	4	仲裁丢失在识别码的bit5
0	0	1	0	1	5	仲裁丢失在识别码的bit6
0	0	1	1	0	6	仲裁丢失在识别码的bit7
0	0	1	1	1	7	仲裁丢失在识别码的bit8
0	1	0	0	0	8	仲裁丢失在识别码的bit9
0	1	0	0	1	9	仲裁丢失在识别码的bit10
0	1	0	1	0	10	仲裁丢失在识别码的bit11
0	1	0	1	1	11	仲裁丢失在SRTR位
0	1	1	0	0	12	仲裁丢失在IDE位
0	1	1	0	1	13	仲裁丢失在识别码的bit12
0	1	1	1	0	14	仲裁丢失在识别码的bit13
0	1	1	1	1	15	仲裁丢失在识别码的bit14
1	0	0	0	0	16	仲裁丢失在识别码的bit15
1	0	0	0	1	17	仲裁丢失在识别码的bit16
1	0	0	1	0	18	仲裁丢失在识别码的bit17
1	0	0	1	1	19	仲裁丢失在识别码的bit18
1	0	1	0	0	20	仲裁丢失在识别码的bit19
1	0	1	0	1	21	仲裁丢失在识别码的bit20
1	0	1	1	0	22	仲裁丢失在识别码的bit21
1	0	1	1	1	23	仲裁丢失在识别码的bit22
1	1	0	0	0	24	仲裁丢失在识别码的bit23
1	1	0	0	1	25	仲裁丢失在识别码的bit24
1	1	0	1	0	26	仲裁丢失在识别码的bit25
1	1	0	1	1	27	仲裁丢失在识别码的bit26
1	1	1	0	0	28	仲裁丢失在识别码的bit27
1	1	1	0	1	29	仲裁丢失在识别码的bit28
1	1	1	1	0	30	仲裁丢失在识别码的bit29
1	1	1	1	1	31	仲裁丢失在RTR位

#### 14.4.8 错误警报限制寄存器 (EMLR)

中文名： 错误警报限制寄存器

寄存器位宽： [7: 0]

偏移量： 0xD

复位值: 0x60

位域	位域名称	位宽	访问	描述
7: 0	EML	8	RW	错误警报阈值

#### 14.4.9RX 错误计数寄存器 (RXERR)

中文名: RX 错误计数寄存器

寄存器位宽: [7: 0]

偏移量: 0xE

复位值: 0x60

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

#### 14.4.10TX 错误计数寄存器 (TXERR)

中文名: TX 错误计数寄存器

寄存器位宽: [7: 0]

偏移量: 0xF

复位值: 0x60

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

#### 14.4.11验收滤波器

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

#### 14.4.12RX 信息计数寄存器 (RMCR)

中文名: RX 信息计数寄存器

寄存器位宽: [7: 0]

偏移量: 0x1D

复位值: 0x00

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

### 14.5 公共寄存器

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

#### 14.5.1 总线定时寄存器 0 (BTR0)

中文名: 总线定时寄存器

寄存器位宽: [7: 0]

偏移量: 0x06

复位值: 0x00

注: 在复位模式是可以读写的, 工作模式是只读的

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

### 14.5.2 总线定时寄存器 1 (BTR1)

中文名: 总线定时寄存器 1

寄存器位宽: [7: 0]

偏移量: 0x07

复位值: 0x00

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

### 14.5.3 输出控制寄存器 (OCR)

中文名: 输出控制寄存器

寄存器位宽: [7: 0]

偏移量: 0x08

复位值: 0x00

位域	位域名称	位宽	访问	描述
7: 0	OCR	8	RW	保留



## 15 AC97

### 15.1 AC97 结构描述

在系统里一个 AC97 应用系统如图 15-1 所示。在一个片上系统中，与 AC97 控制器相连的有 3 部分：一是外设总线，接收来自微处理器的控制信息以及配置信息；二是 AC97 Codec，多媒体数字信号编解码器，该解码器对 PCM 信号进行调制，输出人耳接受的模拟声音或者把真实的声音转换为 PCM 信号，转换通过 D/A 转换器实现；三是 DMA 引擎，通过 DMA 的方式写或读 AC97 控制器内部的 FIFO，实现 PCM 音频数据的不间断操作。DMA 是通过微处理器配置的，从处理器设定的内存区域搬运数据给 FIFO 或者把 FIFO 的数据搬运到设定的内存区域。

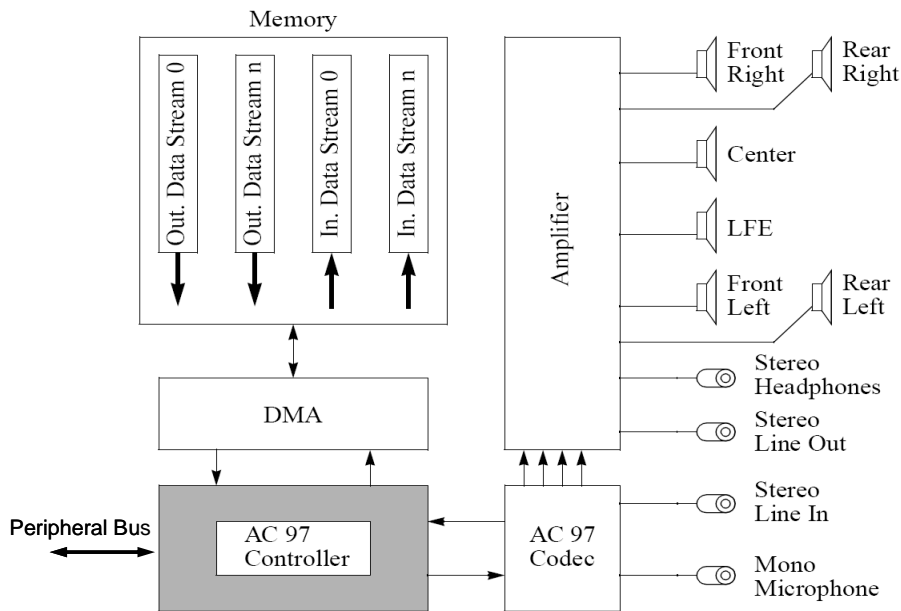


图 15-1 AC97 应用系统

### 15.2 AC97 控制器寄存器

本模块寄存器物理地址基址为 0xBF74000。

寄存器名	宽度	偏移量	描述
CSR	2	0x00	配置状态寄存器
OCC0	24	0x04	输出通道配置寄存器 0
OCC1	24	0x08	保留
OCC2	24	0x0c	保留
ICC	24	0x10	输入通道配置寄存器
CODEC_ID	32	0x14	Codec ID 寄存器
CRAC	32	0x18	Codec 寄存器访问命令
OC0	20	0x20	输出声道 0
OC1	20	0x24	输出声道 1
OC2	20	0x28	保留
OC3	20	0x2c	保留

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

### 15.2.1 CSR 寄存器

中文名： 配置状态寄存器

寄存器位宽： [31: 0]

偏移量： 0x00

复位值： 0x00000000

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

### 15.2.2 OCC 寄存器

中文名： 输出通道配置寄存器

寄存器位宽： [31: 0]

偏移量： 0x04

复位值： 0x00004141

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

### 15.2.3 ICC 寄存器

中文名： 输入通道配置寄存器

寄存器位宽： [31: 0]

偏移量： 0x10

复位值： 0x00410000

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

### 15.2.4 (输入输出) 通道寄存器配置

中文名: 通道配置寄存器

寄存器位宽: [31: 0]

偏移量: 0x14

复位值: 0x00410000

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

### 15.2.5 Codec 寄存器访问命令

中文名: Codec 寄存器访问命令

寄存器位宽: [31: 0]

偏移量: 0x18

复位值: 0x00000000

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

				0: 写
30:23	Reserved	8	R	保留
22:16	CODEC_ADR	7	R/W	Codec 寄存器地址
15:0	CODEC_DAT	16	R/W	Codec 寄存器数据

### 15.2.6 中断状态寄存器/中断掩膜寄存器

中文名： 中断状态/中断掩膜寄存器

寄存器位宽： [31: 0]

偏移量： 0x54/58

复位值： 0x00000000

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

### 15.2.7 中断状态/清除寄存器

中文名： 中断状态/清除寄存器

寄存器位宽： [31: 0]

偏移量： 0x5c

复位值： 0x00000000

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

### 15.2.8 OC 中断清除寄存器

中文名： OC 中断清除

寄存器位宽： [31: 0]

偏移量： 0x60

复位值： 0x00000000

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

### 15.2.9 IC 中断清除寄存器

中文名： IC 中断清除  
寄存器位宽： [31: 0]  
偏移量： 0x64  
复位值： 0x00000000

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

### 15.2.10 CODEC WRITE 中断清除寄存器

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

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

### 15.2.11 CODEC READ 中断清除寄存器

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

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

## 16 I2C

### 16.1 概述

本章给出I2C的详细描述和配置使用。本系统芯片集成了I2C接口，主要用于实现两个器件之间数据的交换。I2C总线是由数据线SDA和时钟SCL构成的串行总线，可发送和接收数据。器件与器件之间进行双向传送，最高传送速率 400kbps。1B芯片共集成 3 路I2C接口，其中第二路和第三路分别通过CAN0 和CAN1 复用实现。复用配置参加 23 章MUX寄存器小节。

I2C_SDA1	CAN0_RX
I2C_SCL1	CAN0_TX
I2C_SDA2	CAN1_RX
I2C_SCL2	CAN1_TX

### 16.2 I2C 控制器结构

I<sup>2</sup>C 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. **时钟发生器模块：**产生分频时钟，同步位命令的工作。
2. **字节命令控制器模块：**将一个命令解释为按字节操作的时序，即把字节操作分解为位操作。
3. **位命令控制器模块：**进行实际数据的传输，以及位命令信号产生。
4. **数据移位寄存器模块：**串行数据移位。

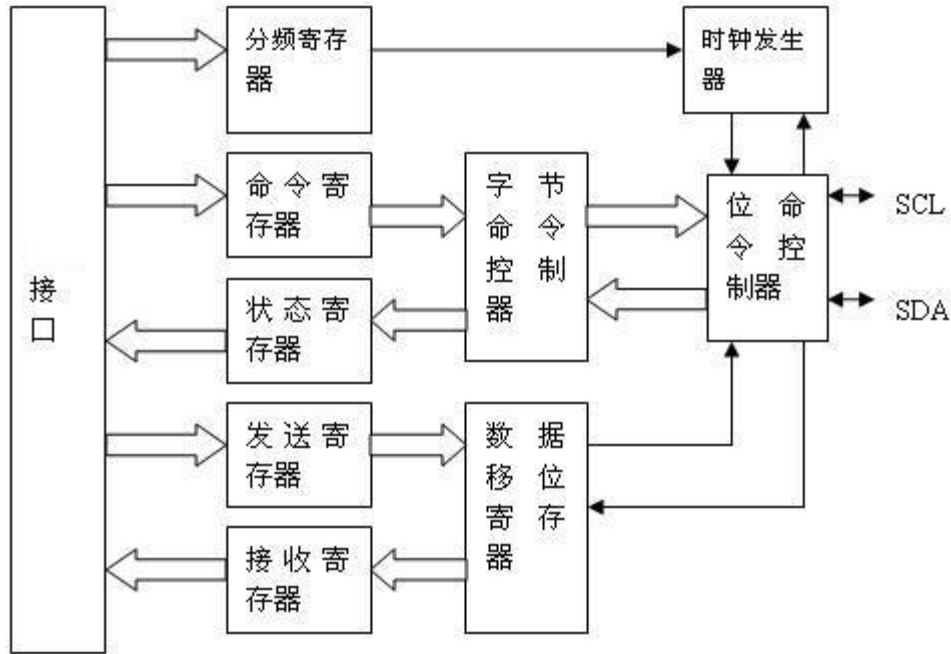


图 17-1 I²C 主控制器结构

### 16.3 I²C 控制器寄存器说明

I2C-0 模块寄存器物理地址基址为：0xbfe58000,地址空间 16KB。

I2C-1 模块寄存器物理地址基址为：0xbfe68000,地址空间 16KB。

I2C-2 模块寄存器物理地址基址为：0xbfe70000,地址空间 16KB。

#### 16.3.1 分频锁存器低字节寄存器（PRERlo）

中文名：分频锁存器低字节寄存器

寄存器位宽：[7: 0]

偏移量：0x00

复位值：0xff

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

#### 16.3.2 分频锁存器高字节寄存器（PRERhi）

中文名：分频锁存器高字节寄存器

寄存器位宽：[7: 0]

偏移量：0x01

复位值：0xff

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

模块中被分频时钟 clock\_a 的频率是 DDR\_clk 频率的的一半（DDR\_clk 配置见 22 章）；假设分频锁存器的值为 prescale，SCL 总线的输出频率为 clock\_s（该时钟根据用户需要和外

部 I2C 设备特性确定)，则应满足如下关系：

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

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

### 16.3.3 控制寄存器（CTR）

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

位域	位域名称	位宽	访问	描述
7	EN	1	RW	模块工作使能位 为 1 正常工作模式, 0 对分频寄存器进行操作
6	IEN	1	RW	中断使能位 为 1 则打开中断
5:0	Reserved	6	RW	保留

### 16.3.4 发送数据寄存器（TXR）

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

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

### 16.3.5 接受数据寄存器（RXR）

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

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

### 16.3.6 命令控制寄存器（CR）

中文名： 命令寄存器  
寄存器位宽： [7: 0]  
偏移量： 0x04  
复位值： 0x00

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



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

都是在 I<sup>2</sup>C 发送数据后硬件自动清零。对这些位读操作时候总是读回‘0’。

### 16.3.7 状态寄存器 (SR)

中文名： 状态寄存器

寄存器位宽： [7: 0]

偏移量： 0x04

复位值： 0x00

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

## 17 PWM

### 17.1 概述

1B 芯片里实现了四路脉冲宽度调节/计数控制器，以下简称 PWM。每一路 PWM 工作和控制方式完全相同。每路 PWM 有一路脉冲宽度输出信号 (pwm\_o)。系统时钟高达 100MHz，计数寄存器和参考寄存器均 24 位数据宽度，使得芯片非常适合高档电机的控制。

四路 PWM 控制器系统的基地址具体如下：

表 17-1 四路控制器描述

名称	基地址 (Base)	中断号
PWM0	0XBFE5:C000	18
PWM1	0XBFE5:C010	19
PWM2	0XBFE5:C020	20
PWM3	0XBFE5:C030	21

每路控制器共有四个控制寄存器，具体描述如下：

表 17-2 控制寄存器描述

名称	地址	宽度	访问	说明
CNTR	Base + 0x0	24	R/W	主计数器
HRC	Base + 0x4	24	R/W	高脉冲定时参考寄存器
LRC	Base + 0x8	24	R/W	低脉冲定时参考寄存器
CTRL	Base + 0xC	8	R/W	控制寄存器

### 17.2 PWM 寄存器说明

#### (1) 实现脉冲宽度功能

CNTR 寄存器可以由系统编程写入获得初始值，系统编程写入完毕后，CNTR 寄存器在系统时钟驱动下不断自加，当到达 LRC 寄存器的值后清 0。然后重新开始不断自加，控制器就产生连续不断的脉冲宽度输出。

表 17-3 主计数器设置

位域	访问	复位值	说明
23: 0	R/W	0x0	主计数器

HRC 寄存器由系统写入，当 CNTR 寄存器的值等于 HRC 的值的时候，控制器产生高脉冲电平。

表 17-4 高脉冲计数器设置

位域	访问	复位值	说明
23: 0	R/W	0x0	高脉冲计数器

LRC 寄存器由系统写入，当 CNTR 寄存器的值等于 LRC 的值的时候，控制器产生低脉冲电平。

表 17-5 低脉冲计数器设置

位域	访问	复位值	说明

23: 0	R/W	0x0	低脉冲计数器
-------	-----	-----	--------

例：如果要产生宽度为系统始终周期 50 倍的高脉宽和 90 倍的低脉宽，在 HRC 中应该配置初始值 $(90-1)=89$ ，在 LRC 寄存器中配置初始值 $(50+90-1)=139$ 。

- (2) 当工作在定时器模式下，CNTR记录内部系统时钟。HRC和LRC寄存器的初始值系统编程写入，当CNTR寄存器的值等于HRC或者LRC的时候，芯片会产生一个中断，这样就实现了定时器功能。
- (3) CTRL控制寄存器，在上面三种工作模式下，控制寄存器的功能不变，根据功能需求选择不同的配置。

表 17-6 控制寄存器设置

位域	访问	复位值	说明
0	R/W	0	EN, 主计数器使能位 置 1 时: CNTR 用来计数 置 0 时: CNTR 停止计数
2: 1	Reserved	2'b0	预留
3	R/W	0	OE,脉冲输出使能控制位,低有效 置 0 时: 脉冲输出使能 置 1 时: 脉冲输出屏蔽
4	R/W	0	SINGLE, 单脉冲控制位 置 1 时: 脉冲仅产生一次 置 0 时: 脉冲持续产生
5	R/W	0	INTE, 中断使能位 置 1 时: 当 CNTR 计数到 LRC 和 CNTR 后送中断 置 0 时: 不产生中断
6	R/W	0	INT, 中断位 读操作: 1 表示有中断产生,0 表示没有中断 写入 1: 清中断
7	R/W	0	CNTR_RST,使得 CNTR 计数器清零 置 1 时: CNTR 计数器清零 置 0 时: CNTR 计数器正常工作

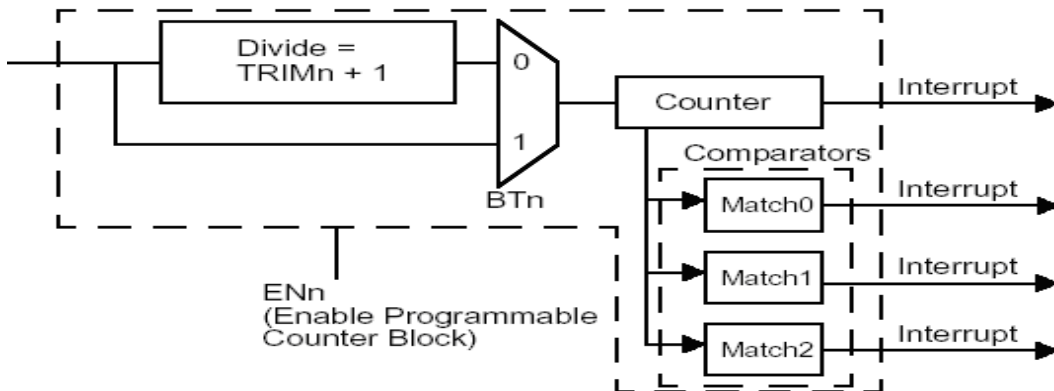
## 18 RTC

### 18.1 概述

龙芯 1B 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 单元运行时功耗为 300 微瓦。

RTC 由 32.768KHz 时钟驱动，内部经可配置的分频器分频进行计数，年月日，时分秒等信息被更新。同时该时钟也用于产生各种定时和计数中断。

RTC 单元由计数器和定时器组成，其构架如下图所示：



### 18.2 寄存器描述

RTC 模块寄存器位于 0xbfe64000——0xbfe67fff 的 16KB 地址空间内，其基地址为 0xbfe64000，所有寄存器位宽均为 32 位。

#### 18.2.1 寄存器地址列表

名称	地址	位宽	RW	描述	复位值
sys_toytrim	0xbfe64020	32	RW	对 32.768kHz 的分频系数（计数器时钟）	
sys_toywrite0	0xbfe64024	32	W	TOY 低 32 位数值写入	
sys_toywrite1	0xbfe64028	32	W	TOY 高 32 位数值写入	
sys_toyread0	0xbfe6402C	32	R	TOY 低 32 位数值读出	
sys_toyread1	0xbfe64030	32	R	TOY 高 32 位数值读出	
sys_toymatch0	0xbfe64034	32	RW	TOY 定时中断 0	
sys_toymatch1	0xbfe64038	32	RW	TOY 定时中断 1	
sys_toymatch2	0xbfe6403C	32	RW	TOY 定时中断 2	
sys_rtctrl	0xbfe64040	32	RW	TOY 和 RTC 控制寄存器	
sys_rtctrim	0xbfe64060	32	RW	对 32.768kHz 的分频系数（定时器	

				时钟)	
sys_rtwrite0	0xbfE64064	32	R	RTC 定时计数写入	
sys_rtread0	0xbfE64068	32	W	RTC 定时计数读出	
sys_rtcmatch0	0xbfE6406C	32	RW	RTC 时钟定时中断 0	
sys_rtcmatch1	0xbfE64070	32	RW	RTC 时钟定时中断 1	
sys_rtcmatch2	0xbfE64074	32	RW	RTC 时钟定时中断 2	

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

### 18.2.2 SYS\_TOYWRITE0

中文名： TOY 计数器低 32 位数值  
寄存器位宽： [31: 0]  
偏移量： 0x20  
复位值： 0x00000000

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

### 18.2.3 SYS\_TOYWRITE1

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

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

### 18.2.4 SYS\_TOYMATCH0/1/2

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

位域	位域名称	访问	缺省	描述
----	------	----	----	----

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

## 18.2.5SYS\_RTCCTRL

中文名: RTC 定时器中断寄存器 0/1/2  
寄存器位宽: [31: 0]  
偏移量: 0x40  
复位值: 0x00000000

位域	位域名称	访问	缺省	描述
31:24	保留	R	0	保留, 置 0
23	ERS	R	0	REN(bit13)写状态
22:21	保留	R	0	保留, 置 0
20	RTS	R	0	Sys_rtctrim 写状态
19	RM2	R	0	Sys_rtcmatch2 写状态
18	RM2	R	0	Sys_rtcmatch2 写状态
17	RM0	R	0	Sys_rtcmatch0 写状态
16	RS	R	0	Sys_rtcwrite 写状态
15	保留	R	0	保留, 置 0
14	BP	R/W	0	旁路 32.768k 晶振 0:选择晶振输入; 1: GPIO8 来驱动计数器, 这是测试模式, GPIO8 通过外部时钟或者 GPIO8 控制器。
13	REN	R/W	0	0: RTC 禁止; 1: RTC 使能
12	BRT	R/W	0	旁路 RTC 分频 0:正常操作; 1:RTC 直接被 32.768k 晶振驱动
11	TEN	R/W	0	0: TOY 禁止; 1: TOY 使能
10	BTT	R/W	0	旁路 TOY 分频 0:正常操作; 1:TOY 直接被 32.768k 晶振驱动
9	保留	R	0	保留, 置 0
8	EO	R/W	0	0: 32.768k 晶振禁止; 1: 32.768k 晶振使能
7	ETS	R	0	TOY 使能写状态
6	保留	R	0	保留, 置 0

5	32S	R	0	0: 32.768k 晶振不工作; 1: 32.768k 晶振正常工作。
4	TTS	R	0	Sys_toytrim 写状态
3	TM2	R	0	Sys_toymatch2 写状态
2	TM1	R	0	Sys_toymatch1 写状态
1	TM0	R	0	Sys_toymatch0 写状态
0	TS	R	0	Sys_toywrite 写状态

### 18.2.6SYS\_RTCMATCH0/1/2

中文名: RTC 定时器中断寄存器 0/1/2  
 寄存器位宽: [31: 0]  
 偏移量: 0x6C/70/74  
 复位值: 0x00000000

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

## 19 NAND

### 19.1 NAND 控制器结构描述

NAND FLASH 控制器最大支持 32GB FLASH 的容量，芯片最多支持 4 个片选和 4 个 RDY 信号。

### 19.2 NAND 控制器寄存器配置描述

NAND 内部的寄存器的设置如下：

地址	寄存器名称
BFE7_8000	NAND_CMD
BFE7_8004	ADDR_L
BFE7_8008	ADDR_H
BFE7_800C	NAND_TIMING
BFE7_8010	ID_L
BFE7_8014	STATUS & ID_H
BFE7_8018	NAND_PARAMETER
BFE7_801C	NAND_OP_NUM
BFE7_8020	CS_RDY_MAP
BFE7_8040	DMA access address

#### 19.2.1 NAND\_CMD (地址：BFE7\_8000)

位	位域名	读写	描述
31:11		R/-	Reserved
23: 20	NAND_CE	R/-	外部 NAND 芯片片选情况
19: 16	NAND_RDY	R/-	外部 NAND 芯片 RDY 情况
15: 11		R/-	Reserved
10	done	R/-	操作完成
9	Spare	R/W	操作发生在 NAND 的 SPARE 区
8	Main	R/W	操作发生在 NAND 的 MAIN 区
7	Read status	R/W	读 NAND 的状态
6	Reset	R/W	Nand 复位操作
5	read id	R/W	读 ID 操作
4	blocks erase	R/W	连续擦除标志，缺省 0；1 有效，连续擦除块的数目由 nand_op_num 决定
3	erase operation	R/W	擦除操作
2	write operation	R/W	写操作
1	read operation	R/W	读操作
0	command valid	R/W	命令有效



### 19.2.2 ADDR\_L (地址: BFE7\_8004)

位	位域名	读写	描述
31:0	Nand_address[31:0]	R/W	读、写、擦除操作起始地址低 32 位

### 19.2.3 ADDR\_H (地址: BFE7\_8008)

位	位域名	读写	描述
31:8		R/-	Reserved
7:0	Nand_address[39:32]	R/W	读、写、擦除操作起始地址高 8 位

### 19.2.4 NAND\_TIMING (地址: BFE7\_800C)

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

### 19.2.5 ID\_L (地址: BFE7\_8010)

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

### 19.2.6 STATUS & ID\_H (地址: BFE7\_8014)

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

### 19.2.7 NAND\_PARAMETER (地址: BFE7\_8018)

位	位域名	读写	描述
31:12		R/-	Reserved
11:8	外部颗粒容量大小	R/W	1: 2Gb 2: 4Gb 3: 8Gb 4: 16Gb 5: 32Gb 6: 64Gb 7: 128Gb 8: 256Gb
7:0		R/-	Reserved

### 19.2.8 NAND\_OP\_NUM (地址: BFE7\_801C)

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

### 19.2.9 CS\_RDY\_MAP (地址: BFE7\_8020)

位	位域名	读写	描述
31:28	NAND_RDY_inter [3]	R/W	4'b0001:NAND_RDY[0] 4'b0010:NAND_RDY[1] 4'b0100:NAND_RDY[2] 4'b1000:NAND_RDY[3]
27:24	NAND_CE[3]	R/W	4'b0001:NAND_CS_inter[0] 4'b0010:NAND_CS_inter[1] 4'b0100:NAND_CS_inter[2] 4'b1000:NAND_CS_inter[3]
23:20	NAND_RDY_inter [2]	R/W	4'b0001:NAND_RDY[0] 4'b0010:NAND_RDY[1] 4'b0100:NAND_RDY[2] 4'b1000:NAND_RDY[3]
19:16	NAND_CS[2]	R/W	4'b0001:NAND_CS_inter[0] 4'b0010:NAND_CS_inter[1] 4'b0100:NAND_CS_inter[2] 4'b1000:NAND_CS_inter[3]
15:12	NAND_RDY_inter [1]	R/W	4'b0001:NAND_RDY[0] 4'b0010:NAND_RDY[1] 4'b0100:NAND_RDY[2] 4'b1000:NAND_RDY[3]
11:8	NAND_CS[1]	R/W	4'b0001:NAND_CS_inter[0] 4'b0010:NAND_CS_inter[1] 4'b0100:NAND_CS_inter[2] 4'b1000:NAND_CS_inter[3]
7:0		R/-	Reserved

### 19.2.10 DMA\_ADDRESS (地址: BFE7\_8040)

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

## 19.3 NAND ADDR 说明

Define:

```
main_op = 0xbfe78000[8];
```

93

```
spare_op = 0xbfe78000[9];
addr_in_nand_flash = { ...A32,A31,A30,A29,A28...A1,A0}=
spare_op ? {addr_h[7:0],addr_l[31:0]} : {addr_h[6:0],addr_l [30:11],1'b0,addr_l [10:0]}
```

NAND 颗粒地址空间组织示例:

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

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

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

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

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

(spare\_op = 1'b0 & main\_op =1'b0) equal to (spare\_op = 1'b0 & main\_op =1'b1);  
nand\_addr\_config\_by\_software=0x30: addr\_in\_nand\_flash = 0x30

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

Examples2:

spare\_op=1'b1 & main\_op=1'b0;(配置出错!! 开始操作不在 spare 区, 下图是可能的错误访问顺序) nand\_addr\_config\_by\_software=0x30: addr\_in\_nand\_flash = 0x30

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

Examples3:

spare\_op = 1'b1 & main\_op =1'b1;  
nand\_addr\_config\_by\_software=0x30: addr\_in\_nand\_flash = 0x30

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

Page 1	op	op	op	op	op	op	op
Page 2	op	op	op	op	op	op	op

Examples4:

(spare\_op=1'b0 & main\_op=1'b0), (equal to spare\_op=1'b0 & main\_op=1'b1);  
nand\_addr\_config\_by\_software =0x830: addr\_in\_nand\_flash = 0x1030(配置出错!! 开始操作在 spare 区, 下图是可能的错误访问顺序)

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

Examples5:

spare\_op = 1'b1 and main\_op = 1'b0;

nand\_addr\_config\_by\_software =0x830: addr\_in\_nand\_flash = 0x830

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

Examples6:

spare\_op = 1'b1 & main\_op = 1'b1;

nand\_addr\_config\_by\_software =0x830: addr\_in\_nand\_flash = 0x830

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

## 20 WATCHDOG

### 20.1 概述

在系统中看门狗定时器（WDT，Watch Dog Timer）实际上是一个计数器，一般给看门狗一个大数，程序开始运行后看门狗开始倒数。如果程序运行正常，过一段时间 CPU 应发出指令让看门狗复位，重新开始倒数。如果看门狗减到 0 就认为程序没有正常工作，强制整个系统复位。下图是看门狗的实现，系统对看门狗进行配置，看门狗内部有个计数器，同时看门狗里面的比较器比较计数器值是否为零，如果为零就发出软复位信号让系统重启。

系统计数采用的时钟频率是为 DDR\_clk 的 2 分频（参考第 22 章）。

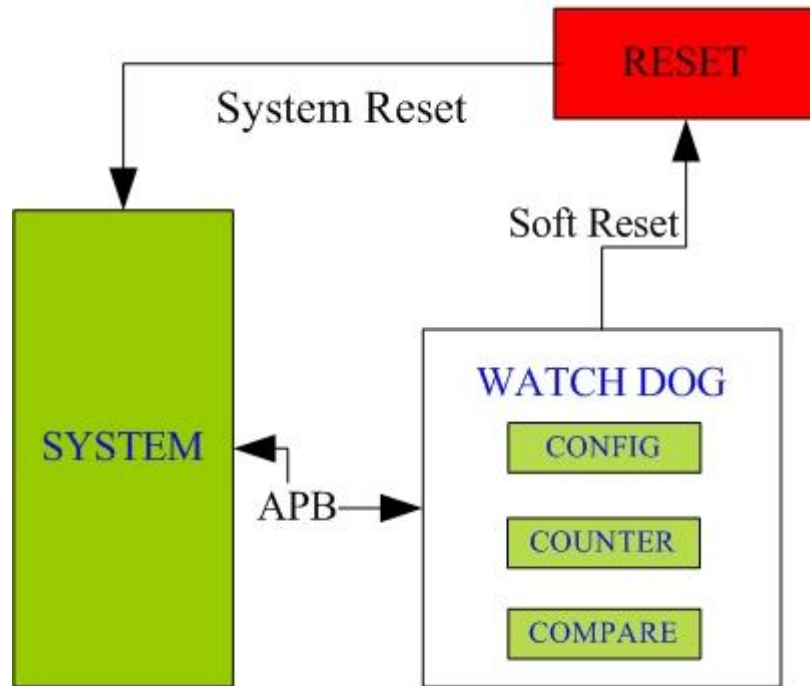


图 20-1 看门狗的结构图

### 20.2 WATCH DOG 寄存器描述

看门狗逻辑可编程寄存器主要有三个，这些寄存器描述如下：

#### 20.2.1 WDT\_EN 地址：（0XBFE5\_C060）

位	位域名	读写	描述
31:1			Reserved
0	WDT_EN	R/W	看门狗使能

### 20.2.2 WDT\_SET (地址: 0XBFE5\_C068)

位	位域名	读写	描述
31:1			Reserved
0	WDT_SET	R/W	看门狗中计数器设置

### 20.2.3 WDT\_timer (地址: 0XBFE5\_C064)

位	位域名	读写	描述
31:0	WDT_timer	R/W	看门狗计数器计数值

系统这三个寄存器的设置顺序: 系统先配置看门狗使能位 WDT\_EN; 然后配置看门狗开始计数器的初始值 WDT\_TIMER, 该值保持在一个特别的寄存器中; 当系统设置 WDT\_SET 后, 计数器开始计数。

## 21 Clock Management

### 21.1 Clock 模块结构描述

时钟模块用来产生系统主要的三个时钟：CPU\_clk、DDR\_clk 和 DC\_clk。系统中集成了一个 PLL，PLL 在系统 RESET 时从外部 PAD 的状态获取初始配置，该 PLL 在进入系统后可以再次配置。PLL 产生一个高频输出 PLL\_clk，系统需要的 CPU\_clk、DDR\_clk 和 DC\_clk 均由此高频输出时钟分频而来。其工作结构图如下：

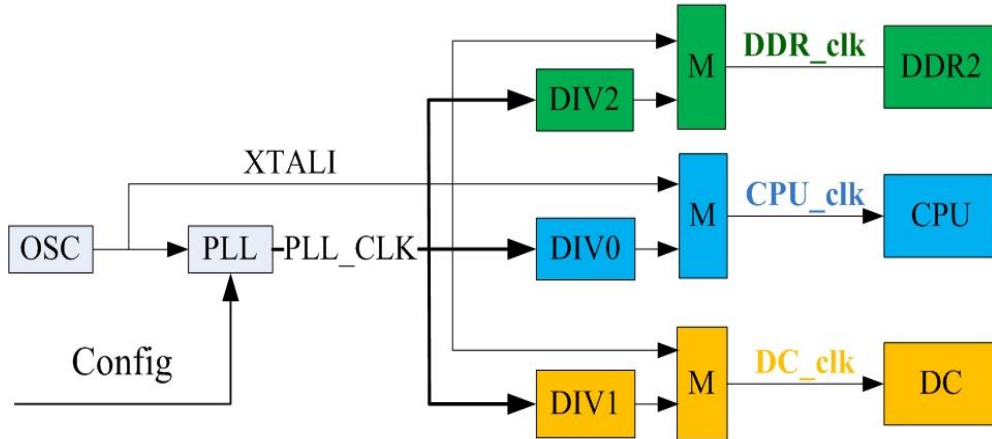


图 22-1 1B 时钟产生模块

### 21.2 Clock 配置描述

在系统 RESET 状态下，时钟模块利用外部 PAD 状态选择 PLL 的配置。具体如下表所示：

表 22-1

PAD	描述	对 PLL 配置作用
NAND_D[5:0]	PLL 输出频率控制	PLL 输出频率： $(NAND\_D[5:0]+12)*33/2\text{Mhz}$
NAND_D6	CPU 和 DDR 频率通路选择，系统复位启动过程中，CPU 和 DDR 频率相同。	CPU_clk 和 DDR2_clk 1: 33Mhz 0: $(NAND\_D[5:0]+12)*33/4\text{Mhz}$
NAND_D7	USB 参考时钟选择	1: 12MHz 时钟，接到 USB0_XO 0: 12MHz 晶体

系统启动后，软件可以精确配置 PLL 的频率，这些参数如下表所示：

PLL\_FREQ:基地址 0XBFE7\_8030:

Bit 位	寄存器描述
[17:0]	PLL 输出频率： $(12+PLL\_FREQ[5:0]+(PLL\_FREQ[17:8]/1024))*33/2\text{Mhz}$

同时，软件可以对内部的 CPU/DDR/DC 时钟频率单独配置。配置过程：对应时钟 Bypass 位置 1，让其切换到 33Mhz 的外部输入，然后让对应逻辑 RST，最后配置需要

分频的倍数，以产生目标时钟。恢复过程：先把 Bypass 位清零，让对应时钟恢复到分频的目标时钟。时钟频率的配置过程中使用了 Glitch free 的电路，确保系统能够正常稳定工作。对应寄存器详细描述如下：

PLL\_DIV\_PARAM: 基地址 0XBFE7\_8030:

Bit 位	寄存器描述	读写
31	DC_DIV 使能	R/W
30	DC_DIV 分频器复位	R/W
29:26	DC_DIV (pll_out/4/DC_DIV)	R/W
25	CPU_DIV 使能	R/W
24	CPU_DIV 分频器复位	R/W
23:20	CPU_DIV	R/W
19	DDR_DIV 使能	R/W
18	DDR_DIV 分频器复位	R/W
17:14	DDR_DIV	R/W
13	DC_BYPASS 使能	R/W
12	DC_BYPASS	R/W
11	DDR_BYPASS 使能	R/W
10	DDR_BYPASS	R/W
9	CPU_BYPASS 使能	R/W
8	CPU_BYPASS	R/W
7: 6	保留	保留
5	DC_RST 使能	R/W
4	DC_RST	R/W
3	DDR_RST 使能	R/W
2	DDR_RST	R/W
1	CPU_RST 使能	R/W
0	CPU_RST	R/W

### 21.3 系统其它 Clock 描述

在 1B 中，SPI、I2C、PWM、CAN、WATCHDOG、UART 等模块工作都需要时钟，这些时钟用来实现计数或确定分频系数。这些时钟工作在相同频率，器频率均为 DDR\_clk 频率的一半。具体分配系数确定和计数确定参考具体章节的描述。



## 22 GPIO and MUX

### 22.1 GPIO 结构描述

GPIO 给芯片的设计和应用提供了灵活外部接口；部分 PAD 通过 MUX 实现，从而在 BGA256 封装情况下提供丰富的外部功能。GPIO 作为输入高电平时候外部可以是 3.3V-5V，输入低电平是 0V；输出高电平是 3.3V，输出低电平是 0V；GPIO 对应的所有 PAD 都是推挽式，驱动能力为 8mA（约 50ohm 内阻）。

PAD	复位状态	PAD 描述	GPIO	第一复用	第二复用	第三复用	UART
PWM0	内部上拉，复位输入	PWM0 波形输出	GPIO00	NAND_RDY*	SPI1_CSN[1]	UART0_RX	
PWM1	内部上拉，复位输入	PWM1 波形输出	GPIO01	NAND_CS*	SPI1_CSN[2]	UART0_TX	
PWM2	内部上拉，复位输入	PWM2 波形输出	GPIO02	NAND_RDY*		UART0_CTS	
PWM3	内部上拉，复位输入	PWM3 波形输出	GPIO03	NAND_CS*		UART0_RTS	
LCD_CLK	内部上拉，复位输入	LCD 时钟	GPIO04				
LCD_VSYNC	内部上拉，复位输入	LCD 列同步	GPIO05				
LCD_HSYNC	内部上拉，复位输入	LCD 行同步	GPIO06				
LCD_EN	内部上拉，复位输入	LCD 使能信号	GPIO07				
LCD_DAT_B0	内部上拉，复位输入	LCD_BLUE0	GPIO08			UART1_RX	
LCD_DAT_B1	内部上拉，复位输入	LCD_BLUE1	GPIO09				
LCD_DAT_B2	内部上拉，复位输入	LCD_BLUE2	GPIO10				
LCD_DAT_B3	内部上拉，复位输入	LCD_BLUE3	GPIO11				
LCD_DAT_B4	内部上拉，复位输入	LCD_BLUE4	GPIO12				
LCD_DAT_G0	内部上拉，复位输入	LCD_GREEN0	GPIO13			UART1_CTS	
LCD_DAT_G1	内部上拉，复位输入	LCD_GREEN1	GPIO14			UART1_RTS	
LCD_DAT_G2	内部上拉，复位输入	LCD_GREEN2	GPIO15				

LCD_DAT_G3	内部上拉, 复位输入	LCD_GREEN3	GPIO16				
LCD_DAT_G4	内部上拉, 复位输入	LCD_GREEN4	GPIO17				
LCD_DAT_G5	内部上拉, 复位输入	LCD_GREEN5	GPIO18				
LCD_DAT_R0	内部上拉, 复位输入	LCD_RED0	GPIO19			UART1_TX	UART1_TX
LCD_DAT_R1	内部上拉, 复位输入	LCD_RED1	GPIO20				
LCD_DAT_R2	内部上拉, 复位输入	LCD_RED2	GPIO21				
LCD_DAT_R3	内部上拉, 复位输入	LCD_RED3	GPIO22				
LCD_DAT_R4	内部上拉, 复位输入	LCD_RED4	GPIO23				
SPI0_CLK	启动配置	SPI0 时钟	GPIO24				
SPI0_MISO	启动配置	SPI0 主入从出	GPIO25				
SPI0_MOSI	启动配置	SPI0 主出从入	GPIO26				
SPI0_CS0	启动配置	SPI0 选通信号 0	GPIO27				
SPI0_CS1	内部上拉, 复位输入	SPI0 选通信号 1	GPIO28				
SPI0_CS2	内部上拉, 复位输入	SPI0 选通信号 2	GPIO29				
SPI0_CS3	内部上拉, 复位输入	SPI0 选通信号 3	GPIO30				
SCL	内部无上拉, 复位输入	第一路 I2C 时钟	GPIO32				
SDA	内部无上拉, 复位输入	第一路 I2C 数据	GPIO33				
AC97_SYNC	内部上拉, 复位输入	AC97 同步信号	GPIO34				
AC97_RST	内部上拉, 复位输入	AC97 复位信号	GPIO35				
AC97_DI	内部上拉, 复位输入	AC97 数据输入	GPIO36				
AC97_DO	内部上拉, 复位输入	AC97 数据输出	GPIO37				
CAN0_RX	内部上拉, 复位输入	CAN0 数据输入	GPIO38	SDA1	SPI1_CSN0	Uart1_DSR	UART1_2RX
CAN0_TX	内部上拉, 复位输入	CAN0 数据输出	GPIO39	SCL1	SPI1_CLK	Uart1_DTR	UART1_2TX
CAN1_RX	内部上拉, 复位输入	CAN1 数据输入	GPIO40	SDA2	SPI1_MOSI	Uart1_DCD	UART1_3RX
CAN1_TX	内部上拉, 复位输入	CAN1 数据输出	GPIO41	SCL2	SPI1_MISO	Uart1_RI	UART1_3TX

UART0_RX	内部上拉, 复位输入	UART0 发送数据	GPIO42	LCD_DAT22	GMAC1_RCTL		UART0_ORX
UART0_TX	内部上拉, 复位输入	UART0 接收数据	GPIO43	LCD_DAT23	GMAC1_RX0		UART0_OTX
UART0_RTS	内部上拉, 复位输入	UART0 请求发送	GPIO44	LCD_DAT16	GMAC1_RX1		UART0_1TX
UART0_CTS	内部上拉, 复位输入	UART0 允许发送	GPIO45	LCD_DAT17	GMAC1_RX2		UART0_1RX
UART0_DSR	内部上拉, 复位输入	UART0 设备准备好	GPIO46	LCD_DAT18	GMAC1_RX3		UART0_2RX
UART0_DTR	内部上拉, 复位输入	UART0 终端准备好	GPIO47	LCD_DAT19			UART0_2TX
UART0_DCD	内部上拉, 复位输入	UART0 载波检测	GPIO48	LCD_DAT20	GMAC1_MDCK		UART0_3RX
UART0_RI	内部上拉, 复位输入	UART0 振铃提示	GPIO49	LCD_DAT21	GMAC1_MDIO		UART0_3TX
UART1_RX	内部上拉, 复位输入	UART1 接收数据	GPIO50		GMAC1_TX0	NAND_RDY*	UART1_ORX
UART1_TX	内部上拉, 复位输入	UART1 发送数据	GPIO51		GMAC1_TX1	NAND_CS*	UART1_OTX
UART1_RTS	内部上拉, 复位输入	UART1 请求发送	GPIO52		GMAC1_TX2	NAND_CS*	UART1_1TX
UART1_CTS	内部上拉, 复位输入	UART1 允许发送	GPIO53		GMAC1_TX3	NAND_RDY*	UART1_1RX
UART2_RX	内部上拉, 复位输入	UART2 接收数据	GPIO54				UART2_RX
UART2_TX	内部上拉, 复位输入	UART2 发送数据	GPIO55				UART2_TX
UART3_RX	内部上拉, 复位输入	UART3 接收数据	GPIO56				UART3_RX
UART3_TX	内部上拉, 复位输入	UART3 发送数据	GPIO57				UART3_TX
UART4_RX	内部上拉, 复位输入	UART4 接收数据	GPIO58				UART4_RX
UART4_TX	内部上拉, 复位输入	UART4 发送数据	GPIO59				UART4_TX
UART5_RX	内部上拉, 复位输入	UART5 接收数据	GPIO60				UART5_RX
UART5_TX	内部上拉, 复位输入	UART5 发送数据	GPIO61				UART5_TX

**Note1:** NAND\_CS\*/NAND\_RDY\*表示 NAND\_CS1/2/3, NAND\_RDY1/2/3 可以配置选择。

龙芯 1B 中 UART0 有 8 个 PAD, UART1 有 4 个 PAD, UART2/3/4/5 各 2 个, UART 本身 PAD 共 20 个, UART 一共可以提供了 10 个两线 UART (上表最后一列红色+黑色)。

UART0 和 UART1 都实现了一分四功能, UART0 有 8 个 PAD; UART1 只有 4 个 PAD, 这个时候利用了 CAN0 和 CAN1 的 4 个 PAD。所以在 CAN0/CAN1 不用的时候, 1B 最多可以提供出来 12 个两线 UART (上表最后一列红色+黑色+绿色)。

当 GMAC1 复用 UART1&UART0 的时候, UART0 复用 PWM 实现两路 2 线串口 (上表橙色部分); UART1 复用 CAN0/ CAN1/ LCD(0.5.6.11)实现全功能串口 (上表粉色部分); UART2/3/4/5 维持不变。在 GMAC1 复用的时候, 1B 还可以有 1 个全功能和 6 个两线串口, 或者 10 个两线串口。

## 22.2 GPIO 寄存器描述

偏移地址	位	寄存器	描述	读写	描述
0xbf010C0	32	GPIOCFG0	配置寄存器 0	R/W	GPIOCFG0[30:0] 分别对应 GPIO30:GPIO0 1:对应 PAD 为 GPIO 功能 0:对应 PAD 为普通功能 复位值: 32'hf0ffff
0xbf010C4	32	GPIOCFG1	配置寄存器 1	R/W	GPIOCFG1[29:0] 分别对应 GPIO61:GPIO32 1:对应 PAD 为 GPIO 功能 0:对应 PAD 为普通功能 复位值: 32'hffffff
0xbf010D0	32	GPIOOE0	输入使能寄存器 0	R/W	GPIOOE0[30:0] 分别对应 GPIO30:GPIO0 1:对应 GPIO 被控制为输入 0:对应 GPIO 被控制为输出 复位值: 32'hf0ffff
0xbf010D4	32	GPIOOE1	输入使能寄存器 1	R/W	GPIOOE1[29:0] 分别对应 GPIO61:GPIO32 1:对应 GPIO 被控制为输入 0:对应 GPIO 被控制为输出 复位值: 32'hffffff
0xbf010E0	32	GPIOIN0	输入寄存器 0	R	GPIOIN0[30:0] 分别对应 GPIO30:GPIO0 1: GPIO 输入值 1; PAD 驱动输入为 3.3V 0: GPIO 输入值 0; PAD 驱动输入为 0V
0xbf010E4	32	GPIOIN1	输入寄存器 1	R	GPIOIN1[29:0] 分别对应 GPIO61:GPIO32 1: GPIO 输入值 1; PAD 驱动输入为 3.3V 0: GPIO 输入值 0; PAD 驱动输入为 0V
0xbf010F0	32	GPIOOUT0	配置输出寄存器 0	R/W	GPIOOUT0[30:0] 分别对应 GPIO30:GPIO0 1: GPIO 输出值 1, PAD 驱动输出 3.3V

					0: GPIO 输出值 0, PAD 驱动输出 0V
0xbf010F4	32	GPIOOUT1	配置输出寄存器 1	R/W	GPIOOUT1[29:0] 分别对应 GPIO61:GPIO32 1: GPIO 输出为 1, PAD 驱动输出 3.3V 0: GPIO 输出为 0, PAD 驱动输出 0V

## 22.3 MUX 寄存器描述

GPIO\_MUX\_CTRL0 基地址 0XBFD0\_0420,寄存器的描述如下, 当管脚配置为 GPIO 功能时, MUX 寄存器的配置不起作用:

位	描述	读写特性
31: 26		保留
28	UART0_UAE_PWM23	R/W
27	UART0_USE_PWM01	R/W
26	UART1_USE_LCD0_5_6_11	R/W
25	I2C2_USE_CAN1	R/W
24	I2C1_USE_CAN0	R/W
23	NAND3_USE_UART5	R/W
22	NAND3_USE_UART4	R/W
21	NAND3_USE_UART1_DAT	R/W
20	NAND3_USE_UART1_CTS	R/W
19	NAND3_USE_PWM23	R/W
18	NAND3_USE_PWM01	R/W
17	NAND2_USE_UART5	R/W
16	NAND2_USE_UART4	R/W
15	NAND2_USE_UART1_DAT	R/W
14	NAND2_USE_UART1_CTS	R/W
13	NAND2_USE_PWM23	R/W
12	NAND2_USE_PWM01	R/W
11	NAND1_USE_UART5	R/W
10	NAND1_USE_UART4	R/W
9	NAND1_USE_UART1_DAT	R/W
8	NAND1_USE_UART1_CTS	R/W
7	NAND1_USE_PWM23	R/W
6	NAND1_USE_PWM01	R/W
5		保留
4	GMAC1_USE_UART1	R/W
3	GMAC1_USE_UART0	R/W
2	LCD_USE_UART0_DAT	R/W
1	LCD_USE_UART15	R/W
0	LCD_USE_UART0	R/W

GPIO\_MUX\_CTRL1 基地址 0XBFD0\_0424,寄存器的描述如下, 当管脚配置为 GPIO 功能时, MUX 寄存器的配置不起作用:

位	描述	读写特性
---	----	------

31	USB_reset	R/W
30:25		保留
24	SPI1_CS_USE_PWM01	R/W
23	SPI1_USE_CAN	R/W
22:21		保留
20	DISABLE_DDR_CONFSPACE	
19:17		保留
16	DDR32TO16EN	
15:14		保留
13	GMAC1_SHUT	
12	GMAC0_SHUT	R/W
11	USB_SHUT	R/W
10:6		保留
5	UART1_3_USE_CAN1	R/W
4	UART1_2_USE_CAN0	R/W
3	GMAC1_USE_TX_CLK	
2	GMAC0_USE_TX_CLK	
1	GMAC1_USE_PWM23	
0	GMAC0_USE_PWM01	

## 23 电气特性

### 23.1 电源域

1B 芯片内部主要包括四个电源域，如下表所示

表 23-1 1B 电源域

电源域	描述
Core	内部主要功能模块的供电。
RTC	系统断电时候，由外部电池供电；系统工作情况下，供电由外部电路切换到普通电源。 RTC 电源电流约 100uA，在断电时间较长的场合建议使用外部 RTC。
DDR	DDR2 接口工作所需电源，不支持休眠到内存。
PAD	普通 3.3V 接口所需电源

### 23.2 系统复位

1B 芯片在系统上电复位时，需要将某些芯片引脚上下拉进行配置。如下表格所示的全部引脚：

表 23-2 1B 上电配置引脚汇总

配置引脚	上下拉	功能
EJTAG_TRST	下拉	TAP 复位
EJTAG_TCK	上拉	TAP 时钟
EJTAG_TDI	上拉	TAP 数据输入
EJTAG_TMS	上拉	TAP 工作模式
SYS_RSTN	上拉	系统复位
NAND_D[5:0]	内部下拉 板上根据需	控制 PLL 输出频率 $(\text{NAND\_D}[5:0]+12)*33/2\text{Mhz}$
NAND_D6	要上下拉	CPU 和 DDR 频率通路选择，系统复位启动过程中，CPU 和 DDR 频率相同 CPU_clk 和 DDR2_clk 1: 33Mhz 0: $(\text{NAND\_D}[5:0]+12)*33/4\text{Mhz}$
NAND_D7		USB 参考时钟选择 1: 12MHz 时钟，接到 USB0_XO 0: 12MHz 晶体

### 23.3 推荐的工作条件

表 23-3 推荐的工作条件

参数	描述	范围			容差	最大电流
		Min.	Typ.	Max.		
VDD_1V2	核电源电压		1.25V		+/-50mV	500mA
PLL_DVDD12	PLL 数字电源		1.25V		+/-50mV	10mA
USB_DVDD1V2	USB 数字电源		1.25V		+/-50mV	20mA
VDD_1V8	DDR 电源电压		1.8V		+/-50mV	500mA
VREF_0V9	DDR 参考电压		0.9V		+/-25mV	
VDD_3V3	IO 电源电压		3.3V		+/-50mV	500mA
PLL_AVDD33	PLL 模拟电源		3.3V		+/-50mV	10mA
USB_AVDD33	USB 模拟电源		3.3V		+/-50mV	40mA
RTC_VDD33	RTC 电源	2.70V	3.3V	3.60V	+/-50mV	100uA
工作温度	商业级	0°C		70°C		
	工业级	-40°C		85°C		

### 23.4 绝对最大额定值

表 23-4 绝对最大额定值

参数	描述	范围	
		Min.	Max.
VDD_1V2	核电源电压	-0.3V	1.4V
PLL_DVDD12	PLL 数字电源	-0.3V	1.4V
USB_DVDD1V2	USB 数字电源	-0.3V	1.4V
VDD_1V8	DDR 电源电压	-0.3V	2.3V
VREF_0V9	DDR 参考电压	-0.3V	2.3V
VDD_3V3	IO 电源电压	-0.3V	4V
PLL_AVDD33	PLL 模拟电源	-0.3V	4V
USB_AVDD33	USB 模拟电源	-0.3V	4V
RTC_VDD33	RTC 电源	-0.3V	4V
存储工作温度	商业级	-10°C	65°C
	工业级	-50°C	95°C
功耗			1.8W

### 23.5 上电过程要求

1B 芯片在系统上电时，对上电次序没有强制性要求。建议先上 1V2，后上 3V3。



## 23.6 参考时钟

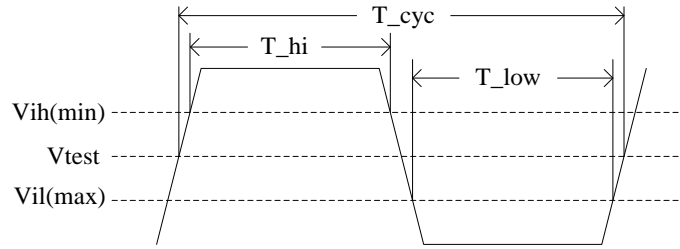


表 23-5 单端参考时钟波形

时钟	参数	描述	最小	最大	单位
XTALI (外接晶振时)	$V_{ih}$	输入高电平电压	2.0	-	V
	$V_{il}$	输入低电平电压	-	0.7	V
	$T_{cyc}$	时钟周期	30	-	ns
	$T_{hi}$	高电平时间	$40\%T_{cyc}$	$60\%T_{cyc}$	ns
	$T_{low}$	低电平时间	$40\%T_{cyc}$	$60\%T_{cyc}$	ns
	$T_{slew}$	斜率	1	4	V/ns

## 23.7 GPIO DC 特性

表 23-6 GPIO DC 特性

参数	描述	最小	最大	单位
$V_{ih}$	输入高电平电压	1.7	3.6	V
$V_{il}$	输入低电平电压	-0.3	0.7	V
$V_{oh}$	输出高电平电压	2.4	-	V
$V_{ol}$	输出低电平电压	-	0.4	V

## 24 热特性

### 24.1 焊接温度

表 24-1 回流焊接温度要求

Profile Feature		Pb-Free Assembly
Average ramp-up rate (T <sub>smax</sub> to T <sub>p</sub> )		3°C/second max.
Preheat	Temperature Min (T <sub>smin</sub> )	150 °C
	Temperature Max (T <sub>smax</sub> )	200 °C
	Time (T <sub>smin</sub> to T <sub>smax</sub> ) (t <sub>s</sub> )	60-180 seconds
Time maintained above	Temperature (T <sub>L</sub> )	217 °C
	Time (t <sub>L</sub> )	60-150 seconds
Peak Temperature (T <sub>p</sub> )		260°C
Time within 5°C of actual Peak Temperature (t <sub>p</sub> ) <sup>2</sup>		20-40 seconds
Ramp-down Rate		6 °C/second max.
Time 25°C to Peak Temperature		8 minutes max.

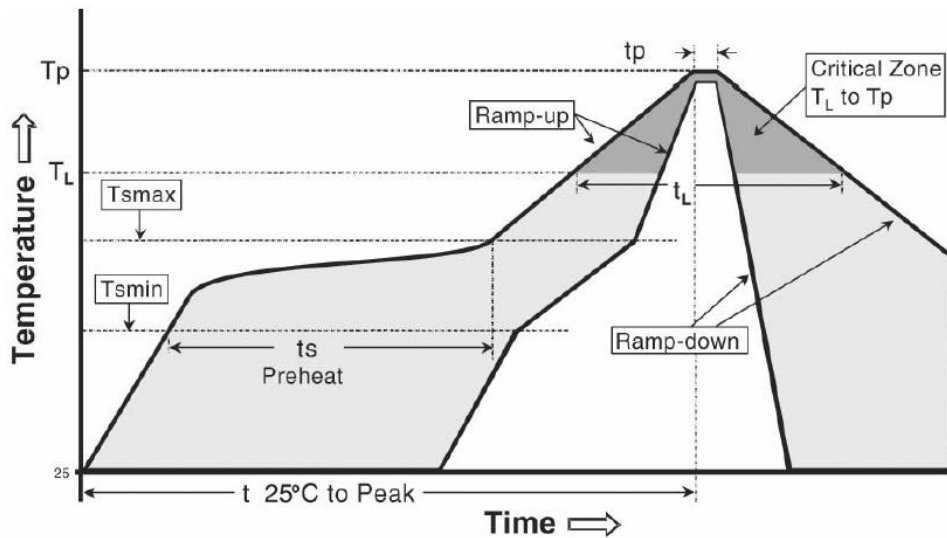


图 24-1 焊接回流曲线