

S3C2440A

32-位 CMOS

微型控制器

用户手册

修订版本 1

第一章产品概述

-----马志晶译

S3C2440A

目录

第一章产品概述.....1-3

 特性.....1-3

 内部结构图.....1-7

 管脚分配.....1-9

 信号说明.....1-22

 s3c2440a特殊寄存器.....1-27

S3C2440A

1

产品概述

引言

三星公司推出的16/32位RISC微处理器S3C2440A，为手持设备和一般类型应用提供了低价格、低功耗、高性能小型微控制器的解决方案。

为了降低整体系统成本，S3C2440A 提供了一下丰富的内部设备

S3C2440A 采用了 ARM920t 的内核，0.13um 的 CMOS 标准宏单元和存储器单元。其低功耗，简单，优雅，且全静态设计特别适合于对成本和功率敏感型的应用。它采用了新的总线架构 Advanced Micro controller Bus Architecture (AMBA)。

S3C2440A 的杰出的特点是其核心处理器(CPU)，是一个由 Advanced RISC Machines 有限公司设计的 16/32 位 ARM920T 的 RISC 处理器。ARM920T 实现了 MMU，AMBA BUS 和 Harvard 高速缓冲体系结构。这一结构具有独立的 16KB 指令 Cache 和 16KB 数据 Cache。每个都是由具有 8 字长的行组成。通过提供一套完整的通用系统外设，S3C2440A 减少整体系统成本和无需配置额外的组件。

综合对芯片的功能描述，本手册将介绍 S3C2440A 集成的以下片上功能：

- 1.2V 内核供电，1.8V/2.5V/3.3V 存储器供电，3.3V 外部 I/O 供电 具备 16KB 的 I-Cache 和 16KB DCache/MMU 微处理器
- 外部存储控制器(SDRAM 控制和片选逻辑)
- LCD 控制器（最大支持 4K 色 STN 和 256K 色 TFT）提供 1 通道 LCD 专用 DMA。
- 4 通道 DMA 并有外部请求引脚。
- 3 通道 UART(IrDA 1.0, 64 字节 Tx FIFO, 和 64 字节 Rx FIFO)
- 2 通道 SPI
- 1 通道 IIC-BUS 接口（多主支持）
- 1 通道 IIS-BUS 音频编解码器接口
- AC'97 解码器接口
- 兼容 SD 主接口协议 1.0 版和 MMC 卡协议 2.11 兼容版。
- 2 端口 USB 主机/1 端口 USB 设备（1.1 版）
- 4 通道 PWM 定时器和 1 通道内部定时器 / 看门狗定时器
- 8 通道 10 比特 ADC 和触摸屏接口
- 具有日历功能的 RTC
- 相机接口（最大 4096 × 4096 像素的投入支持。2048 × 2048 像素的投入，支持缩放）
- 130 个通用 I/O 口和 24 通道外部中断源。
- 具有普通，慢速，空闲和掉电模式。
- 具有 PLL 片上时钟发生器

特性

体系结构

- 为手持设备和通用嵌入式应用提供片上集成系统解决方案。
- 16/32 位 RISC 体系结构和 ARM920T 内核强大的指令集

- 加强的ARM 体系结构MMU 用于支持WinCE,EPOC 32 和Linux.
- 指令高速存储缓冲器 (I-Cache), 数据高速存储缓冲器 (D-Cache), 写缓冲器和物理地址TAG RAM 减少主存带宽和响应性带来的影响.
- 采用ARM920T CPU 内核支持ARM 调试体系结构.
- 内部高级微控制总线 (AMBA) 体系结构(AMBA2.0, AHB/APB).

系统管理器

- 支持大/小端方式.
- 支持高速总线模式和异步总线模式.
- 寻址空间: 每bank 128M 字节 (总共1G 字节).
- 支持可编程的每bank 8/16/32 位数据总线带宽.
- 从bank 0 到bank 6 都采用固定的bank 起始寻址.
- bank7 具有可编程的bank 的起始地址和大小
- 8 个存储器bank:
 - 其中6 个适用于ROM,SRAM,和其他
 - 另外2 个适用于ROM/SRAM 和同步DRAM.
- 所有的存储器bank 都具有可编程的操作周期.
- 支持外部等待信号延长总线周期.
- 支持掉电时的SDRAM 自刷新模式.
- 支持各种型号的ROM 引导 (NOR/NAND Flash, EEPROM, 或其他).

NAND Flash 启动引导

- 支持从NAND flash 存储器的启动.
- 采用4KB 内部缓冲器进行启动引导.
- 支持启动之后NAND 存储器仍然作为外部存储器使用.
- 支持先进的 NAND flash

Cache 存储器

- 64 项全相连模式, 采用I-Cache(16KB)和D-Cache(16KB).
- 每行8 字长度, 其中每行带有一个有效为和两个dirty 位.
- 伪随机数或轮转循环替换算法位.
- 采用写穿式 (write-through) 或写回式 (write-back) cache 操作来更新主存储器.
- 写缓冲器可以保存16 个字的数据和4 个地址.

时钟和电源管理

- 片上MPLL 和UPLL:
 - 采用UPLL 产生操作USB 主机/设备的时钟
 - MPLL 产生最大400MHZ@ 1.3V操作MCU 所需要的时钟
- 通过软件可以有选择性的为每个功能模块提供时钟.
- 电源模式: 正常, 慢速, 空闲和掉电模式
 - 正常模式: 正常运行模式
 - 慢速模式: 不加PLL 的低时钟频率模式.
 - 空闲模式: 只停止CPU 的时钟;
 - 掉电模式: 所有外设和内核的电源都切断了;
- 可以通过EINT[15:0]或RTC 报警中断来从掉电模式中唤醒处理器

特点 (续)

中断控制器

- 60 个中断源 (1 个看门狗定时器, 5 个定时器, 9 个UARTs, 24 个外部中断, 4 个DMA,2

个RTC,2 个ADC,1 个IIC,2 个SPI,1 个SDI,2 个USB,1 个LCD,和1 个电池故障, 1个NAND 和 2 个Camera), 1 AC97音频

- 电平/边沿触发模式的外部中断源
- 可编程的边沿/电平触发极性
- 支持为紧急中断请求提供快速中断服务

具有脉冲带宽调制功能的定时器 (PWM)

- 4 通道16 位具有PWM 功能的定时器, 1 通道16 位内部定时器, 可基于DMA 或中断工作
- 可编程的占空比周期, 频率和极性
- 能产生死区
- 支持外部时钟源

RTC (实时时钟)

- 全面的时钟特性: 秒、分、时、日期, 星期, 月和年;
- 32.768KHz 工作
- 具有报警中断
- 具有节拍中断

通用I/O 端口

- 24 个外部中断端口
- 130 个多功能输入/输出端口

DMA 控制器

- 4 通道的DMA 控制器;
- 支持存储器到存储器, IO 到存储器, 存储器到IO 和IO 到IO 的传输s
- 采用触发传输模式来加快传输速率

LCD 控制器STN LCD 显示特性

- 支持3 种类型的STN LCD 显示屏: 4 位双扫描, 4 位单扫描, 8 位单扫描显示类型
- 支持单色模式、4 级、16 级灰度STN LCD、256 色和4096 色STN LCD
- 支持多种不同尺寸的液晶屏

-LCD 实际尺寸的典型值是: 640×480, 320×240, 160×160 及其他.

-最大虚拟屏幕大小是4M 字节.

-256 色模式下支持的最大虚拟屏是: 4096×1024, 2048×2048, 1024×4096 等

TFT 彩色显示屏

- 支持彩色TFT 的1, 2, 4 或8bbp(像素每位)调色显示
- 支持16, 24bbp 无调色真彩显示 TFT
- 在24bbp 模式下支持最大16M 色TFT
- lpc3600定时控制器, 为嵌入式lts350Q1-PD1/2 (SAMSUNG 3.5" Portrait/ 256kcolor/ Reflective a-Si TFT LCD)
- lpc3600定时控制器, 为嵌入式lts350Q1-PE1/2 (SAMSUNG 3.5" Portrait / 256Kcolor/ Transflective a-Si TFT LCD)

支持多种不同尺寸的液晶屏

-典型实屏尺寸: 640×480, 320×240, 160×160 及其他

-最大虚拟屏大小4M 字节.

-64K 色彩模式下最大的虚拟屏尺寸为2048×1024 及其他

UART

- 3 通道UART, 可以基于DMA 模式或中断模式工作

- 支持5 位, 6 位, 7 位或者8 位串行数据发送/接收
- 支持外部时钟作为UART 的运行时钟 (UEXTCLK)
- 可编程的波特率
- 支持IrDA1.0
- 具有测试用的还回模式
- 每个通道都具有内部64 字节的发送FIFO 和64 字节的接收FIFO.

特点 (续)

A/D 转换和触摸屏接口

- 8 通道多路复用ADC
- 最大500KSPS/10 位精度
- 内部TFT 直接触摸屏接口

看门狗定时器

- 16 位看门狗定时器
- 在定时器溢出时发生中断请求或系统复位

IIC 总线接口

- 1 通道多主IIC 总线
- 可进行串行, 8 位, 双向数据传输, 标准模式下数据传输速度可达100kbit/s, 快速模式下可达到400kbit/s.

IIS 总线接口

- 1 通道音频IIS 总线接口, 可基于DMA 方式工作
- 串行, 每通道8/16 位数据传输
- 发送和接收具备128 字节(64 字节加64 字节)FIFO
- 支持IIS 格式和MSB-justified 数据格式

AC97 音频解码器接口

- 支援16位采样
- 1-ch 立体声 PCM 输入/ 1-ch 立体声 PCM 输出1-ch MIC 输入

USB 主设备

- 2 个 USB 主设备接口
- 遵从OHCI Rev.1.0 标准
- 遵从OHCI Rev.1.0 标准

USB 从设备

- 1 个USB 从设备接口
- 具备5 个Endpoint
- 兼容USB ver1.1 标准

SD 主机接口

- 正常, 中断和dma数据传输模式 (字节, 半字节, 文字传递)
- DMA burst4接入支持 (只字转让)
- 兼容SD 存储卡协议1.0 版
- 兼容SDIO 卡协议1.0 版
- 发送和接收具有64字节FIFO
- 兼容MMC 卡协议2.11 版

SPI 接口

- 兼容2 通道SPI 协议2.11 版

- 发送和接收具有2×8 位的移位寄存器
- 可以基于DMA 或中断模式工作

相机接口

- 支持 ITU-R BT 601/656 8-bit 模式
- 具有DZI (数字变焦) 能力
- 具有极性可编程视频同步信号
- 最大值支持. 4096 x 4096 像素输入(支持 2048 × 2048像素输入缩放)
- 镜头旋转 (x轴, y轴, 和180 °旋转)
- 相机输出格式 (16/24-bit的RGB与YCBCR 4:2:0/4:2:2格式)

工作电压

- 内核: 300MHz 时 1.20V
400MHz 时 1.3V

内存: 支持1.8v / 2.5v/3.0v/3.3v

- 输入/输出: 3.3v

操作频率

- Fclk 最高达 400MHz
- Hclk 最高达 136MHz
- Pclk 最高达 68MHz

封装

- 289-FBGA

内部结构图

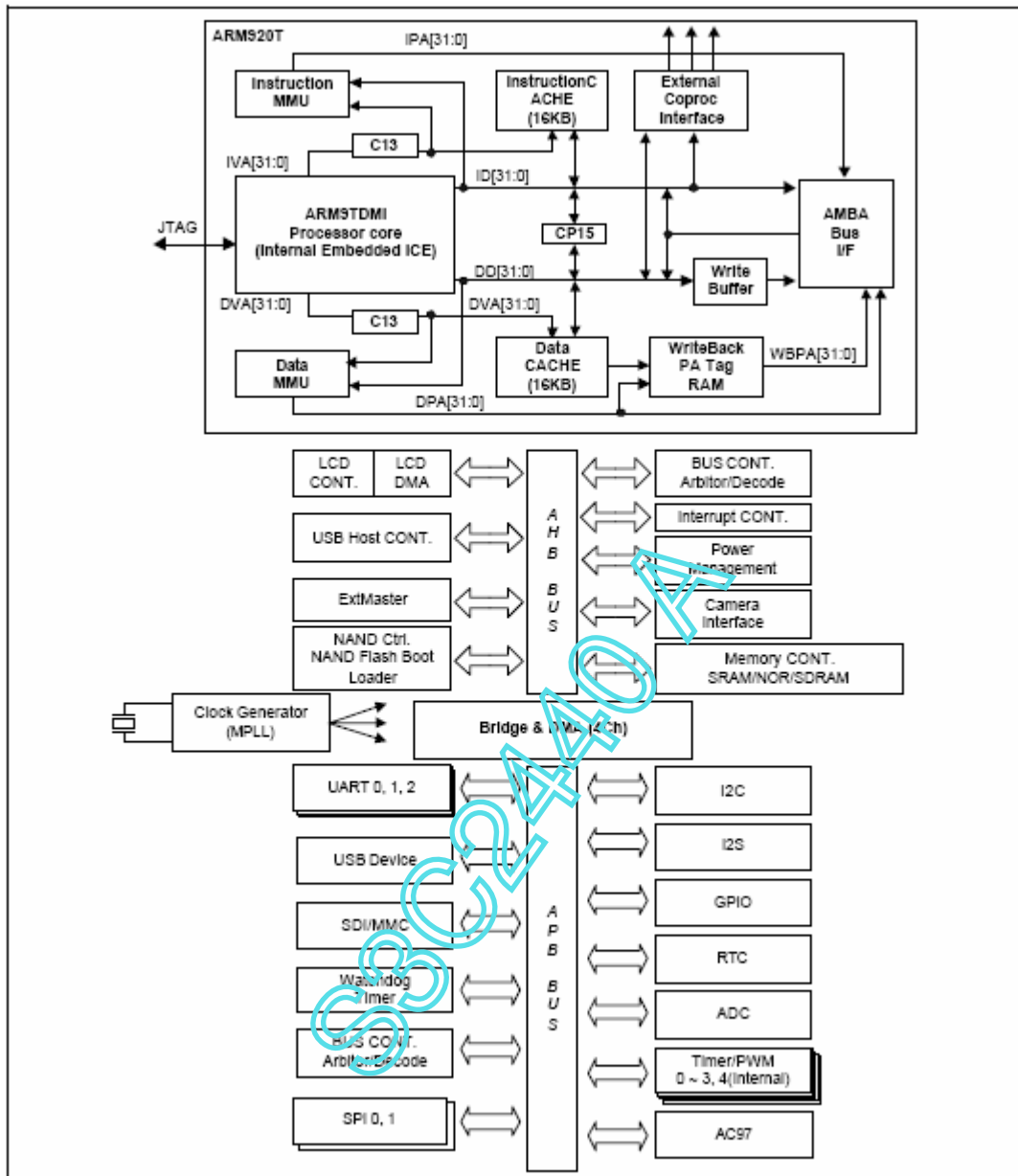


Figure 1-1. S3C2440A Block Diagram

图 1-1. S3C2440A 方框图

管脚分配

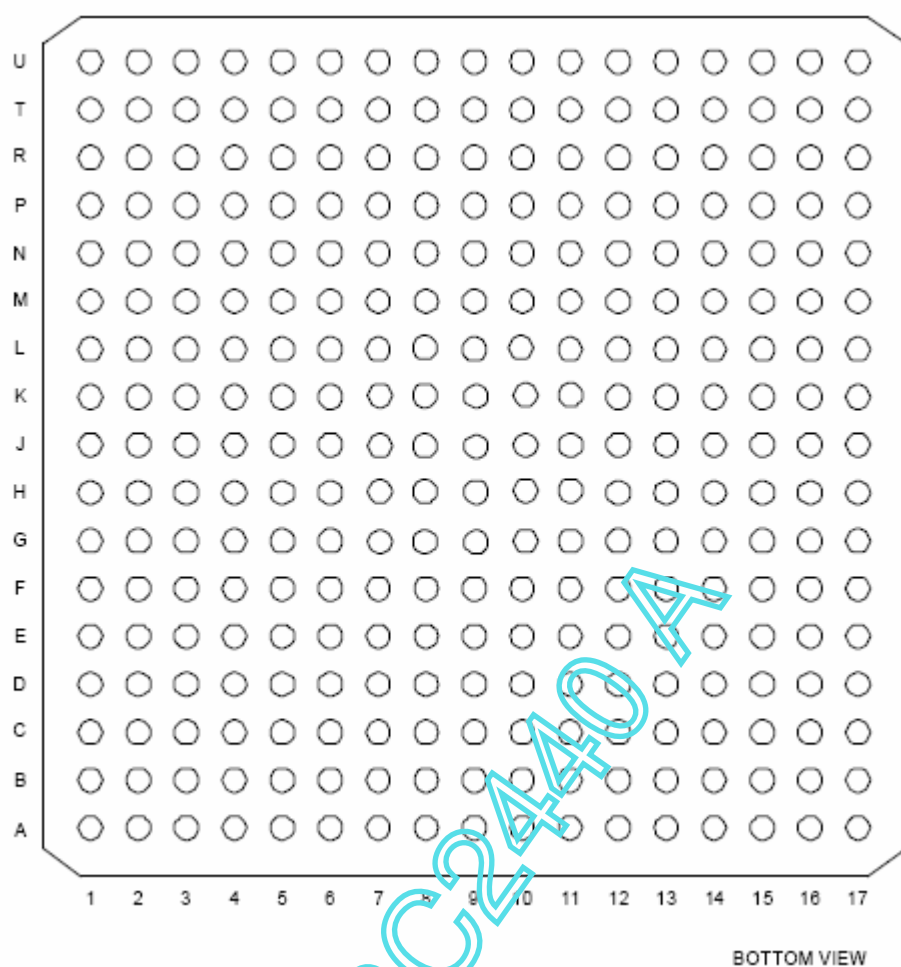


图1-2. S3C2440A引脚定义图(289-FBGA)

表 1-1 。 289 针脚 fbga 的管脚分配-密码命令 (表 1 3)

Pin Number	Pin Name	Pin Number	Pin Name	Pin Number	Pin Name
A1	VDDi	C1	VDDMOP	E1	nFRE/GPA20
A2	SCKE	C2	nGCS5/GPA18	E2	VSSMOP
A3	VSSi	C3	nGCS2/GPA13	E3	nGCS7
A4	VSSi	C4	nGCS3/GPA14	E4	nWAIT
A5	VSSMOP	C5	nOE	E5	nBE3
A6	VDDi	C6	nSRAS	E6	nWE
A7	VSSMOP	C7	ADDR4	E7	ADDR1
A8	ADDR10	C8	ADDR11	E8	ADDR6
A9	VDDMOP	C9	ADDR15	E9	ADDR14
A10	VDDi	C10	ADDR21/GPA6	E10	ADDR23/GPA8
A11	VSSMOP	C11	ADDR24/GPA9	E11	DATA2
A12	VSSi	C12	DATA1	E12	DATA20
A13	DATA3	C13	DATA6	E13	DATA19
A14	DATA7	C14	DATA11	E14	DATA18
A15	VSSMOP	C15	DATA13	E15	DATA17
A16	VDDi	C16	DATA16	E16	DATA21
A17	DATA10	C17	VSSi	E17	DATA24
B1	VSSMOP	D1	ALE/GPA18	F1	VDDi
B2	nGCS1/GPA12	D2	nGCS6	F2	VSSi
B3	SCLK1	D3	nGCS4/GPA15	F3	nFWE/GPA19
B4	SCLK0	D4	nBE0	F4	nFCE/GPA22
B5	nBE1	D5	nBE2	F5	CLE/GPA17
B6	VDDMOP	D6	nSCAS	F6	nGCS0
B7	ADDR2	D7	ADDR7	F7	ADDR0/GPA0
B8	ADDR9	D8	ADDR5	F8	ADDR3
B9	ADDR12	D9	ADDR16/GPA1	F9	ADDR18/GPA3
B10	VSSi	D10	ADDR20/GPA5	F10	DATA4
B11	VDDi	D11	ADDR26/GPA11	F11	DATA5
B12	VDDMOP	D12	DATA0	F12	DATA27
B13	VSSMOP	D13	DATA8	F13	DATA31
B14	VDDMOP	D14	DATA14	F14	DATA26
B15	DATA9	D15	DATA12	F15	DATA22
B16	VDDMOP	D16	VSSMOP	F16	VDDi
B17	DATA15	D17	VSSMOP	F17	VDDMOP

表 1-1 。 289 引脚 fpga 的管脚分配-密码令 (第 2 页共 3 页) (续)

Pin Number	Pin Name	Pin Number	Pin Name	Pin Number	Pin Name
G1	VSSOP	J1	VDDOP	L1	LEND/GPC0
G2	CAMHREF/GPJ10	J2	VDDIarm	L2	VDDIarm
G3	CAMDATA1/GPJ1	J3	CAMCLKOUT/GPJ11	L3	nXDACK0/GPB9
G4	VDDalive	J4	CAMRESET/GPJ12	L4	VCLK/GPC1
G5	CAMPCLK/GPJ8	J5	TOUT1/GPB1	L5	nXBREQ/GPB6
G6	FRnB	J6	TOUT0/GPB0	L6	VD1/GPC9
G7	CAMVSYNC/GPJ9	J7	TOUT2/GPB2	L7	VFRAME/GPC3
G8	ADDR8	J8	CAMDATA8/GPJ6	L8	I2SSDI/AC_SDATA_IN
G9	ADDR17/GPA2	J9	SDDAT3/GPE10	L9	SPICLK0/GPE13
G10	ADDR25/GPA10	J10	EINT10/nSS0/GPG2	L10	EINT15/SPICLK1/GPG7
G11	DATA28	J11	TXD2/nRTS1/GPH6	L11	EINT22/GPG14
G12	DATA25	J12	PWREN	L12	Xtortc
G13	DATA23	J13	TCK	L13	EINT2/GPF2
G14	XTIpl	J14	TMS	L14	EINT5/GPF5
G15	XTOpl	J15	RXD2/nCTS1/GPH7	L15	EINT6/GPF6
G16	DATA29	J16	TDO	L16	EINT7/GPF7
G17	VSSi	J17	VDDalive	L17	nRTS0/GPH1
H1	VSSIarm	K1	VSSIarm	M1	VLINE/GPC2
H2	CAMDATA7/GPJ7	K2	nBATT_KI/GPB5	M2	LCD_LPCREV/GPC6
H3	CAMDATA4/GPJ4	K3	TOUT3/GPB3	M3	LCD_LPCOE/GPC5
H4	CAMDATA3/GPJ3	K4	TCLD0/GPB4	M4	VM/GPC4
H5	CAMDATA2/GPJ2	K5	nXDREQ1/GPB8	M5	VD9/GPD1
H6	CAMDATA0/GPJ0	K6	nXDREQ0/GPB10	M6	VD8/GPC14
H7	CAMDATA5/GPJ5	K7	nXDACK1/GPB7	M7	VD16/SPIMISO1/GPD8
H8	ADDR13	K8	SDCMD/GPE6	M8	SDDAT1/GPE8
H9	ADDR19/GPA4	K9	SPIMISO0/GPE11	M9	IICSDA/GPE15
H10	ADDR22/GPA7	K10	EINT13/SPIMISO1/GPG5	M10	EINT20/GPG12
H11	VSSOP	K11	nCTS0/GPH0	M11	EINT17/nRTS1/GPG9
H12	EXTCLK	K12	VDDOP	M12	VSSA_UPLL
H13	DATA30	K13	TXD0/GPH2	M13	VDDA_UPLL
H14	nBATT_FLT	K14	RXD0/GPH3	M14	Xtirtc
H15	nTRST	K15	UEXTCLK/GPH8	M15	EINT3/GPF3
H16	nRESET	K16	TXD1/GPH4	M16	EINT1/GPF1
H17	TDI	K17	RXD1/GPH5	M17	EINT4/GPF4

表 1-1 。 289 针脚 fpga 的管脚分配-密码顺序 (表 3) (续)

Pin Number	Pin Name	Pin Number	Pin Name	Pin Number	Pin Name
N1	VSSOP	P15	AIN3	T12	VDDOP
N2	VD0/GPC8	P16	XP/AIN7	T13	OM3
N3	VD4/GPC12	P17	UPLLCAP	T14	VSSA_ADC
N4	VD2/GPC10	R1	VD3/GPC11	T15	OM0
N5	VD10/GPD2	R2	VD8/GPD0	T16	YM/AIN4
N6	VD15/GPD7	R3	VD11/GPD3	T17	YP/AIN5
N7	VD22/nSS1/GPD14	R4	VD13/GPD5	U1	VDDiarm
N8	SDCLK/GPE5	R5	VD18/SPICLK1/GPD10	U2	VDDiarm
N9	EINT8/GPG0	R6	VD21 /GPD13	U3	VSSOP
N10	EINT18/nCTS1/GPG10	R7	I2SSCLK/AC_BIT_CLK	U4	VSSiarm
N11	DP0	R8	SDDAT0/GPE7	U5	VD23/nSS0/GPD15
N12	DN1/PDN0	R9	CLKOUT0/GPH9	U6	I2SSDO/AC_SDATA_OUT
N13	nRSTOUT/GPA21	R10	EINT11/nSS1/GPG3	U7	VSSiarm
N14	MPLLCAP	R11	EINT14/SPIMOSI1/GPG6	U8	IICSCS/GPE14
N15	VDD_RTC	R12	NCON	U9	VSSOP
N16	VDDA_MPLL	R13	OM1	U10	VSSiarm
N17	EINT0/GPF0	R14	AIN0	U11	VDDi
P1	LCD_LPCREVB/GPC7	R15	AIN2	U12	EINT19/TCLK1/GPG11
P2	VD5/GPC13	R16	XM/AIN6	U13	EINT23/GPG15
P3	VD7/GPC15	R17	VSSA_VPLL	U14	DP1/PDP0
P4	VD12/GPD4	T1	VSSiarm	U15	VSSOP
P5	VD14/GPD6	T2	VSSiarm	U16	Vref
P6	VD20/GPD12	T3	VDDOP	U17	AIN1
P7	I2SLRCK/AC_SYNC	T4	VD17/SPIMOSI1/GPD9		
P8	SDDAT2/GPE9	T5	VD19/GPD11		
P9	SPIMOSI0/GPE12	T6	VDDiarm		
P10	CLKOUT1/GPH10	T7	CDCLK/AC_nRESET		
P11	EINT12/LCD_PWREN/GPG4	T8	VDDiarm		
P12	DN0	T9	EINT9/GPG1		
P13	OM2	T10	EINT16/GPG8		
P14	VDDA_ADC	T11	EINT21/GPG13		

表 1-2 。 s3c2440a 289 引脚 fbga 的管脚分配 (表 1/9)

Pin Number	Pin Name	Default Function	I/O State @BUS REQ	I/O State @Sleep	I/O State @nRESET	I/O Type
F7	ADDR0/GPA0	ADDR0	Hi-z/-	O(L)/-	O(L)	t10s
E7	ADDR1	ADDR1	Hi-z	O(L)	O(L)	t10s
B7	ADDR2	ADDR2	Hi-z	O(L)	O(L)	t10s
F8	ADDR3	ADDR3	Hi-z	O(L)	O(L)	t10s
C7	ADDR4	ADDR4	Hi-z	O(L)	O(L)	t10s
D8	ADDR5	ADDR5	Hi-z	O(L)	O(L)	t10s
E8	ADDR6	ADDR6	Hi-z	O(L)	O(L)	t10s
D7	ADDR7	ADDR7	Hi-z	O(L)	O(L)	t10s
G8	ADDR8	ADDR8	Hi-z	O(L)	O(L)	t10s
B8	ADDR9	ADDR9	Hi-z	O(L)	O(L)	t10s
A8	ADDR10	ADDR10	Hi-z	O(L)	O(L)	t10s
C8	ADDR11	ADDR11	Hi-z	O(L)	O(L)	t10s
B9	ADDR12	ADDR12	Hi-z	O(L)	O(L)	t10s
H8	ADDR13	ADDR13	Hi-z	O(L)	O(L)	t10s
E9	ADDR14	ADDR14	Hi-z	O(L)	O(L)	t10s
C9	ADDR15	ADDR15	Hi-z	O(L)	O(L)	t10s
D9	ADDR16/GPA1	ADDR16	Hi-z/-	O(L)/-	O(L)	t10s
G9	ADDR17/GPA2	ADDR17	Hi-z/-	O(L)/-	O(L)	t10s
F9	ADDR18/GPA3	ADDR18	Hi-z/-	O(L)/-	O(L)	t10s
H9	ADDR19/GPA4	ADDR19	Hi-z/-	O(L)/-	O(L)	t10s
D10	ADDR20/GPA5	ADDR20	Hi-z/-	O(L)/-	O(L)	t10s
C10	ADDR21/GPA6	ADDR21	Hi-z/-	O(L)/-	O(L)	t10s
H10	ADDR22/GPA7	ADDR22	Hi-z/-	O(L)/-	O(L)	t10s
E10	ADDR23/GPA8	ADDR23	Hi-z/-	O(L)/-	O(L)	t10s
C11	ADDR24/GPA9	ADDR24	Hi-z/-	O(L)/-	O(L)	t10s
G10	ADDR25/GPA10	ADDR25	Hi-z/-	O(L)/-	O(L)	t10s
D11	ADDR26/GPA11	ADDR26	Hi-z/-	O(L)/-	O(L)	t10s
R14	AIN0	AIN0	-	-	AI	r10
U17	AIN1	AIN1	-	-	AI	r10
R15	AIN2	AIN2	-	-	AI	r10
P15	AIN3	AIN3	-	-	AI	r10
T16	YM/AIN4	AIN4	-/-	-/-	AI	r10
T17	YP/AIN5	YP	-/-	-/-	AI	r10
R16	XM/AIN6	AIN6	-/-	-/-	AI	r10

表 1-2 。 s3c2440a 289 引脚 fbga 的管脚分配 (表 2/9) (续))

Pin Number	Pin Name	Default Function	I/O State @BUS REQ	I/O State @Sleep	I/O State @nRESET	I/O Type
P16	XP/AIN7	XP	-/-	-/-	AI	r10
H6	CAMDATA0/GPJ0	GPJ0	-/-	Hi-z/-	I	t8
G3	CAMDATA1/GPJ1	GPJ1	-/-	Hi-z/-	I	t8
H5	CAMDATA2/GPJ2	GPJ2	-/-	Hi-z/-	I	t8
H4	CAMDATA3/GPJ3	GPJ3	-/-	Hi-z/-	I	t8
H3	CAMDATA4/GPJ4	GPJ4	-/-	Hi-z/-	I	t8
H7	CAMDATA5/GPJ5	GPJ5	-/-	Hi-z/-	I	t8
J8	CAMDATA6/GPJ6	GPJ6	-/-	Hi-z/-	I	t8
H2	CAMDATA7/GPJ7	GPJ7	-/-	Hi-z/-	I	t8
G5	CAMPCLK/GPJ8	GPJ8	-/-	Hi-z/-	I	t8
G7	CAMVSYNC/GPJ9	GPJ9	-/-	Hi-z/-	I	t8
G2	CAMHREF/GPJ10	GPJ10	-/-	Hi-z/-	I	t8
J3	CAMCLKOUT/GPJ11	GPJ11	-/-	Hi-z/-	I	t8
J4	CAMRESET/GPJ12	GPJ12	-/-	Hi-z/-	I	t8
D12	DATA0	DATA0	Hi-z	Hi-z,O(L)	I	b12s
C12	DATA1	DATA1	Hi-z	Hi-z,O(L)	I	b12s
E11	DATA2	DATA2	Hi-z	Hi-z,O(L)	I	b12s
A13	DATA3	DATA3	Hi-z	Hi-z,O(L)	I	b12s
F10	DATA4	DATA4	Hi-z	Hi-z,O(L)	I	b12s
F11	DATA5	DATA5	Hi-z	Hi-z,O(L)	I	b12s
C13	DATA6	DATA6	Hi-z	Hi-z,O(L)	I	b12s
A14	DATA7	DATA7	Hi-z	Hi-z,O(L)	I	b12s
D13	DATA8	DATA8	Hi-z	Hi-z,O(L)	I	b12s
B15	DATA9	DATA9	Hi-z	Hi-z,O(L)	I	b12s
A17	DATA10	DATA10	Hi-z	Hi-z,O(L)	I	b12s
C14	DATA11	DATA11	Hi-z	Hi-z,O(L)	I	b12s
D15	DATA12	DATA12	Hi-z	Hi-z,O(L)	I	b12s
C15	DATA13	DATA13	Hi-z	Hi-z,O(L)	I	b12s
D14	DATA14	DATA14	Hi-z	Hi-z,O(L)	I	b12s
B17	DATA15	DATA15	Hi-z	Hi-z,O(L)	I	b12s
C16	DATA16	DATA16	Hi-z	Hi-z,O(L)	I	b12s
E15	DATA17	DATA17	Hi-z	Hi-z,O(L)	I	b12s
E14	DATA18	DATA18	Hi-z	Hi-z,O(L)	I	b12s

表 1-2 。 s3c2440a 289 引脚 fbga 的管脚分配 (表 3/9) (续))

Pin Number	Pin Name	Default Function	I/O State @BUS REQ	I/O State @Sleep	I/O State @nRESET	I/O Type
E13	DATA19	DATA19	Hi-z	Hi-z,O(L)	I	b12s
E12	DATA20	DATA20	Hi-z	Hi-z,O(L)	I	b12s
E16	DATA21	DATA21	Hi-z	Hi-z,O(L)	I	b12s
F15	DATA22	DATA22	Hi-z	Hi-z,O(L)	I	b12s
G13	DATA23	DATA23	Hi-z	Hi-z,O(L)	I	b12s
E17	DATA24	DATA24	Hi-z	Hi-z,O(L)	I	b12s
G12	DATA25	DATA25	Hi-z	Hi-z,O(L)	I	b12s
F14	DATA26	DATA26	Hi-z	Hi-z,O(L)	I	b12s
F12	DATA27	DATA27	Hi-z	Hi-z,O(L)	I	b12s
G11	DATA28	DATA28	Hi-z	Hi-z,O(L)	I	b12s
G16	DATA29	DATA29	Hi-z	Hi-z,O(L)	I	b12s
H13	DATA30	DATA30	Hi-z	Hi-z,O(L)	I	b12s
F13	DATA31	DATA31	Hi-z	Hi-z,O(L)	I	b12s
P12	DN0	DN0	-	-	AI	us
N11	DP0	DP0	-	-	AI	us
N12	DN1/PDN0	DN1	-/-	-	AI	us
U14	DP1/PDP0	DP1	-/-	-	AI	us
N17	EINT0/GPF0	GPF0	-/-	Hi-z/-	I	t8
M16	EINT1/GPF1	GPF1	-/-	Hi-z/-	I	t8
L13	EINT2/GPF2	GPF2	-/-	Hi-z/-	I	t8
M15	EINT3/GPF3	GPF3	-/-	Hi-z/-	I	t8
M17	EINT4/GPF4	GPF4	-/-	Hi-z/-	I	t8
L14	EINT5/GPF5	GPF5	-/-	Hi-z/-	I	t8
L15	EINT6/GPF6	GPF6	-/-	Hi-z/-	I	t8
L16	EINT7/GPF7	GPF7	-/-	Hi-z/-	I	t8
N9	EINT8/GPG0	GPG0	-/-	Hi-z/-	I	t8
T9	EINT9/GPG1	GPG1	-/-	Hi-z/-	I	t8
J10	EINT10/nSS0/GPG2	GPG2	-/-/-	Hi-z/Hi-z/-	I	t8
R10	EINT11/nSS1/GPG3	GPG3	-/-/-	Hi-z/Hi-z/-	I	t8
P11	EINT12/LCD_PWREN/GPG4	GPG4	-/-/-	Hi-z/O(L)/-	I	t8
K10	EINT13/SPIMISO1/GPG5	GPG5	-/-/-	Hi-z/Hi-z/-	I	t8
R11	EINT14/SPIMOSI1/GPG6	GPG6	-/-/-	Hi-z/Hi-z/-	I	t8
L10	EINT15/SPICLK1/GPG7	GPG7	-/-/-	Hi-z/Hi-z/-	I	t8

表 1-2 。 s3c2440a 289 引脚 fbga 的管脚分配 (表 4/9) (续))

Pin Number	Pin Name	Default Function	I/O State @BUS REQ	I/O State @Sleep	I/O State @nRESET	I/O Type
T10	EINT16/GPG8	GPG8	-/-	Hi-z/-	I	t8
M11	EINT17/nRTS1/GPG9	GPG9	-/-/-	Hi-z/O(H)/-	I	t8
N10	EINT18/nCTS1/GPG10	GPG10	-/-/-	Hi-z/Hi-z/-	I	t8
U12	EINT19/TCLK1/GPG11	GPG11	-/-/-	Hi-z/Hi-z/-	I	t12
M10	EINT20/GPG12	GPG12	-/-	Hi-z/-	I	t12
T11	EINT21/GPG13	GPG13	-/-	Hi-z/-	I	t12
L11	EINT22/GPG14	GPG14	-/-	Hi-z/-	I	t12
U13	EINT23/GPG15	GPG15	-/-	Hi-z/-	I	t12
H12	EXTCLK	EXTCLK	-	-	AI	is
P17	UPLLCAP	UPLLCAP	-	-	AI	r50
N14	MPLLCAP	MPLLCAP	-	-	AI	r50
H14	nBATT_FLT	nBATT_FLT	-	-	I	is
D4	nBE0	nBE0	Hi-z	Hi-z,O(H)	O(H)	t10s
B5	nBE1	nBE1	Hi-z	Hi-z,O(H)	O(H)	t10s
D5	nBE2	nBE2	Hi-z	Hi-z,O(H)	O(H)	t10s
E5	nBE3	nBE3	Hi-z	Hi-z,O(H)	O(H)	t10s
R12	NCON	NCON	-	-	I	is
G6	FRnB	FRnB	-	Hi-z,O(L)	I	d2s
F3	nFWE/GPA19	GPA19	O(H)/-	Hi-z,O(H)/-	O(H)	t10s
E1	nFRE/GPA20	GPA20	O(H)/-	Hi-z,O(H)/-	O(H)	t10s
F4	nFCE/GPA22	GPA22	O(H)/-	Hi-z,O(H)/-	O(H)	t10s
F5	CLE/GPA17	GPA17	O(L)/-	Hi-z,O(L)/-	O(L)	t10s
D1	ALE/GPA18	GPA18	O(L)/-	Hi-z,O(L)/-	O(L)	t10s
N13	nRSTOUT/GPA21	GPA21	-/-	O(L)/-	O(L)	b8
C5	nOE	nOE	Hi-z	Hi-z,O(H)	O(H)	t10s
H16	nRESET	nRESET	-	-	I	is
F8	nGCS0	nGCS0	Hi-z	Hi-z,O(H)	O(H)	t10s
B2	nGCS1/GPA12	GPA12	Hi-z/-	Hi-z,O(H)/-	O(H)	t10s
C3	nGCS2/GPA13	GPA13	Hi-z/-	Hi-z,O(H)/-	O(H)	t10s
C4	nGCS3/GPA14	GPA14	Hi-z/-	Hi-z,O(H)/-	O(H)	t10s
D3	nGCS4/GPA15	GPA15	Hi-z/-	Hi-z,O(H)/-	O(H)	t10s
C2	nGCS5/GPA16	GPA16	Hi-z/-	Hi-z,O(H)/-	O(H)	t10s
D2	nGCS6	nGCS6	Hi-z	Hi-z,O(H)	O(H)	t10s

表 1-2 。 s3c2440a 289 引脚 fpga 的管脚分配 (表 5/9) (续)

Pin Number	Pin Name	Default Function	I/O State @BUS REQ	I/O State @Sleep	I/O State @nRESET	I/O Type
E3	nGCS7	nGCS7	Hi-z	Hi-z,O(H)	O(H)	t10s
D8	nSCAS	nSCAS	Hi-z	Hi-z,O(H)	O(H)	t10s
C8	nSRAS	nSRAS	Hi-z	Hi-z,O(H)	O(H)	t10s
H15	nTRST	nTRST	I	-	I	is
E4	nWAIT	nWAIT	-	Hi-z,O(L)	I	d2s
E6	nWE	nWE	Hi-z	Hi-z,O(H)	O(H)	t10s
J6	TOUT0/GPB0	GPB0	-/-	O(L)/-	I	t8
J5	TOUT1/GPB1	GPB1	-/-	O(L)/-	I	t8
J7	TOUT2/GPB2	GPB2	-/-	O(L)/-	I	t8
K3	TOUT3/GPB3	GPB3	-/-	O(L)/-	I	t8
K4	TCLK0/GPB4	GPB4	-/-	-/-	I	t8
K2	nXBACK/GPB5	GPB5	-/-	O(H)/-	I	t8
L5	nXBREQ/GPB6	GPB6	-/-	-/-	I	t8
K7	nXDACK1/GPB7	GPB7	-/-	O(H)/-	I	t8
K5	nXDREQ1/GPB8	GPB8	-/-	-/-	I	t8
L3	nXDACK0/GPB9	GPB9	-/-	O(H)/-	I	t8
K6	nXDREQ0/GPB10	GPB10	-/-	-/-	I	t8
T15	OM0	OM0	-	-	I	is
R13	OM1	OM1	-	-	I	is
P13	OM2	OM2	-	-	I	is
T13	OM3	OM3	-	-	I	is
J12	PWREN	PWREN	O(H)	O(L)	O(H)	b8
K11	nCTS0/GPH0	GPH0	-/-	-/-	I	t8
L17	nRTS0/GPH1	GPH1	-/-	O(H)/-	I	t8
K13	TXD0/GPH2	GPH2	-/-	O(H)/-	I	t8
K14	RXD0/GPH3	GPH3	-/-	-/-	I	t8
K16	TXD1/GPH4	GPH4	-/-	O(H)/-	I	t8
K17	RXD1/GPH5	GPH5	-/-	-/-	I	t8
J11	TXD2/nRTS1/GPH6	GPH6	-/-/-	O(H)/O(H)/-	I	t8
J15	RXD2/nCTS1/GPH7	GPH7	-/-/-	Hi-z/Hi-z/-	I	t8
K15	UEXTCLK/GPH8	GPH8	-/-	Hi-z/-	I	t8
R9	CLKOUT0/GPH9	GPH9	-/-	O(L)/-	I	t12
P10	CLKOUT1/GPH10	GPH10	-/-	O(L)/-	I	t12

表 1-2 。 s3c2440a 289 引脚 fpga 的管脚分配 (表 6/9) (续))

Pin Number	Pin Name	Default Function	I/O State @BUS REQ	I/O State @Sleep	I/O State @nRESET	I/O Type
A2	SCKE	SCKE	Hi-z	O(L)	O(H)	t10s
B4	SCLK0	SCLK0	Hi-z	O(L)	O(SCLK)	t12s
B3	SCLK1	SCLK1	Hi-z	O(L)	O(SCLK)	t12s
P7	I2SLRCK/AC_SYNC	GPE0	-/-	Hi-z/-	I	t8
R7	I2SSCLK/AC_BIT_CLK	GPE1	-/-	Hi-z/-	I	t8
T7	CDCLK/AC_nRESET	GPE2	-/-	Hi-z/-	I	t8
L8	I2SSDI/AC_SDATA_IN	GPE3	-/-/-	Hi-z/Hi-z/-	I	t8
U8	I2SSDO/AC_SDATA_OUT	GPE4	-/-/-	O(L)/Hi-z/-	I	t8
N8	SDCLK/GPE5	GPE5	-/-	O(L)/-	I	t8
K8	SDCMD/GPE6	GPE6	-/-	Hi-z/-	I	t8
R8	SDDAT0/GPE7	GPE7	-/-	Hi-z/-	I	t8
M8	SDDAT1/GPE8	GPE8	-/-	Hi-z/-	I	t8
P8	SDDAT2/GPE9	GPE9	-/-	Hi-z/-	I	t8
J9	SDDAT3/GPE10	GPE10	-/-	Hi-z/-	I	t8
K9	SPIMISO0/GPE11	GPE11	-/-	Hi-z/-	I	t8
P9	SPIMOSI0/GPE12	GPE12	-/-	Hi-z/-	I	t8
L9	SPICLK0/GPE13	GPE13	-/-	Hi-z/-	I	t8
U8	IICSL/GPE14	GPE14	-/-	Hi-z/-	I	d8
M9	IICSDA/GPE15	GPE15	-/-	Hi-z/-	I	d8
J13	TCK	TCK	-	-	I	is
H17	TDI	TDI	-	-	I	is
J16	TDO	TDO	O	O	O	ot
J14	TMS	TMS	I	-	I	is
L1	LEND/GPC0	GPC0	-/-	O(L)/-	I	t8
L4	VCLK/GPC1	GPC1	-/-	O(L)/-	I	t8
M1	VLINE/GPC2	GPC2	-/-	O(L)/-	I	t8
L7	VFRAME/GPC3	GPC3	-/-	O(L)/-	I	t8
M4	VM/GPC4	GPC4	-/-	O(L)/-	I	t8
M3	LCD_LPCOE/GPC5	GPC5	-/-	O(L)/-	I	t8
M2	LCD_LPCREV/GPC6	GPC6	-/-	O(L)/-	I	t8
P1	LCD_LPCREVB/GPC7	GPC7	-/-	O(L)/-	I	t8
N2	VD0/GPC8	GPC8	-/-	O(L)/-	I	t8
L6	VD1/GPC9	GPC9	-/-	O(L)/-	I	t8

表 1-2 。 s3c2440a 289 引脚 fpga 的管脚分配 (表 7/9) (续)

Pin Number	Pin Name	Default Function	I/O State @BUS REQ	I/O State @Sleep	I/O State @nRESET	I/O Type
N4	VD2/GPC10	GPC10	-/-	O(L)/-	I	t8
R1	VD3/GPC11	GPC11	-/-	O(L)/-	I	t8
N3	VD4/GPC12	GPC12	-/-	O(L)/-	I	t8
P2	VD5/GPC13	GPC13	-/-	O(L)/-	I	t8
M6	VD6/GPC14	GPC14	-/-	O(L)/-	I	t8
P3	VD7/GPC15	GPC15	-/-	O(L)/-	I	t8
R2	VD8/GPD0	GPD0	-/-	O(L)/-	I	t8
M5	VD9/GPD1	GPD1	-/-	O(L)/-	I	t8
N5	VD10/GPD2	GPD2	-/-	O(L)/-	I	t8
R3	VD11/GPD3	GPD3	-/-	O(L)/-	I	t8
P4	VD12/GPD4	GPD4	-/-	O(L)/-	I	t8
R4	VD13/ GPD5	GPD5	-/-/-	O(L)/O/-	I	t8
P5	VD14/GPD6	GPD6	-/-/-	O(L)/O/-	I	t8
N6	VD15/GPD7	GPD7	-/-/-	O(L)/O/-	I	t8
M7	VD16/SPIMISO1/GPD8	GPD8	-/-/-	O(L)/Hi-z/-	I	t8
T4	VD17/SPIMOSI1/GPD9	GPD9	-/-/-	O(L)/Hi-z/-	I	t8
R5	VD18/SPICLK1/GPD10	GPD10	-/-/-	O(L)/Hi-z/-	I	t8
T5	VD19//GPD11	GPD11	-/-/-	O(L)/Hi-z/-	I	t8
P6	VD20/ GPD12	GPD12	-/-/-	O(L)/Hi-z/-	I	t8
R6	VD21/ GPD13	GPD13	-/-/-	O(L)/Hi-z/-	I	t8
N7	VD22/nSS1/GPD14	GPD14	-/-/-	O(L)/Hi-z/-	I	t8
U5	VD23/nSS0/GPD15	GPD15	-/-/-	O(L)/Hi-z/-	I	t8
U16	Vref	Vref	-	-	AI	ia
G14	XTIpll	XTIpll	-	-	AI	m26
M14	Xtirtc	Xtirtc	-	-	AI	nc
G15	XTOpll	XTOpll	-	-	AO	m26
L12	Xtortc	Xtortc	-	-	AO	nc
N15	VDD_RTC	VDD_RTC	P	P	P	drtc
P14	VDDA_ADC	VDDA_ADC	P	P	P	d33th
N16	VDDA_MPLL	VDDA_MPLL	P	P	P	d12t
M13	VDDA_UPLL	VDDA_UPLL	P	P	P	d12t
G4	VDDalive	VDDalive	P	P	P	d12i
J17	VDDalive	VDDalive	P	P	P	d12i

表 1-2 。 s3c2440a 289 引脚 fpga 的管脚分配 (表 8/9) (续))

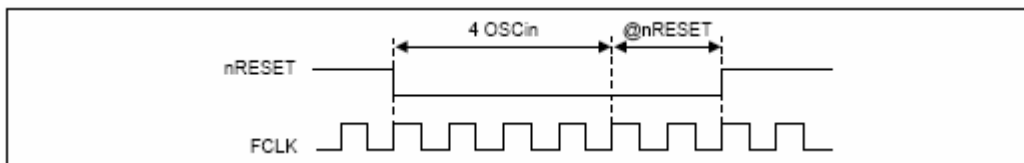
Pin Number	Pin Name	Default Function	I/O State @BUS REQ	I/O State @Sleep	I/O State @nRESET	I/O Type
A1	VDDi	VDDi	P	P	P	d12c
A10	VDDi	VDDi	P	P	P	d12c
A16	VDDi	VDDi	P	P	P	d12c
A6	VDDi	VDDi	P	P	P	d12c
B11	VDDi	VDDi	P	P	P	d12c
F1	VDDi	VDDi	P	P	P	d12c
F16	VDDi	VDDi	P	P	P	d12c
U11	VDDi	VDDi	P	P	P	d12c
L2	VDDiarm	VDDiarm	P	P	P	d12c
T6	VDDiarm	VDDiarm	P	P	P	d12c
T8	VDDiarm	VDDiarm	P	P	P	d12c
U1	VDDiarm	VDDiarm	P	P	P	d12c
J2	VDDiarm	VDDiarm	P	P	P	d12c
U2	VDDiarm	VDDiarm	P	P	P	d12c
A9	VDDMOP	VDDMOP	P	P	P	d33o
B12	VDDMOP	VDDMOP	P	P	P	d33o
B14	VDDMOP	VDDMOP	P	P	P	d33o
B16	VDDMOP	VDDMOP	P	P	P	d33o
B6	VDDMOP	VDDMOP	P	P	P	d33o
C1	VDDMOP	VDDMOP	P	P	P	d33o
F17	VDDMOP	VDDMOP	P	P	P	d33o
J1	VDDOP	VDDOP	P	P	P	d33o
T12	VDDOP	VDDOP	P	P	P	d33o
T3	VDDOP	VDDOP	P	P	P	d33o
K12	VDDOP	VDDOP	P	P	P	d33o
T14	VSSA_ADC	VSSA_ADC	P	P	P	sth
R17	VSSA_MPLL	VSSA_MPLL	P	P	P	st
M12	VSSA_UPLL	VSSA_UPLL	P	P	P	st
A12	VSSi	VSSi	P	P	P	si
A3	VSSi	VSSi	P	P	P	si
A4	VSSi	VSSi	P	P	P	si
B10	VSSi	VSSi	P	P	P	si
C17	VSSi	VSSi	P	P	P	si

表 1-2 。 s3c2440a 289 引脚 fpga 的管脚分配 (表 9/9) (续))

Pin Number	Pin Name	Default Function	I/O State @BUS REQ	I/O State @Sleep	I/O State @nRESET	I/O Type
F2	VSSi	VSSi	P	P	P	si
G17	VSSi	VSSi	P	P	P	si
H1	VSSiarm	VSSiarm	P	P	P	si
K1	VSSiarm	VSSiarm	P	P	P	si
T1	VSSiarm	VSSiarm	P	P	P	si
T2	VSSiarm	VSSiarm	P	P	P	si
U10	VSSiarm	VSSiarm	P	P	P	si
U4	VSSiarm	VSSiarm	P	P	P	si
U7	VSSiarm	VSSiarm	P	P	P	si
A11	VSSMOP	VSSMOP	P	P	P	so
A15	VSSMOP	VSSMOP	P	P	P	so
A5	VSSMOP	VSSMOP	P	P	P	so
A7	VSSMOP	VSSMOP	P	P	P	so
B1	VSSMOP	VSSMOP	P	P	P	so
B13	VSSMOP	VSSMOP	P	P	P	so
D16	VSSMOP	VSSMOP	P	P	P	so
D17	VSSMOP	VSSMOP	P	P	P	so
E2	VSSMOP	VSSMOP	P	P	P	so
G1	VSSOP	VSSOP	P	P	P	so
N1	VSSOP	VSSOP	P	P	P	so
U15	VSSOP	VSSOP	P	P	P	so
U3	VSSOP	VSSOP	P	P	P	so
U9	VSSOP	VSSOP	P	P	P	so
H11	VSSOP	VSSOP	P	P	P	so

注释:

1. @BUS REQ 表示处于外部总线控制状态, 即总线被其他总线所有者占用。
2. “-” 标记表示在BUS REQ 模式中引脚没有变化。
3. Hi-z 或Pre 表示高阻状态或保持之前的状态, 究竟是哪一个是哪一个由MISCCR 寄存器的设置决定。
4. AI/AO表示模拟输入/模拟输出。
5. P, I, 和 O分别表示电源, 输入和输出。
6. I/O 状态@nRESET 表示 I/O 脚在复位脚 nRESET 处于低电平下的状态, 如下图所示。



下表说明我 I/O 类型和描述。

Input (I)/Output (O) Type	Descriptions
d12i(vdd12ih)	1.2V V_{DD} for alive power
d12o(vdd12ih_core), si(vssih)	1.2V V_{DD}/V_{SS} for internal logic
d33o(vdd33oph), so(vssoph)	3.3V V_{DD}/V_{SS} for external logic
d33th(vdd33th_abb), sth(vssbbh_abb)	3.3V V_{DD}/V_{SS} for analog circuitry
d12t(vdd12t_abb), st(vssbb_abb)	1.2V V_{DD}/V_{SS} for analog circuitry
drto(vdd30th_rtc)	3.0V V_{DD} for RTC power
t8(phbsu100ct8sm)	Bi-directional pad, LVCMOS schmitt-trigger, 100k Ω pull-up resistor with control, tri-state, $I_o = 8mA$
is(phis)	Input pad, LVCMOS schmitt-trigger level
us(pbusb0)	USB pad
t10(phtot10cd)	5V tolerant output pad, tri-state.
ot(phot8)	Output pad, tri-state, $I_o = 8mA$
b8(phob8)	Output pad, $I_o = 8mA$
t16(phot16sm)	Output pad, tri-state, medium slew rate, $I_o = 16mA$
r10(phiar10_abb)	Analog input pad with 10 Ω resistor
ia(phiia_abb)	Analog input pad
gp(phgpad_option)	Pad for analog pin
m26(phsoscsm26_2440a)	Oscillator cell with enable and feedback resistor
t12(phbsu100ct12sm)	Bi-directional pad, LVCMOS schmitt-trigger, 100k Ω pull-up resistor with control, tri-state, $I_o = 12mA$
d8(phbsd8sm)	Bi-directional pad, LVCMOS schmitt-trigger, open drain, $I_o = 8mA$
t10s(phtot10cd_10_2440a)	output pad, LVCMOS, tri-state, output drive strenth control, $I_o = 4, 6, 8, 10mA$
b12s(phtbsu100ct12cd_12_2440a)	Bi-directional pad, LVCMOS schmitt-trigger, 100k Ω pull-up resistor with control, tri-state output drive strenth control, $I_o = 6, 8, 10, 12mA$
d2s(phtbsd2_2440a)	Bi-directional pad, LVCMOS schmitt-trigger, open-drain, output drive strenth control, $I_o = 2mA$
r50(phoar50_abb)	Analog output pad, 50k Ω resistor, separated bulk-bias
t12s(phtot12cd_12_2440a)	output pad, LVCMOS, tri-state, output drive strenth control, $I_o = 6, 8, 10, 12mA$
nc(phnc)	No connection pad

信号说明

表 1-3 。 s3c2440a 信号说明 (表 1/6)

Signal	Input/Output	Descriptions
Bus Controller		
OM[1:0]	I	OM[1:0] sets S3C2440A in the TEST mode, which is used only at fabrication. Also, it determines the bus width of nGCS0. The pull-up/down resistor determines the logic level during RESET cycle. 00: Nand-boot 01: 16-bit 10: 32-bit 11: Test mode
ADDR[26:0]	O	ADDR[26:0] (Address Bus) outputs the memory address of the corresponding bank .
DATA[31:0]	IO	DATA[31:0] (Data Bus) inputs data during memory read and outputs data during memory write. The bus width is programmable among 8/16/32-bit.
nGCS[7:0]	O	nGCS[7:0] (General Chip Select) are activated when the address of a memory is within the address region of each bank. The number of access cycles and the bank size can be programmed.
nWE	O	nWE (Write Enable) indicates that the current bus cycle is a write cycle.
nOE	O	nOE (Output Enable) indicates that the current bus cycle is a read cycle.
nXBREQ	I	nXBREQ (Bus Hold Request) allows another bus master to request control of the local bus. BACK active indicates that bus control has been granted.
nXBACK	O	nXBACK (Bus Hold Acknowledge) indicates that the S3C2440A has surrendered control of the local bus to another bus master.
nWAIT	I	nWAIT requests to prolong a current bus cycle. As long as nWAIT is L, the current bus cycle cannot be completed.
SDRAM/SRAM		
nSRAS	O	SDRAM row address strobe
nSCAS	O	SDRAM column address strobe
nSCS[1:0]	O	SDRAM chip select
DQM[3:0]	O	SDRAM data mask
SCLK[1:0]	O	SDRAM clock
SCKE	O	SDRAM clock enable
nBE[3:0]	O	Upper byte/lower byte enable (In case of 16-bit SRAM)
nWBE[3:0]	O	Write byte enable
NAND Flash		
CLE	O	Command latch enable
ALE	O	Address latch enable
nFCE	O	Nand flash chip enable
nFRE	O	Nand flash read enable
nFWE	O	Nand flash write enable
NCON	I	Nand flash configuration
FRnB	I	Nand flash ready/busy

* If NAND flash controller isn't used, it has to be pull-up. (VDDMOP)

表 1-3 。 s3c2440a 信号说明 (表 2/6)

Signal	Input/ Output	Descriptions
LCD Control Unit		
VD[23:0]	O	STN/TFT/SEC TFT: LCD data bus
LCD_PWREN	O	STN/TFT/SEC TFT: LCD panel power enable control signal
VCLK	O	STN/TFT: LCD clock signal
VFRAME	O	STN: LCD frame signal
VLINE	O	STN: LCD line signal
VM	O	STN: VM alternates the polarity of the row and column voltage
VSYNC	O	TFT: Vertical synchronous signal
HSYNC	O	TFT: Horizontal synchronous signal
VDEN	O	TFT: Data enable signal
LEND	O	TFT: Line end signal
STV	O	SEC TFT: SEC(Samsung Electronics Company) TFT LCD panel control signal
CPV	O	SEC TFT: SEC(Samsung Electronics Company) TFT LCD panel control signal
LCD_HCLK	O	SEC TFT: SEC(Samsung Electronics Company) TFT LCD panel control signal
TP	O	SEC TFT: SEC(Samsung Electronics Company) TFT LCD panel control signal
STH	O	SEC TFT: SEC(Samsung Electronics Company) TFT LCD panel control signal
LCD_LPCOE	O	SEC TFT: Timing control signal for specific TFT LCD
LCD_LPCREV	O	SEC TFT: Timing control signal for specific TFT LCD
LCD_LPCREVB	O	SEC TFT: Timing control signal for specific TFT LCD
CAMERA Interface		
CAMRESET	O	Software reset to the camera
CAMCLKOUT	O	Master clock to the camera
CAMPCLK	I	Pixel clock from camera
CAMHREF	I	Horizontal sync signal from camera
CAMVSYNC	I	Vertical sync signal from camera
CAMDATA[7:0]	I	Pixel data for Y/CbCr
Interrupt Control Unit		
EINT[23:0]	I	External interrupt request
DMA		
nXDREQ[1:0]	I	External DMA request
nXDACK[1:0]	O	External DMA acknowledge

表 1-3 。 s3c2440a 信号说明 (表 3/6)

Signal	Input/Output	Descriptions
UART		
RxD[2:0]	I	UART receives data input
TxD[2:0]	O	UART transmits data output
nCTS[1:0]	I	UART clear to send input signal
nRTS[1:0]	O	UART request to send output signal
UEXTCLK	I	External clock input for UART
ADC		
AIN[7:0]	AI	ADC input[7:0]. If it isn't used pin, it has to be low (ground).
Vref	AI	ADC Vref
IIC-Bus		
IICSDA	IO	IIC-bus data
IICSCL	IO	IIC-bus clock
IIS-Bus		
I2SLRCK	IO	IIS-bus channel select clock
I2SSDO	O	IIS-bus serial data output
I2SSDI	I	IIS-bus serial data input
I2SSCLK	IO	IIS-bus serial clock
CDCLK	O	CODEC system clock
AC'97		
AC_SYNC		48kHz fixed rate sample sync
AC_BIT_CLK	IO	12.288MHz serial data clock
AC_nRESET	O	AC'97 Master H/W Reset
AC_SDATA_IN	I	Serial, time division multiplexed, AC'97 input stream
AC_SDATA_OUT	O	Serial, time division multiplexed, AC'97 output stream
Touch Screen		
nXPON	O	Plus X-axis on-off control signal
XMON	O	Minus X-axis on-off control signal
nYPON	O	Plus Y-axis on-off control signal
YMON	O	Minus Y-axis on-off control signal
USB Host		
DN[1:0]	IO	DATA(-) from USB host. (Need to 15k Ω pull-down)
DP[1:0]	IO	DATA(+) from USB host. (Need to 15k Ω pull-down)
USB Device		
PDN0	IO	DATA(-) for USB peripheral. (Need to 470k Ω pull-down for power consumption in sleep mode)
PDP0	IO	DATA(+) for USB peripheral. (Need to 1.5k Ω pull-up)

表 1-3 。 s3c2440a 信号说明 (表 4/6)

Signal	Input/Output	Description
SPI		
SPIMISO[1:0]	IO	SPIMISO is the master data input line, when SPI is configured as a master. When SPI is configured as a slave, these pins reverse its role.
SPIMOSI[1:0]	IO	SPIMOSI is the master data output line, when SPI is configured as a master. When SPI is configured as a slave, these pins reverse its role.
SPICLK[1:0]	IO	SPI clock
nSS[1:0]	I	SPI chip select(only for slave mode)
SD		
SDDAT[3:0]	IO	SD receive/transmit data
SDCMD	IO	SD receive response/ transmit command
SDCLK	O	SD clock
General Port		
GPn[129:0]	IO	General input/output ports (some ports are output only)
TIMMER/PWM		
TOUT[3:0]	O	Timer output[3:0]
TCLK[1:0]	I	External timer clock input
JTAG TEST LOGIC		
nTRST	I	nTRST (TAP Controller Reset) resets the TAP controller at start. If debugger is used, A 10K pull-up resistor has to be connected. If debugger (black ICE) is not used, nTRST pin must be issued by a low active pulse (Typically connected to nRESET).
TMS	I	TMS (TAP Controller Mode Select) controls the sequence of the TAP controller's states. A 10K pull-up resistor has to be connected to TMS pin.
TCK	I	TCK (TAP Controller Clock) provides the clock input for the JTAG logic. A 10K pull-up resistor must be connected to TCK pin.
TDI	I	TDI (TAP Controller Data Input) is the serial input for test instructions and data. A 10K pull-up resistor must be connected to TDI pin.
TDO	O	TDO (TAP Controller Data Output) is the serial output for test instructions and data.

表 1-3 。 s3c2440a 信号说明 (表 5/6)

Signal	Input/Output	Description
Reset, Clock & Power		
XTOpll	AO	Crystal Output for internal osc circuit. When OM[3:2] = 00b, XTIpil is used for MPLL CLK source and UPLL CLK source. When OM[3:2] = 01b, XTIpil is used for MPLL CLK source only. When OM[3:2] = 10b, XTIpil is used for UPLL CLK source only. If it isn't used, it has to be a floating pin.
MPLLCAP	AI	Loop filter capacitor for main clock.
UPLLCAP	AI	Loop filter capacitor for USB clock.
XTIrto	AI	32 kHz crystal input for RTC. If it isn't used, it has to be High (VDDRTC).
XTOrtc	AO	32 kHz crystal output for RTC. If it isn't used, it has to be Float.
CLKOUT[1:0]	O	Clock output signal. The CLKSEL of MISCCR register configures the clock output mode among the MPLL CLK, UPLL CLK, FCLK, HCLK, PCLK.
nRESET	ST	nRESET suspends any operation in progress and places S3C2440A into a known reset state. For a reset, nRESET must be held to L level for at least 4 OSCin after the processor power has been stabilized.
nRSTOUT	O	For external device reset control (nRSTOUT = nRESET & nWDTRST & SW_RESET)
PWREN	O	1.2V/1.3V core power on-off control signal
nBATT_FLT	I	Probe for battery state(Does not wake up at Sleep mode in case of low battery state). If it isn't used, it has to be High (VDDOP).
OM[3:2]	I	OM[3:2] determines how the clock is made. OM[3:2] = 00b, Crystal is used for MPLL CLK source and UPLL CLK source. OM[3:2] = 01b, Crystal is used for MPLL CLK source and EXTCLK is used for UPLL CLK source. OM[3:2] = 10b, EXTCLK is used for MPLL CLK source and Crystal is used for UPLL CLK source. OM[3:2] = 11b, EXTCLK is used for MPLL CLK source and UPLL CLK source.
EXTCLK	I	External clock source. When OM[3:2] = 11b, EXTCLK is used for MPLL CLK source and UPLL CLK source. When OM[3:2] = 10b, EXTCLK is used for MPLL CLK source only. When OM[3:2] = 01b, EXTCLK is used for UPLL CLK source only. If it isn't used, it has to be High (VDDOP).
XTIpil	AI	Crystal Input for internal osc circuit. When OM[3:2] = 00b, XTIpil is used for MPLL CLK source and UPLL CLK source. When OM[3:2] = 01b, XTIpil is used for MPLL CLK source only. When OM[3:2] = 10b, XTIpil is used for UPLL CLK source only. If it isn't used, XTIpil has to be High (VDDOP).

表 1-3 。 s3c2440a 信号说明 (表 6/6)

Signal	Input/Output	Description
Power		
VDDalive	P	S3C2440A reset block and port status register V_{DD} . It should be always supplied whether in normal mode or in Sleep mode.
VDDiarm	P	S3C2440A core logic V_{DD} for ARM core.
VDDi	P	S3C2440A core logic V_{DD} for Internal block.
VSSi/VSSiarm	P	S3C2440A core logic V_{SS}
VDDi_MPLL	P	S3C2440A MPLL analog and digital V_{DD} .
VSSi_MPLL	P	S3C2440A MPLL analog and digital V_{SS} .
VDDOP	P	S3C2440A I/O port VDD (3.3V)
VDDMOP	P	S3C2440A memory I/O V_{DD} 3.3V: SCLK up to 135 MHz 2.5V: SCLK up to 135 MHz 1.8V: SCLK up to 93 MHz
VSSOP	P	S3C2440A I/O port VSS
RTCvDD	P	RTC V_{DD} (3.0V, Input range: 1.8 ~ 3.6V). This pin must be connected to power properly if RTC isn't used.
VDDi_UPLL	P	S3C2440A UPLL analog and digital V_{DD}
VSSi_UPLL	P	S3C2440A UPLL analog and digital V_{SS}
VDDA_ADC	P	S3C2440A ADC V_{DD} (3.3V)
VSSA_ADC	P	S3C2440A ADC V_{SS}

注释:

- 1 。 I/O: 输入/输出。
- 2 。 AI/AO: 表示模拟输入/模拟输出。
- 3 。 ST : 施密特触发。
- 4 。 P: 表示电源。

s3c2440a特殊寄存器

表 1-4 。 s3c2440a 特殊寄存器（表 1/14 ）

Register Name	Address (B. Endian)	Address (L. Endian)	Acc. Unit	Read/ Write	Function
Memory Controllers					
BWSCON	0x48000000		W	R/W	Bus width & wait status control
BANKCON0	0x48000004				Boot ROM control
BANKCON1	0x48000008				BANK1 control
BANKCON2	0x4800000C				BANK2 control
BANKCON3	0x48000010				BANK3 control
BANKCON4	0x48000014				BANK4 control
BANKCON5	0x48000018				BANK5 control
BANKCON6	0x4800001C				BANK6 control
BANKCON7	0x48000020				BANK7 control
REFRESH	0x48000024				DRAM/SDRAM refresh control
BANKSIZE	0x48000028				Flexible bank size
MRSRB6	0x4800002C				Mode register set for SDRAM BANK6
MRSRB7	0x48000030				Mode register set for SDRAM BANK7

表 1-4 。 s3c2440a 特殊寄存器 (表 2/14)

Register Name	Address (B. Endian)	Address (L. Endian)	Acc. Unit	Read/ Write	Function
USB Host Controller					
HcRevision	0x49000000	←	W		Control and status group
HcControl	0x49000004				
HcCommonStatus	0x49000008				
HcInterruptStatus	0x4900000C				
HcInterruptEnable	0x49000010				
HcInterruptDisable	0x49000014				
HcHCCA	0x49000018				Memory pointer group
HcPeriodCurrentED	0x4900001C				
HcControlHeadED	0x49000020				
HcControlCurrentED	0x49000024				
HcBulkHeadED	0x49000028				
HcBulkCurrentED	0x4900002C				
HcDoneHead	0x49000030				Frame counter group
HcRmInterval	0x49000034				
HcFmRemaining	0x49000038				
HcFmNumber	0x4900003C				
HcPeriodicStart	0x49000040				
HcLSThreshold	0x49000044				
HcRhDescriptorA	0x49000048				Root hub group
HcRhDescriptorB	0x4900004C				
HcRhStatus	0x49000050				
HcRhPortStatus1	0x49000054				
HcRhPortStatus2	0x49000058				
Interrupt Controller					
SRCPND	0X4A000000	←	W	R/W	Interrupt request status
INTMOD	0X4A000004			W	Interrupt mode control
INTMSK	0X4A000008			R/W	Interrupt mask control
PRIORITY	0X4A00000C			W	IRQ priority control
INTPND	0X4A000010			R/W	Interrupt request status
INTOFFSET	0X4A000014			R	Interrupt request source offset
SUBSRCPND	0X4A000018			R/W	Sub source pending
INTSUBMSK	0X4A00001C			R/W	Interrupt sub mask

表 1-4 。 s3c2440a 特殊寄存器 (表 3/14)

Register Name	Address (B. Endian)	Address (L. Endian)	Acc. Unit	Read/Write	Function
DMA					
DISRC0	0x4B000000	←	W	R/W	DMA 0 initial source
DISRCC0	0x4B000004				DMA 0 initial source control
DIDST0	0x4B000008				DMA 0 initial destination
DIDSTC0	0x4B00000C				DMA 0 initial destination control
DCON0	0x4B000010				DMA 0 control
DSTAT0	0x4B000014			R	DMA 0 count
DCSRC0	0x4B000018				DMA 0 current source
DCDST0	0x4B00001C				DMA 0 current destination
DMASKTRIG0	0x4B000020			R/W	DMA 0 mask trigger
DISRC1	0x4B000040				DMA 1 initial source
DISRCC1	0x4B000044				DMA 1 initial source control
DIDST1	0x4B000048				DMA 1 initial destination
DIDSTC1	0x4B00004C				DMA 1 initial destination control
DCON1	0x4B000050				DMA 1 control
DSTAT1	0x4B000054			R	DMA 1 count
DCSRC1	0x4B000058				DMA 1 current source
DCDST1	0x4B00005C				DMA 1 current destination
DMASKTRIG1	0x4B000060			R/W	DMA 1 mask trigger
DISRC2	0x4B000080				DMA 2 initial source
DISRCC2	0x4B000084				DMA 2 initial source control
DIDST2	0x4B000088				DMA 2 initial destination
DIDSTC2	0x4B00008C				DMA 2 initial destination control
DCON2	0x4B000090				DMA 2 control
DSTAT2	0x4B000094			R	DMA 2 count
DCSRC2	0x4B000098				DMA 2 current source
DCDST2	0x4B00009C				DMA 2 current destination
DMASKTRIG2	0x4B0000A0			R/W	DMA 2 mask trigger
DISRC3	0x4B0000C0	←	W	R/W	DMA 3 initial source
DISRCC3	0x4B0000C4				DMA 3 initial source control
DIDST3	0x4B0000C8				DMA 3 initial destination
DIDSTC3	0x4B0000CC				DMA 3 initial destination control
DCON3	0x4B0000D0				DMA 3 control
DSTAT3	0x4B0000D4			R	DMA 3 count
DCSRC3	0x4B0000D8				DMA 3 current source
DCDST3	0x4B0000DC				DMA 3 current destination
DMASKTRIG3	0x4B0000E0			R/W	DMA 3 mask trigger

表 1-4 。 s3c2440a 特殊寄存器 (表 4/14)

Register Name	Address (B. Endian)	Address (L. Endian)	Acc. Unit	Read/Write	Function
Clock & Power Management					
LOCKTIME	0x4C000000	←	W	R/W	PLL lock time counter
MPLLCON	0x4C000004				MPLL control
UPLLCON	0x4C000008				UPLL control
CLKCON	0x4C00000C				Clock generator control
CLKSLOW	0x4C000010				Slow clock control
CLKDIVN	0x4C000014				Clock divider control
CAMDIVN	0x4C000018				Camera clock divider control
LCD Controller					
LCDCON1	0x4D000000	←	W	R/W	LCD control 1
LCDCON2	0x4D000004				LCD control 2
LCDCON3	0x4D000008				LCD control 3
LCDCON4	0x4D00000C				LCD control 4
LCDCON5	0x4D000010				LCD control 5
LCDSADDR1	0x4D000014				STN/TFT: frame buffer start address 1
LCDSADDR2	0x4D000018				STN/TFT: frame buffer start address 2
LCDSADDR3	0x4D00001C				STN/TFT: virtual screen address set
REDLUT	0x4D000020				STN: red lookup table
GREENLUT	0x4D000024				STN: green lookup table
BLUELUT	0x4D000028				STN: blue lookup table
DITHMODE	0x4D00004C				STN: dithering mode
TPAL	0x4D000050				TFT: temporary palette
LCDINTPND	0x4D000054				LCD interrupt pending
LCDSRCPND	0x4D000058				LCD interrupt source
LCDINTMSK	0x4D00005C				LCD interrupt mask
TCONSEL	0x4D000060				TCON(LPC3600/LCC3600) control

表 1-4 。 s3c2440a 特殊寄存器（表 5/14 ）

Register Name	Address (B. Endian)	Address (L. Endian)	Acc. Unit	Read/Write	Function
NAND Flash					
NFCONF	0x4E000000	←	W	R/W	NAND flash configuration
NFCONT	0x4E000004				NAND flash control
NFCMD	0x4E000008				NAND flash command
NFADDR	0x4E00000C				NAND flash address
NFDATA	0x4E000010				NAND flash data
NFMECC0	0x4E000014				NAND flash main area ECC0/1
NFMECC1	0x4E000018				NAND flash main area ECC2/3
NFSECC	0x4E00001C				NAND flash spare area ECC
NFSTAT	0x4E000020				NAND flash operation status
NFSTAT0	0x4E000024				NAND flash ECC status for I/O[7:0]
NFSTAT1	0x4E000028				NAND flash ECC status for I/O[15:8]
NFMECC0	0x4E00002C			R	NAND flash main area ECC0 status
NFMECC1	0x4E000030				NAND flash main area ECC1 status
NFSECC	0x4E000034				NAND flash spare area ECC status
NFSBLK	0x4E000038			R/W	NAND flash start block address
NFEBLK	0x4E00003C				NAND flash end block address

表 1-4 。 s3c2440a 特殊寄存器 (表 6/14)

Register Name	Address (B. Endian)	Address (L. Endian)	Acc. Unit	Read/Write	Function
Camera Interface					
CISRCFMT	0x4F000000	←	W	RW	Input source format
CIWDOFST	0x4F000004				Window offset register
CIGCTRL	0x4F000008				Global control register
CICOYSA1	0x4F000018				Y 1 st frame start address for codec DMA
CICOYSA2	0x4F00001C				Y 2 nd frame start address for codec DMA
CICOYSA3	0x4F000020				Y 3 rd frame start address for codec DMA
CICOYSA4	0x4F000024				Y 4 th frame start address for codec DMA
CICOCBSA1	0x4F000028				Cb 1 st frame start address for codec DMA
CICOCBSA2	0x4F00002C				Cb 2 nd frame start address for codec DMA
CICOCBSA3	0x4F000030				Cb 3 rd frame start address for codec DMA
CICOCBSA4	0x4F000034				Cb 4 th frame start address for codec DMA
CICOCRSA1	0x4F000038				Cr 1 st frame start address for codec DMA
CICOCRSA2	0x4F00003C				Cr 2 nd frame start address for codec DMA
CICOCRSA3	0x4F000040				Cr 3 rd frame start address for codec DMA
CICOCRSA4	0x4F000044				Cr 4 th frame start address for codec DMA
CICOTRGFMT	0x4F000048				Target image format of codec DMA
CICOCTRL	0x4F00004C				Codec DMA control related
CICOSCPREPRATIO	0x4F000050				Codec pre-scaler ratio control
CICOSCPREDST	0x4F000054				Codec pre-scaler destination format
CICOSCCTRL	0x4F000058				Codec main-scaler control
CICOTAREA	0x4F00005C				Codec scaler target area
CICOSTATUS	0x4F000064				Codec path status
CIPRCLRSA1	0x4F00006C				RGB 1 st frame start address for preview DMA
CIPRCLRSA2	0x4F000070				RGB 2 nd frame start address for preview DMA
CIPRCLRSA3	0x4F000074				RGB 3 rd frame start address for preview DMA
CIPRCLRSA4	0x4F000078				RGB 4 th frame start address for preview DMA
CIPRTRGFMT	0x4F00007C				Target image format of preview DMA
CIPRCTRL	0x4F000080				Preview DMA control related
CIPRSCPREPRATIO	0x4F000084				Preview pre-scaler ratio control
CIPRSCPREDST	0x4F000088				Preview pre-scaler destination format
CIPRSCCTRL	0x4F00008C				Preview main-scaler control
CIPRTAREA	0x4F000090				Preview scaler target area
CIPRSTATUS	0x4F000098				Preview path status
CIIMGCP	0x4F0000A0				Image capture enable command

表 1-4 。 s3c2440a 特殊寄存器 (表 7/14)

Register Name	Address (B. Endian)	Address (L. Endian)	Acc. Unit	Read/Write	Function
UART					
ULCON0	0x50000000	←	W	R/W	UART 0 line control
UCON0	0x50000004				UART 0 control
UFCON0	0x50000008				UART 0 FIFO control
UMCON0	0x5000000C				UART 0 modem control
UTRSTAT0	0x50000010			R	UART 0 Tx/Rx status
UERSTAT0	0x50000014				UART 0 Rx error status
UFSTAT0	0x50000018				UART 0 FIFO status
UMSTAT0	0x5000001C				UART 0 modem status
UTXH0	0x50000023	0x50000020	B	W	UART 0 transmission hold
URXH0	0x50000027	0x50000024		R	UART 0 receive buffer
UBRDIV0	0x50000028	←	W	R/W	UART 0 baud rate divisor
ULCON1	0x50004000				UART 1 line control
UCON1	0x50004004				UART 1 control
UFCON1	0x50004008				UART 1 FIFO control
UMCON1	0x5000400C				UART 1 modem control
UTRSTAT1	0x50004010			R	UART 1 Tx/Rx status
UERSTAT1	0x50004014				UART 1 Rx error status
UFSTAT1	0x50004018				UART 1 FIFO status
UMSTAT1	0x5000401C				UART 1 modem status
UTXH1	0x50004023	0x50004020	B	W	UART 1 transmission hold
URXH1	0x50004027	0x50004024		R	UART 1 receive buffer
UBRDIV1	0x50004028	←	W	R/W	UART 1 baud rate divisor
ULCON2	0x50008000				UART 2 line control
UCON2	0x50008004				UART 2 control
UFCON2	0x50008008				UART 2 FIFO control
UTRSTAT2	0x50008010			R	UART 2 Tx/Rx status
UERSTAT2	0x50008014				UART 2 Rx error status
UFSTAT2	0x50008018				UART 2 FIFO status
UTXH2	0x50008023	0x50008020	B	W	UART 2 transmission hold
URXH2	0x50008027	0x50008024		R	UART 2 receive buffer
UBRDIV2	0x50008028	←	W	R/W	UART 2 baud rate divisor

表 1-4 。 s3c2440a 特殊寄存器（表 8/14 ）

Register Name	Address (B. Endian)	Address (L. Endian)	Acc. Unit	Read/ Write	Function
PWM Timer					
TCFG0	0x51000000	←	W	R/W	Timer configuration
TCFG1	0x51000004				Timer configuration
TCON	0x51000008				Timer control
TCNTB0	0x5100000C				Timer count buffer 0
TCMPB0	0x51000010				Timer compare buffer 0
TCNTO0	0x51000014			R	Timer count observation 0
TCNTB1	0x51000018			R/W	Timer count buffer 1
TCMPB1	0x5100001C				Timer compare buffer 1
TCNTO1	0x51000020			R	Timer count observation 1
TCNTB2	0x51000024			R/W	Timer count buffer 2
TCMPB2	0x51000028				Timer compare buffer 2
TCNTO2	0x5100002C			R	Timer count observation 2
TCNTB3	0x51000030			R/W	Timer count buffer 3
TCMPB3	0x51000034				Timer compare buffer 3
TCNTO3	0x51000038			R	Timer count observation 3
TCNTB4	0x5100003C			R/W	Timer count buffer 4
TCNTO4	0x51000040			R	Timer count observation 4

表 1-4 。 s3c2440a 特殊寄存器 (表 9/14)

Register Name	Address (B. Endian)	Address (L. Endian)	Acc. Unit	Read/ Write	Function
USB Device					
FUNC_ADDR_REG	0x52000143	0x52000140	B	R/W	Function address
PWR_REG	0x52000147	0x52000144			Power management
EP_INT_REG	0x5200014B	0x52000148			EP interrupt pending and clear
USB_INT_REG	0x5200015B	0x52000158			USB interrupt pending and clear
EP_INT_EN_REG	0x5200015F	0x5200015C			Interrupt enable
USB_INT_EN_REG	0x5200016F	0x5200016C			Interrupt enable
FRAME_NUM1_REG	0x52000173	0x52000170		R	Frame number lower byte
FRAME_NUM2_REG	0x52000177	0x52000174			Frame number higher byte
INDEX_REG	0x5200017B	0x52000178		R/W	Register index
EP0_CSR	0x52000187	0x52000184			Endpoint 0 status
IN_CSR1_REG	0x52000187	0x52000184			In endpoint control status
IN_CSR2_REG	0x5200018B	0x52000188			In endpoint control status
MAXP_REG	0x52000183	0x52000180			Endpoint max packet
OUT_CSR1_REG	0x52000193	0x52000190			Out endpoint control status
OUT_CSR2_REG	0x52000197	0x52000194			Out endpoint control status
OUT_FIFO_CNT1_REG	0x5200019B	0x52000198		R	Endpoint out write count
OUT_FIFO_CNT2_REG	0x5200019F	0x5200019C			Endpoint out write count
EP0_FIFO	0x520001C3	0x520001C0		R/W	Endpoint 0 FIFO
EP1_FIFO	0x520001C7	0x520001C4			Endpoint 1 FIFO
EP2_FIFO	0x520001CB	0x520001C8			Endpoint 2 FIFO
EP3_FIFO	0x520001CF	0x520001CC			Endpoint 3 FIFO
EP4_FIFO	0x520001D3	0x520001D0			Endpoint 4 FIFO
EP1_DMA_CON	0x52000203	0x52000200			EP1 DMA Interface control
EP1_DMA_UNIT	0x52000207	0x52000204			EP1 DMA Tx unit counter
EP1_DMA_FIFO	0x5200020B	0x52000208			EP1 DMA Tx FIFO counter
EP1_DMA_TTC_L	0x5200020F	0x5200020C			EP1 DMA Total Tx counter
EP1_DMA_TTC_M	0x52000213	0x52000210			EP1 DMA Total Tx counter
EP1_DMA_TTC_H	0x52000217	0x52000214			EP1 DMA Total Tx counter
EP2_DMA_CON	0x5200021B	0x52000218	B	R/W	EP2 DMA interface control
EP2_DMA_UNIT	0x5200021F	0x5200021C			EP2 DMA Tx Unit counter
EP2_DMA_FIFO	0x52000223	0x52000220			EP2 DMA Tx FIFO counter
EP2_DMA_TTC_L	0x52000227	0x52000224			EP2 DMA total Tx counter
EP2_DMA_TTC_M	0x5200022B	0x52000228			EP2 DMA total Tx counter

表 1-4 。 s3c2440a 特殊寄存器 (表 10/14)

Register Name	Address (B. Endian)	Address (L. Endian)	Acc. Unit	Read/ Write	Function
USB Device (Continued)					
EP2_DMA_TTC_H	0x5200022F	0x5200022C			EP2 DMA Total Tx counter
EP3_DMA_CON	0x52000243	0x52000240			EP3 DMA Interface control
EP3_DMA_UNIT	0x52000247	0x52000244			EP3 DMA Tx Unit counter
EP3_DMA_FIFO	0x5200024B	0x52000248			EP3 DMA Tx FIFO counter
EP3_DMA_TTC_L	0x5200024F	0x5200024C			EP3 DMA Total Tx counter
EP3_DMA_TTC_M	0x52000253	0x52000250			EP3 DMA Total Tx counter
EP3_DMA_TTC_H	0x52000257	0x52000254			EP3 DMA Total Tx counter
EP4_DMA_CON	0x5200025B	0x52000258			EP4 DMA Interface control
EP4_DMA_UNIT	0x5200025F	0x5200025C			EP4 DMA Tx Unit counter
EP4_DMA_FIFO	0x52000263	0x52000260			EP4 DMA Tx FIFO counter
EP4_DMA_TTC_L	0x52000267	0x52000264			EP4 DMA Total Tx counter
EP4_DMA_TTC_M	0x5200026B	0x52000268			EP4 DMA Total Tx counter
EP4_DMA_TTC_H	0x5200026F	0x5200026C			EP4 DMA Total Tx counter
Watchdog Timer					
WTCN	0x53000000	←	W	R/W	Watchdog timer mode
WTDAT	0x53000004				Watchdog timer data
WTCNT	0x53000008				Watchdog timer count
IIC					
IICCON	0x54000000	←	W	R/W	IIC control
IICSTAT	0x54000004				IIC status
IICADD	0x54000008				IIC address
IICDS	0x5400000C				IIC data shift
IICLC	0x54000010				IIC multi-master line control
IIS					
IISCON	0x55000000,02	0x55000000	HW,W	R/W	IIS control
IISMOD	0x55000004,08	0x55000004			IIS mode
IISPSR	0x55000008,0A	0x55000008			IIS prescaler
IISFCON	0x5500000C,0E	0x5500000C			IIS FIFO control
IISFIFO	0x55000012	0x55000010	HW		IIS FIFO entry

表 1-4 。 s3c2440a 特殊寄存器 (表 11/14)

Register Name	Address (B. Endian)	Address (L. Endian)	Acc. Unit	Read/Write	Function
I/O port					
GPACON	0x56000000	←	W	R/W	Port A control
GPADAT	0x56000004				Port A data
GPBCON	0x56000010				Port B control
GPBDAT	0x56000014				Port B data
GPBUP	0x56000018				Pull-up control B
GPCCON	0x56000020				Port C control
GPCDAT	0x56000024				Port C data
GPCUP	0x56000028				Pull-up control C
GPDCON	0x56000030				Port D control
GPDDA1T	0x56000034				Port D data
GPDUP	0x56000038				Pull-up control D
GPECON	0x56000040				Port E control
GPEDAT	0x56000044				Port E data
GPEUP	0x56000048				Pull-up control E
GPFCON	0x56000050				Port F control
GPFDAT	0x56000054				Port F data
GPFUP	0x56000058				Pull-up control F
GPGCON	0x56000060				Port G control
GPGDAT	0x56000064				Port G data
GPGUP	0x56000068				Pull-up control G
GPHCON	0x56000070				Port H control
GPHDAT	0x56000074				Port H data
GPHUP	0x56000078				Pull-up control H
GPJCON	0x560000D0				Port J control
GPJDAT	0x560000D4				Port J data
GPJUP	0x560000D8				Pull-up control J
MISCCR	0x56000080				Miscellaneous control
DCLKCON	0x56000084				DCLK0/1 control
EXTINT0	0x56000088				External interrupt control register 0
EXTINT1	0x5600008C				External interrupt control register 1
EXTINT2	0x56000090				External interrupt control register 2

表 1-4 。 s3c2440a 特殊寄存器 (表 12/14)

Register Name	Address (B. Endian)	Address (L. Endian)	Acc. Unit	Read/Write	Function
i/o port (continued)					
EINTFLT0	0x56000094	←	W	R/W	Reserved
EINTFLT1	0x56000098				Reserved
EINTFLT2	0x5600009C				External interrupt filter control register 2
EINTFLT3	0x560000A0				External interrupt filter control register 3
EINTMASK	0x560000A4				External interrupt mask
EINTPEND	0x560000A8				External interrupt pending
GSTATUS0	0x560000AC			R	External pin status
GSTATUS1	0x560000B0			R/W	Chip ID
GSTATUS2	0x560000B4				Reset status
GSTATUS3	0x560000B8				Inform register
GSTATUS4	0x560000BC				Inform register
MSLCON	0x560000CC				Memory sleep control register
RTC					
RTCCON	0x57000043	0x57000040	B	R/W	RTC control
TICNT	0x57000047	0x57000044			Tick time count
RTCALM	0x57000053	0x57000050			RTC alarm control
ALMSEC	0x57000057	0x57000054			Alarm second
ALMMIN	0x5700005B	0x57000058			Alarm minute
ALMHOUR	0x5700005F	0x5700005C			Alarm hour
ALMDATE	0x57000063	0x57000060			alarm day
ALMMON	0x57000067	0x57000064			Alarm month
ALMYEAR	0x5700006B	0x57000068			Alarm year
BCDSEC	0x57000073	0x57000070			BCD second
BCDMIN	0x57000077	0x57000074			BCD minute
BCDHOUR	0x5700007B	0x57000078			BCD hour
BCDDATE	0x5700007F	0x5700007C			BCD day
BCDDAY	0x57000083	0x57000080			BCD date
BCDMON	0x57000087	0x57000084			BCD month
BCDYEAR	0x5700008B	0x57000088			BCD year

表 1-4 。 s3c2440a 特殊寄存器 (表 13/14)

Register Name	Address (B. Endian)	Address (L. Endian)	Acc. Unit	Read/Write	Function
A/D Converter					
ADCCON	0x58000000	←	W	R/W	ADC control
ADCTSC	0x58000004				ADC touch screen control
ADCDLY	0x58000008				ADC start or interval delay
ADCDAT0	0x5800000C			R	ADC conversion data
ADCDAT1	0x58000010				ADC conversion data
ADCUPDN	0x58000014			R/W	Stylus up or down interrupt status
SPI					
SPCON0,1	0x59000000,20	←	W	R/W	SPI control
SPSTA0,1	0x59000004,24			R	SPI status
SPPIN0,1	0x59000008,28			R/W	SPI pin control
SPPRE0,1	0x5900000C,2C				SPI baud rate prescaler
SPTDAT0,1	0x59000010,30				SPI Tx data
SPRDAT0,1	0x59000014,34			R	SPI Rx data
SD Interface					
SDICON	0x5A000000	←	W	R/W	SDI control
SDIPRE	0x5A000004				SDI baud rate prescaler
SDICARG	0x5A000008				SDI command argument
SDICCON	0x5A00000C				SDI command control
SDICSTA	0x5A000010			R/(C)	SDI command status
SDIRSP0	0x5A000014			R	SDI response
SDIRSP1	0x5A000018				SDI response
SDIRSP2	0x5A00001C				SDI response
SDIRSP3	0x5A000020				SDI response
SDITIMER	0x5A000024			R/W	SDI data / busy timer
SDIBSIZE	0x5A000028				SDI block size
SDIDCON	0x5A00002C				SDI data control
SDIDCNT	0x5A000030			R	SDI data remain counter
SDIDSTA	0x5A000034			R/(C)	SDI data status
SDIFSTA	0x5A000038			R	SDI FIFO status
SDIIMSK	0x5A00003C	←	W		SDI interrupt mask
SDIDAT	0x5A000043	0x5A000040	B	R/W	SDI data

表 1-4 。 s3c2440a 特殊寄存器（表 14/14 ）

Register Name	Address (B. Endian)	Address (L. Endian)	Acc. Unit	Read/Write	Function
AC97 Audio-CODEC Interface					
AC_GLBCTRL	0x5B000000	←	W	R/W	AC97 global control register
AC_GLBSTAT	0x5B000004			R	AC97 global status register
AC_CODEC_CMD	0x5B000008			R/W	AC97 codec command register
AC_CODEC_STAT	0x5B00000C			R	AC97 codec status register
AC_PCMADDR	0x5B000010				AC97 PCM out/in channel FIFO address register
AC_MICADDR	0x5B000014				AC97 mic in channel FIFO address register
AC_PCMDATA	0x5B000018			R/W	AC97 PCM out/in channel FIFO data register
AC_MICDATA	0x5B00001C				AC97 MIC in channel FIFO data register

S3C2440A 专用寄存器注释：

1. 在小端模式下，必须使用小端地址；大端模式下，必须使用大端地址；
2. 每个特殊寄存器必须按照推荐的方式进行操作；
3. 除了ADC寄存器，RTC寄存器和UART寄存器外，其他寄存器都必须以字为单元（32位）进行读写。
4. 对ADC,RTC,UART寄存器进行读/写时，必须仔细考虑使用的大/小端模式。
5. W：32位寄存器，必须用LDR/STR指令或整型数型指针(int *)进行访问；
HW：16位寄存器，必须用LDRH/STRH或短整型数指针(short int *)访问；
B：8位寄存器，必须用LDRB/STRB或字符型指针(char int *)访问。

第二章

处理器工作模式

2.1 概述

S3C2440 采用了非常先进的 ARM920T 内核，它是由 ARM(Advanced RISC Machines) 公司研制的。

2.2 处理工作状态

从程序员的角度上看，ARM920T 可以工作在下面两种工作状态下的一种：

- ARM 状态：执行 32 位字对齐的 ARM 指令
- THUMB 状态：执行 16 位半字对齐的 THUMB 指令。在这种状态下，PC 寄存器的第一位来选择一个字中的哪个半字

注意：这两种状态的转换不影响处理模式和寄存器的内容。

2.3 切换状态

进入 THUMB 状态

进入 THUMB 状态，可以通过执行 BX 指令，同时将操作数寄存器的状态位（0 位）置 1 来实现。

当从异常（IRQ, FIQ, UNDEF, ABORT, SWI 等）返回时，只要进入异常处理前处理器处于 THUMB 状态，也会自动进入 THUMB 状态。

进入 ARM 状态

进入 ARM 状态，可以通过执行 BX 指令，并且操作数寄存器的状态位（0 位）清零来实现。

当处理进入异常（IRQ, FIQ, RESET, UNDEF, ABORT, SWI 等）。这时，PC 值保持在异常模式下的 link 寄存器中，并从异常向量地址处开始执行处理程序。

存储空间的格式

ARM920T 将存储器空间视为从 0 开始由字节组成的线性集合，字节 0 到 3 中保存了第一个字节，字节 4 到 7 中保存第二个字，以此类推，ARM920T 对存储的字，可以按照小端或大端的方式对待。

大端格式:

在这种格式中, 字数据的高字节存储在低地址中, 而字数据的低字节则存放在高地址中, 因此字节 0 存储在数据的 24 到 31 行里

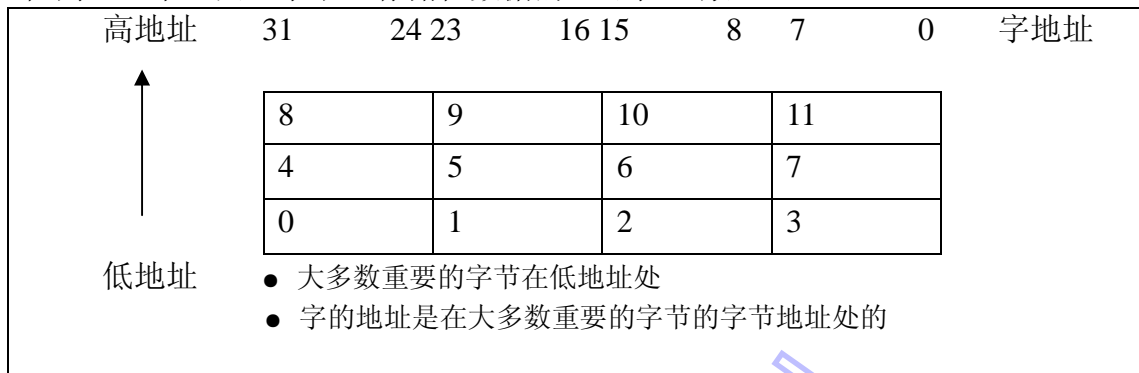


图 2-1 以大端格式存储字数据

小端格式:

与大端格式相反, 在小端存储格式中, 低地址中存放的是字数据的低字节, 高地址存放的是字数据的高字节。

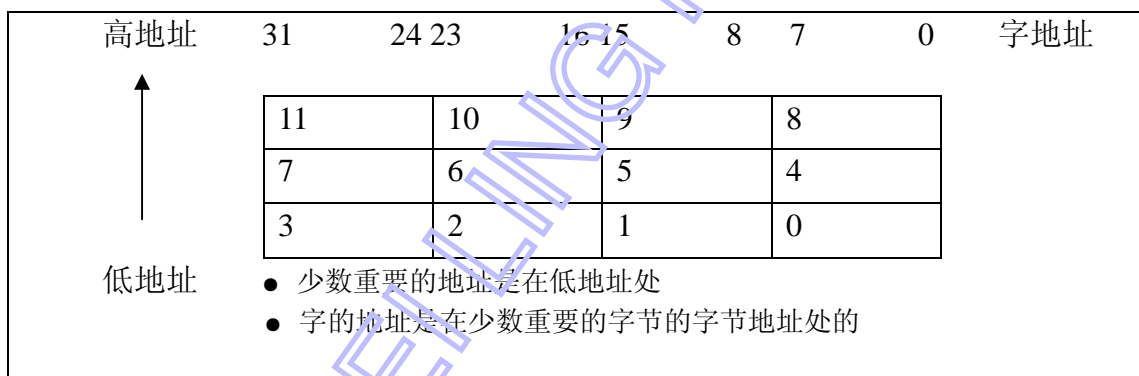


图 2-2 以小端格式存储字数据

2.4 指令长度

指令可以是 32 位长度(在 ARM 状态下) 或 16 位长度 (在 THUMB 状态) 。

数据类型

ARM920T 支持字节(8 位), 半字(16 位) 和字(32 位) 数据类型。字必须按照 4 字节对齐, 半字必须是 2 字节对齐。

2.5 操作模式

ARM920T 支持 7 种操作模式:

- 用户模式(user 模式), 运行应用的普通模式

- 快速中断模式(fiq 模式), 用于支持数据传输或通道处理
- 中断模式(irq 模式), 用于普通中断处理
- 超级用户模式(svc 模式), 操作系统的保护模式
- 异常中断模式(abt 模式), 输入数据后登入或预取异常中断指令
- 系统模式(sys 模式), 使操作系统使用的一个有特权的用户模式
- 未定义模式(und 模式), 执行了未定义指令时进入该模式]

外部中断, 异常操作或软件控制都可以改变中断模式。大多数应用程序都是在用户模式下进行, 进入特权模式是为了处理中断或异常请求或操作保护资源服务的。

2.6 寄存器

ARM 共有 37 个 32 位的寄存器, 其中 31 个是通用寄存器, 6 个是状态寄存器。但在同一时间, 对程序员来说并不是多有的寄存器都可见。在某一时刻存储器是否可见, 是由处理器当前的工作状态和工作模式决定的。

ARM 状态寄存器

在ARM状态下, 任何时刻都可以看到16个通用寄存器, 1个或2个状态寄存器。在特权模式(非用户模式)下会切换到具体模式下的寄存器组, 其中包括模式专用的私有(banked)寄存器。图2-3显示了在每个模式下哪种寄存器是可见的; 私有寄存器上都有一个黑三角标记。

ARM状态寄存器系列中含有16个直接操作寄存器: R0到R15。除了R15外其他的都是通用寄存器, 可用来存放地址或数据值。除此之外, 实际上有17个寄存器用来存放状态信息。

寄存器14	专职持有返回点的地址, 在系统执行一条“跳转并链接”(BL)指令的时候, R14将收到一个R15的拷贝。其他时候, 它可以用作一个通用寄存器。相应的私有寄存器R14_svc, R14_irq, R14_fiq, R14_abt和R14_und都同样用来保存在中断或异常发生时, 或在中断和异常时执行了BL指令时, R15的返回值。
寄存器15	这个寄存器是程序计数器(PC)。在ARM状态下, R15的bits[1:0]为0, bits[31:2]保存了PC的值。在Thumb 状态下, bits[0]为0同时bits[31:1]保存了PC值。
寄存器16	这个寄存器是CPSR(当前程序状态寄存器), 用来保存当前代码标志和当前模式位。

FIQ模式拥有7个私有寄存器R8-14(R8_fiq-R14_fiq)。在ARM状态下, 多数FIQ处理都不需要保存任何寄存器。用户、中断、异常中止, 超级用户和未定义模式都拥有2个私有寄存器, R13和R14。允许这些模式都可拥有1个私有堆栈指针和

链接寄存器。

ARM 状态下寄存器集

系统和用户	FIQ	超级用户	异常中止	IRQ	未定义
R0	R0	R0	R0	R0	R0
R1	R1	R1	R1	R1	R1
R2	R2	R2	R2	R2	R2
R3	R3	R3	R3	R3	R3
R4	R4	R4	R4	R4	R4
R5	R5	R5	R5	R5	R5
R6	R6	R6	R6	R6	R6
R7	R7	R7	R7	R7	R7
R8	R8_fiq	R8	R8	R8	R8
R9	R9_fiq	R9	R9	R9	R9
R10	R10_fiq	R10	R10	R10	R10
R11	R11_fiq	R11	R11	R11	R11
R12	R12_fiq	R12	R12	R12	R12
R13	R13_fiq	R13_svc	R13_abt	R13_irq	R13_und
R14	R14_fiq	R14_svc	R14_abt	R14_irq	R14_und
R15 (PC)	R15 (PC)	R15 (PC)	R15 (PC)	R15 (PC)	R15 (PC)

ARM 状态下程序状态寄存器

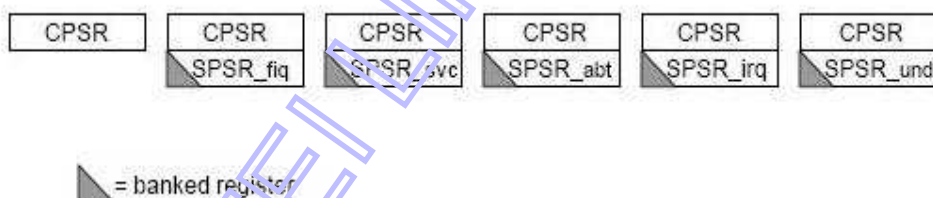


图 2-3 ARM 状态下的寄存器结构

THUMB 状态寄存器

THUMB 状态寄存器是 ARM 状态寄存器的一个子集。程序员可以直接操作 8 个通用寄存器 R0-R7，同样可以这样操作程序计数器(PC)，堆栈指针寄存器(SP)，链接寄存器(LR)，和 CPSR。它们都是各个特权模式下的私有寄存器，链接寄存器和程序状态寄存器(SPSRs)。如图 2-4

THUMB状态下的寄存器集

系统和用户	FIQ	超级	中止	IRQ	未定义
R0	R0	R0	R0	R0	R0
R1	R1	R1	R1	R1	R1
R2	R2	R2	R2	R2	R2
R3	R3	R3	R3	R3	R3
R4	R4	R4	R4	R4	R4
R5	R5	R5	R5	R5	R5
R6	R6	R6	R6	R6	R6
R7	R7	R7	R7	R7	R7
SP	SP_fiq	SP_svc	SP_abt	SP_irq	SP_und
LR	LR_fiq	LR_svc	LR_abt	LR_irq	LR_und
PC	PC	PC	PC	PC	PC

THUMB状态下的程序寄存器

CPSR	CPSR	CPSR	CPSR	CPSR	CPSR
	SPSR_fiq	SPSR_svc	SPSR_abt	SPSR_irq	SPSR_und

▲ = 影子寄存器

图 2-4Thumb 状态下寄存器结构

ARM 和 THUMB 状态寄存器间的关系:

- THUMB 状态下 R0-R7 和 ARM 状态下 R0-R7 是等同的
- THUMB 状态下 CPSRs 和 SPSRs 跟 ARM 状态的 CPSR 和 SPSRs 是等同的
- THUMB 状态下的 SP 映射在 ARM 状态下得 R13 上
- THUMB 状态下的 LR 映射在 ARM 状态下得 R14 上
- THUMB 状态下程序计数器映射在 ARM 状态下的程序计数器上(R15)

图 2-5 显示了它们的关系:

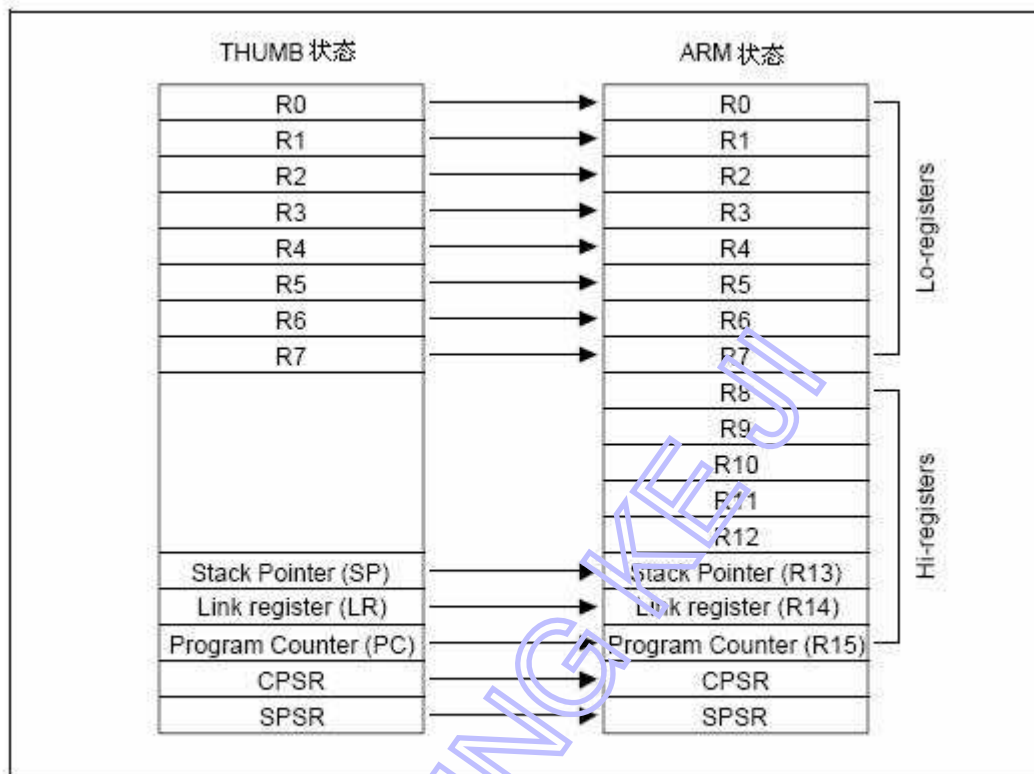


图2-5 THUMB状态下和ARM状态下寄存器之间的映射关系

在 THUMB 状态下访问高地址寄存器

在 THUMB 状态下寄存器 R8-15(高地址寄存器)不是标准寄存器集。但是，汇编语言的程序员可以访问它们并用它们作快速暂存。

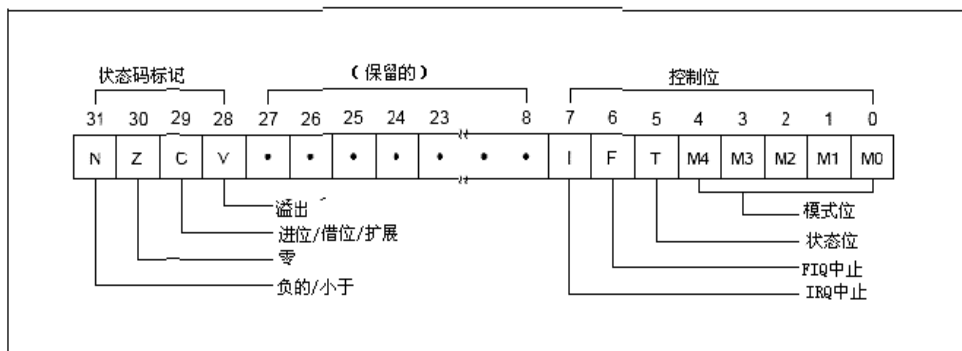
采用 MOV 指令的某个变型，从 R0-R7（低地址寄存器）的某个寄存器传送数据到达高地址寄存器，或者从高地址寄存器传送到低地址寄存器。还可以采用 CMP 和 ADD 指令，将高地址寄存器的值与低地址寄存器的值进行比较或相加。想获得这方面更多的信息，请参考图 3-34。

2.7 程序寄存器状态

ARM920T 具有一个当前程序状态寄存器（CPSR），另外还有 5 个保存程序状态寄存器（SPSRs）用于异常中断处理。这些寄存器的功能有：

- 保留最近完成的 ALU 操作的信息
- 控制中断的使能和禁止
- 设置处理器的操作模式

下图 2-6 显示了程序状态寄存器的位定义：



2.7.1 条件码标志

N、Z、C、V 均为条件码标志位。它们的内容根据算术或逻辑运算的结果所改变，并且来作为一些指令是否运行的检测条件。

在 ARM 状态下，绝大多数指令都是有条件执行的，详情请看表 3-2；

在 Thumb 状态下，仅有分支指令是有条件执行的，详情请看图 3-46。

2.7.2 控制位

PSR 的低 8 位(包括 I、F、T、M[4:0])称为控制位，当发生异常时这些位将被改变。如果处理器工作在特权模式下，这些位也可以由软件操纵。

T 标记位	该位反映处理器的运行状态。该位被设置为 1 时，处理器执行在 THUMB 状态，否则执行在 ARM 状态，这些由外部信号 TBIT 反映出来。注意：软件决不能改变 CPSR 的 TBIT 状态。如果这样做，处理器将会进入一种不可预知的状态。
中断禁止位	I、F 位为中断禁止位，当它们被设置成 1 时可以相应的禁止 IRQ 和 FIQ 中断。
模式位	M4, M3, M2, M1 和 M0 位(M[4:0])是模式位，它们决定了处理器的操作模式，如下表 2-7 所示。并不是所有的模式组合位都决定一个有效的处理器模式，只有那些明确地描述值才能被使用，用户必须意识到任何一种非法的值写入模式位，处理器都会进入到一种不可重获的状态，如果这种情况发生，就要进行复位。
保留位	PSR 中的其余位为保留位，当改变 PSR 中的位或者控制位时，必须确保保留位不被改变，在程序中也不要使用保留位来存储数据值。因此在将来的处理器中它们也许作为 1 或 0 来读。

图2-7 PSR模式位的值

M[4:0]	模式	可视的THUMB状态寄存器	可视的ARM状态寄存器
10000	用户模式	R7..R0, LR, SP, PC, CPSR	R14..R0, PC, CPSR
10001	FIQ 模式	R7..R0, LR_fiq, SP_fiq, PC, CPSR, SPSR_fiq	R7..R0, R14_fiq..R8_fiq, PC, CPSR, SPSR_fiq
10010	IRQ 模式	R7..R0, LR_irq, SP_irq, PC, CPSR, SPSR_irq	R12..R0, R14_irq, R13_irq, PC, CPSR, SPSR_irq
10011	超级用户模式	R7..R0, LR_svc, SP_svc, PC, CPSR, SPSR_svc	R12..R0, R14_svc, R13_svc, PC, CPSR, SPSR_svc
10111	中止	R7..R0, LR_abt, SP_abt, PC, CPSR, SPSR_abt	R12..R0, R14_abt, R13_abt, PC, CPSR, SPSR_abt
11011	未定义模式	R7..R0, LR_und, SP_und, PC, CPSR, SPSR_und	R12..R0, R14_und, R13_und, PC, CPSR
11111	系统模式	R7..R0, LR, SP, PC, CPSR	R14..R0, PC, CPSR

保留位 PSR 中的其余位为保留位，当改变 PSR 中的位或者控制位时，必须确保保留位不被改变，在程序中也不要使用保留位来存储数据值。因此在将来的处理器中它们也许作为 1 或 0 来读。

2.8 异常

当正常的程序执行流程被临时中断时，称为产生了异常。例如程序执行转向一个外设的中断请求。在异常能被处理前，当前处理器的状态必须被保留，这样按处理程序完成时就能恢复原始的程序。

有可能同时产生好几个异常，如果出现这种情况，就应该按固定的顺序处理。详情请看后面对异常优先级的说明。

2.8.1 进入异常时的行为

当一个异常发生时，ARM920T 将进行以下步骤：

1. 将下一条指令的地址保存到相应的 Link 寄存器中。如果异常是从 ARM

状态进入的，下一条指令的地址拷贝到 **Link** 寄存器（根据异常的类型，数值为当前 **PC+4** 或 **PC+8**，具体请看表 2-8）。如果异常是从 **THUMB** 态进入，那么写入到 **link** 寄存器的值是当前的 **PC** 偏移一个值。这表示异常处理程序不需要关心是从哪种状态进入异常的。例如，在 **SWI** 情况下，无论是来自 **ARM** 或 **THUMB** 状态，处理程序只要采用 **MOVSPC, R14-svc** 语句，总可以返回到原始程序的下一条语句。

2. 拷贝 **CPSR** 到相应的 **SPSR**；
3. 根据异常类型强制改变 **CPRS** 模式位的值；
4. 令 **PC** 的值指向异常处理向量所指的下一条指令。

这时也可能设置中断禁止标志，以防止不可估计的异常嵌套发生。

当处理器处于 **Thumb** 状态时发生了异常，当 **PC** 载入异常矢量所在地址时，它将自动的切换到 **ARM** 状态。

2.82 离开异常处理时的行为

当完成异常处理时：

1. 将 **Link** 寄存器，减去相应的偏移量，赋给 **PC**（偏移量的值由异常的类型决定）；
2. 拷贝回 **SPSR** 到 **CPSR**；
3. 如果在进入中断时设置了中断禁止标志，清除它。

注意：你不需要特别指明切换回 **THUMB** 状态。因为原来的 **CPSR** 被自动的保存到了 **SPSR**。

2.83 异常进入/退出的总结

表 2-8 总结了在进入异常时，保留到相应的 **R14** 中的 **PC** 的值，和推荐使用的退出异常处理时采用的语句。

	返回指令	进入异常时R14的值		Notes
		ARM R14_x	THUMB R14_x	
BL	MOV PC, R14	PC + 4	PC + 2	1
SWI	MOVS PC, R14_svc	PC + 4	PC + 2	1
UDEF	MOVS PC, R14_und	PC + 4	PC + 2	1
FIQ	SUBS PC, R14_fiq, #4	PC + 4	PC + 4	2
IRQ	SUBS PC, R14_irq, #4	PC + 4	PC + 4	2
PABT	SUBS PC, R14_abt, #4	PC + 4	PC + 4	1
DABT	SUBS PC, R14_abt, #8	PC + 8	PC + 8	3
RESET	NA	—	—	4

注意:

- 1.这里 PC 所赋的是 BL/SWI/未定义模式等指令所取的地址,它们在预取的阶段就被中断了。
- 2.这里 PC 所赋的是由于 FIQ 或 IRQ 取得了优先权,而没有来得及得到执行的指令地址。
- 3.这里 PC 所赋的地址是 Load 或 Store 指令的地址,它们在执行时产生了数据的异常中断。
- 4.在 R14_svc 复位之前保存的数值是不可预知的。

2.8.4 FIQ 中断

FIQ (快速中断请求) 异常通常是用来支持数据传输和通道操作的,在 ARM 状态下,它具有充分的私有寄存器,用来减少寄存器存取的需要(从而减少进入中断前的“上下文切换”的工作)。

FIQ 中断是由外部设备通过拉低 nFIQ 引脚触发的。通过对 ISYNC 输入引脚的控制 nFIQ 可以区别同步或异步的传输情况,当 ISYNC 为低电平 nFIQ 和 nIRQ 将被认为是异步的,中断之前产生同步周期延长的话会影响处理器的流程。

不管是 ARM 还是 Thumb 状态下的异常,FIQ 处理程序都可以通过执行以下的语句来退出中断处理:

```
SUBS    PC, R14-fiq, #4
```

通过设置 CPSR 的 F 标记位可以禁止 FIQ 中断(但是要注意到在用户模式下是不可行的)。如果 F 标记位已经清除,ARM920T 在每个指令的最后检测来自 FIQ 中断同步器的低电输出。

2.8.5 IRQ 中断

IRQ (中断请求) 异常是由 nIRQ 输入低电平引发的普通中断。IRQ 中断相对 FIQ 中断来说是优先级低,当一个 FIQ 中断序列进入时它将被屏蔽。IRQ 也可以通过设置 CPRS 中的“I”标志来禁止,只能在特权(非用户)模式下这样做。

无论 IRQ 发生在 ARM 或者 Thumb 状态下,都可以采用以下语句来退出中断处理:

```
SUBS    PC, R14-fiq, #4
```

2.8.6 Abort 异常中止

异常中止表示当前存储访问不能完成。通过外部的 ABORT 输入信号来告知内核。ARM920T 在每次的存储操作中检测该异常是否发生。

有两种类型的异常中止:

- 预取指异常中断: 指令预取时产生

- 数据异常中断：数据访问时产生

如果产生预取指中止，所取得的指令将会被标志为无效的，但是异常不会立即发生，要直到取指到达了管道的头部才会发生。/如果这些指令不执行-例如在管道内发生了分支跳转，那么异常就不会发生了。

如果产生数据异常中止，根据指令类型进行操作：

- 简单数据传输指令 (LDM, STR) 写回改变的基址寄存器：异常中断处理器必须清楚这些。
- 取消交换指令尽管它还没执行
- 数据块传输指令 (LDM, STM) 完成。如果设置为写回，基址已经矫正。如果指令超出了数据的写基址（传输目录中有它的基址），就应该防止写超出。在中止异常将会发生时，所有寄存器的覆盖写入都是禁止的。这意味着特别是 R15（经常是最后一个改变的寄存器）的值将在中止的 LDM 指令中保留下来。

Abort 机制使得页面虚拟存储器机制得以实现。在采用虚拟存储器的系统中，处理器可以产生任意的地址。当某个地址的数据无效，MMU（存储器管理单元）将产生一个 abort 中止。这样 abort 的处理程序也就不需要了解实际可用存储空间的大小，也不需要了解异常中断对他的影响。

在完成 了异常中断的处理后，通过以下语句推出中断处理（与 ARM 状态还是 Thumb 状态无关）：

```
SUBS    PC, R14-abt,#4; 预取指 abort
```

```
SUBS    PC, R14-abt,#8; 数据 abort
```

通过执行该语句，就恢复了 PC 和 CPSR，并重试被中断的指令。

2.8.7 软件中断

SWI（软件中断指令）用来进入超级用户模式，通常用于请求特殊的超级用户功能。SWI 的处理程序通过执行以下状态（ARM 或 Thumb）的语句，退出异常处理；

```
MOV     PC, R14-svc
```

通过执行该语句，就恢复了 PC 和 CPSR，并返回到 SWI 后面的指令上。

注意：

前面提到的 nFIQ, nIRQ, ISYNC, LOCK, BIGEND,和 ABORT 引脚只存在于 ARM920TCPU 的内核里。

2.8.8 未定义指令

当 ARM920T 遇到一个它不能执行的指令，它将产生一个未定义指令陷阱。这个机制是软件仿真器用来扩展 Thumb 和 ARM 指令集用的。

在完成对未知指令的处理后，陷阱处理程序应该执行以下的语句退出异常处

理（无论是 ARM 或 Thumb 状态）：

MOVS PC, R14-und

通过执行该语句，恢复了 CPSR，并返回执行未定义指令的下一条指令。

2.8.9 异常中断向量

异常中断向量的地址如下图所示：

表格 2-3 异常中断向量

地址	异常中断类型	进入时处理器模式
0x00000000	Reset	Supervisor
0x00000004	Undefined instruction	Undefined
0x00000008	Software interrupt	Supervisor
0x0000000C	Abort (prefetch)	Abort
0x00000010	Abort (data)	Abort
0x00000014	Reserved	Reserved
0x00000018	IRQ	IRQ
0x0000001C	FIQ	FIQ

2.8.10 异常中断优先级

当多个异常中断同时发生时，处理器根据一个固定的优先级系统来决定处理他们的顺序。

最高优先级：

1. 复位
2. 数据 abort
3. FIQ
4. IRQ
5. 预取指 abort

最低优先级：

6. 未定义指令，软件中断。

并非所有的异常中断都可能同时发生：

未定义指令和软件中断是相互排斥的，因为他们都对应于当前指令的唯一的（非重叠的）解码结果。

如果一个数据 abort 和 FIQ 中断同时发生了，并且此时的 FIQ 中断是使能的，ARM920T 先进入到数据 abort 处理程序，然后立即进入 FIQ 向量。从 FIQ 正常的返回后，数据 abort 的处理程序才恢复执行。将数据 abort 设计为比 FIQ 拥有更高的优先级，可以确保传输错误不能逃避检测。这种情况下进入 FIQ 异常处理时间延长了，这一时间必须考虑到 FIQ 中断最长反映时间的计算中去。

2.8.11 中断反应时间

最坏情况下的 FIQ 中断的反映时间，假设它是使能的，包括通过同步器最长请求时间（如果是异步则是 $T_{syncmax}$ ），加上最长指令的完成时间(T_{ldm} ，最长指令是 LDM，它装载了所有的寄存器包括 PC)，加上数据 abort 进入时间 (T_{exc})，加上进入 FIQ 所需要的时间 (T_{fiq})。在这些时间的最后，ARM920T 会执行位于 0x1C 的指令。

$T_{syncmax}$ 是 3 个处理器周期， T_{ldm} 是 20 个， T_{exc} 是 3 个， T_{fiq} 是 2 个周期，因此总共是 28 个处理器周期。在一个连续的 20MHz 的处理器时钟系统里，它的使用时间超过了 1.4 微妙。最长 IRQ 的反应时间计算是类似的，但是必须考虑到更高优先级的 FIQ 中断可以推迟任意长时间进入 IRQ 中断处理。最小的 FIQ 或 IRQ 的反应时间包括通过同步器($T_{syncmin}$)的最短时间加上 T_{fiq} ，它是 4 个处理器周期。

2.8.12 复位

当 nRESET 信号为低，ARM920T 放弃任何指令的执行，并从增加的地址处取指令。

当 nRESET 信号为高时 ARM920T 进行如下操作：

- 1 将当前的 PC 值和 CPSR 值写入 R14_svc 和 SPSR_svc，已保存的 PC 和 CPSR 的值是未知的。
- 2 强制 M[4:0]为 10011(超级用户模式)，将 CPSR 中的“I”和“F”位设为 1，并将 T 位清零。
- 3 强制 PC 从 0x00 地址处取得下一条指令。
- 4 恢复为 ARM 状态并开始执行。

第五章 存储器控制器

5.1 概述

s3c2440A 的存储器控制器提供访问外部存储器所需的存储器控制信号。

s3c2440A 的存储器控制器有以下特性：

- 大小端（通过软件选择）
- 地址空间：每个 bank 有 128M 的字节（总共 1G 字节 /8 个 banks）
- 可编程的访问位宽，bank0(16/32 位)，其他 bank（8/16/32 位）
- 共 8 个存储器 banks
 - 6 个是 ROM，SRAM 等类型存储器 bank
 - 2 个是可以作为 ROM、SRAM、SDRAM 等存储器 bank
- 7 个固定的存储器 bank 起始地址
- 最后一个 bank 的起始地址可调整
- 最后两个 bank 大小可编程
- 所有存储器 bank 的访问周期可编程
- 总线访问周期可通过插入外部 wait 来延长
- 支持 SDRAM 的自刷新和掉电模式

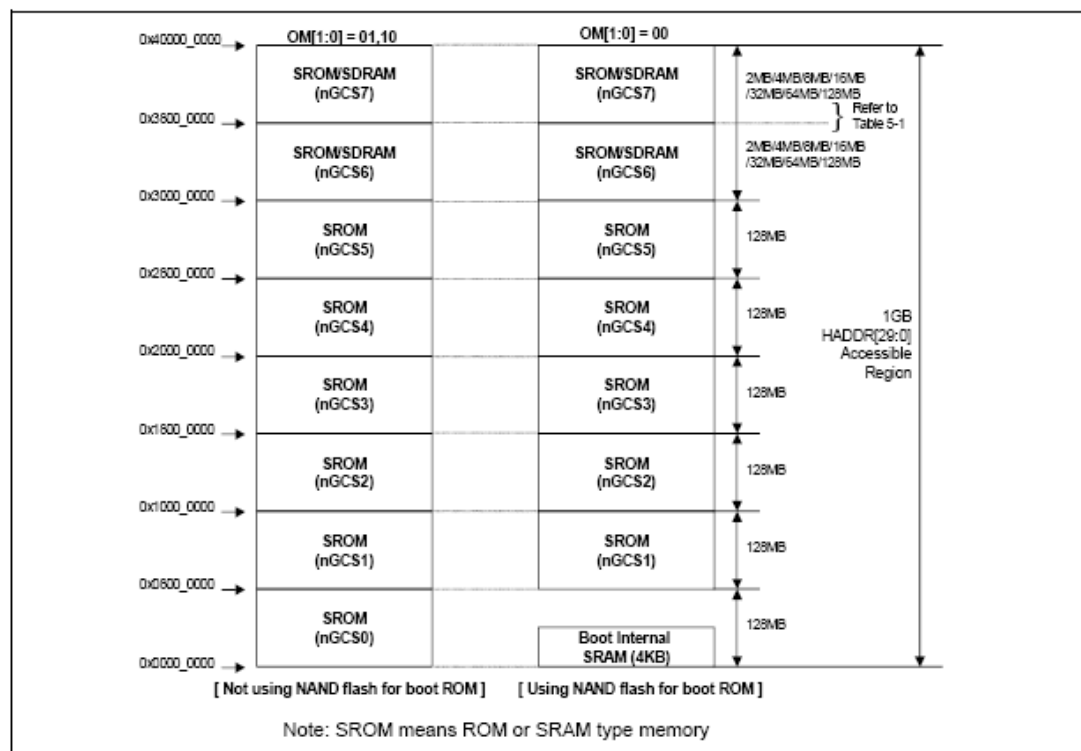


Figure 5-1. S3C2440A Memory Map after Reset

地址	2MB	4MB	8MB	16MB	32MB	64MB	128MB
Bank6							
起始地址	0x3000_0000	0x3000_0000	0x3000_0000	0x3000_0000	0x3000_0000	0x3000_0000	0x3000_0000
结束地址	0x301f_ffff	0x303f_ffff	0x307f_ffff	0x30ff_ffff	0x31ff_ffff	0x33ff_ffff	0x37ff_ffff
Bank7							
起始地址	0x3020_0000	0x3040_0000	0x3080_0000	0x3100_0000	0x3200_0000	0x3400_0000	0x3800_0000
结束地址	0x303f_ffff	0x307f_ffff	0x30ff_ffff	0x31ff_ffff	0x33ff_ffff	0x37ff_ffff	0x3fff_ffff

5.2 功能描述

5.2.1 BANK0 总线宽度

BANK0 的数据总线 (nGCS0) 应该被配置为 16 位和 32 位中的一个。因为 BANK0 作为启动 ROM bank 工作时, BANK 的总线宽度应该在第一次 ROM 访问之前被决定, 其依赖于 OM[0:1]在重启时的逻辑电平。

OM1(操作模式 1)	OM2 (操作模式 2)	启动 ROM 数据宽度
0	0	Nand Flash 模式
0	1	16 位
1	0	32 位
1	1	测试模式

存储器 (SRAM/SDRAM) 地址引脚连接

存储器地址引脚	S3c2440A 地址 @8 位数据总线	S3c2440A 地址 @16 位数据总线	S3c2440A 地址 @32 位数据总线
A0	A0	A1	A1
A1	A1	A2	A2
...

存储器 (SDRAM) 地址引脚连接

Bank 大小	总线宽度	基本单元	存储器配置	Bank 地址
2MByte	x8	16Mbit	(1M x 8 x 2Bank) x 1	A20
	x16		(512K x 16 x 2B) x 1	
4MB	x16		(1M x 8 x 2B) x 2	A21
	x16		(1Mx 8 x 2B) x 2	
8MB	x16	16Mb	(2M x 4 x 2B) x 4	A22
	x32		(1Mx 8x 2B) x 4	
	x8	64Mb	(4Mx 8 x 2B) x 1	A[22:21]
	x8		(2M x 8 x 4B) x 1	
	x16		(2M x 16 x 2B) x 1	A22
	x16		(1M x 16 x 4B) x 1	A[22:21]
	x32		(512K x 32 x 4B) x 1	
16MB	x32	16Mb	(2Mx 4 x 2B) x 8	A23
	x8	64Mb	(8Mx 4 x 2B) x 2	
	x8		(4Mx 4 x 4B) x 2	A[23:22]
	x16		(4Mx 8 x 2B) x 2	A23
	x16		(2Mx 8 x 4B) x 2	A[23:22]
	x32		(2Mx 16 x 2B) x 2	A23
	x32		(1Mx 16x 4B) x 2	A[23:22]
	x8	128Mb	(4Mx 8 x 4B) x 1	
	x16		(2Mx 16 x 4B) x 1	
32MB	x16	64Mb	(8Mx 4 x 2B) x 4	A24
	x16		(4Mx 4 x 4B) x 4	A[24:23]
	x32		(4Mx 8 x 2B) x 4	A24
	x32		(2Mx 8 x 4B) x 4	A[24:23]
	x16	128Mb	(4Mx 8 x 4B) x 2	
	x32		(2Mx 16 x 4B) x 2	
	X8	256Mb	(8Mx 8x 4B) x 1	
	x16		(4Mx 16x 4B) x 1	
64MB	x32	128Mb	(4M x 8 x 4B) x 4	A[25:24]
	x16	256Mb	(8M x 8 x 4B) x 2	
	x32		(4M x 16 x 4B) x 2	
	X8	512Mb	(16M x 8 x 4B) x 1	
128MB	x32	256Mb	(8M x 8 x 4Bank) x 4	A[26:25]
	X8	512Mb	(32M x 4 x 4B) x 2	
	X16		(16Mx 8 x 4B) x 2	
	x32		(8M x 16 x 4B) x 2	

5.2.2 nWAIT 引脚操作

如果相对应每个存储器 bank 的 WAIT 位 (BWSCON 寄存器中的 WS_n 位) 使能, 当存储器 bank 被激活, nOE 信号低电平有效持续时间可以被外部引脚 nWAIT 所延长。从 tacc-1 时刻开始检测 nWAIT 的状态。在采用 nWAIT 是高电平后的下一个时钟周期, nOE 也会被置高电平。nWE 与 nOE 也有同样的关系。

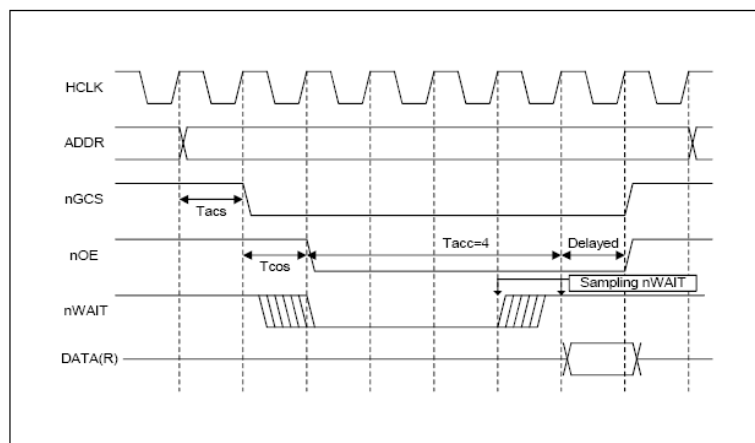


Figure 5-2. S3C2440A External nWAIT Timing Diagram (Tacc=4)

5.2.3 nXBREQ/nXBACK 引脚操作

如果 nXBREQ 引脚被拉低电平, s3c2440A 将通过置 nXBACK 引脚低电平对此响应。如果 nXBACK 为低电平, 则地址/数据总线和存储器控制信号将处于如表 1-1 所示的高阻态。当 nXBREQ 引脚被拉高以后, nXBACK 引脚也会恢复高电平。

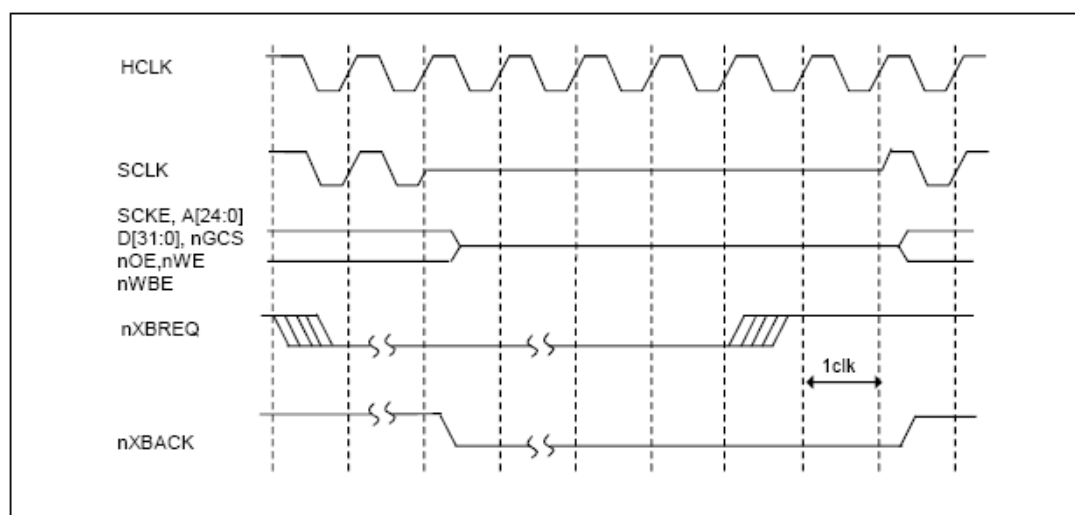


Figure 5-3. S3C2440A nXBREQ/nXBACK Timing Diagram

5.3 存储器接口示例

5.3.1 ROM 存储器接口示例

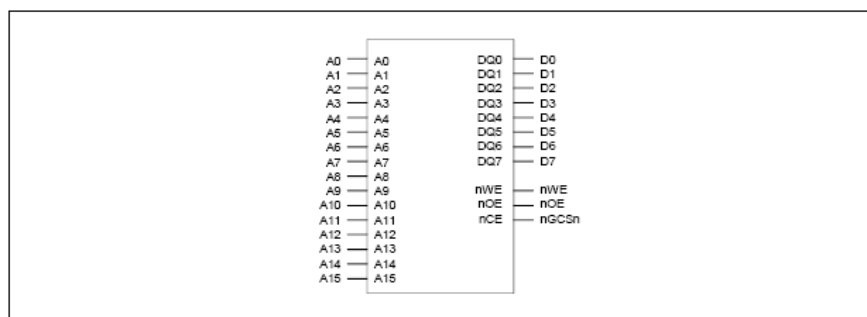


Figure 5-4. Memory Interface with 8-bit ROM

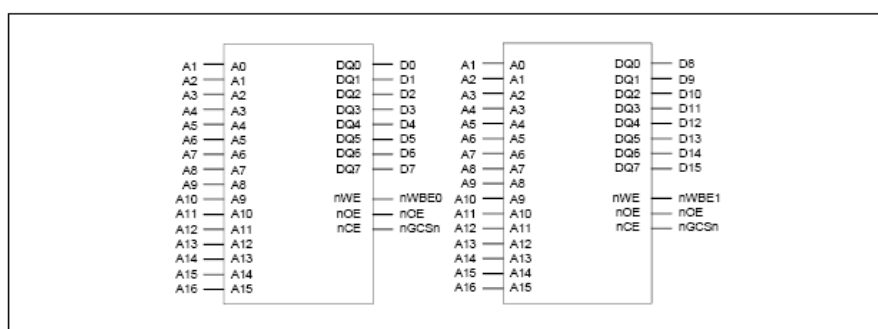


Figure 5-5. Memory Interface with 8-bit ROM x 2

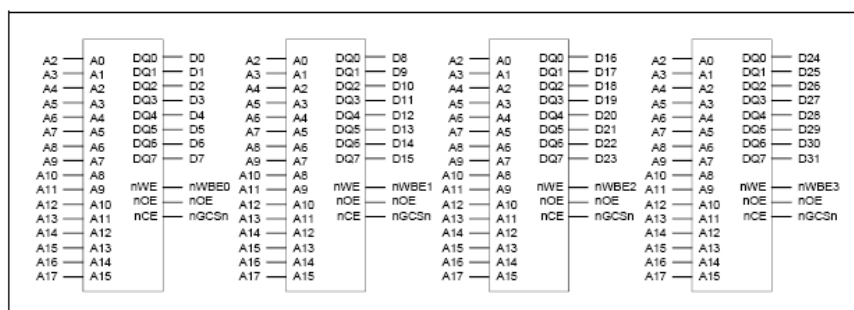


Figure 5-6. Memory Interface with 8-bit ROM x 4

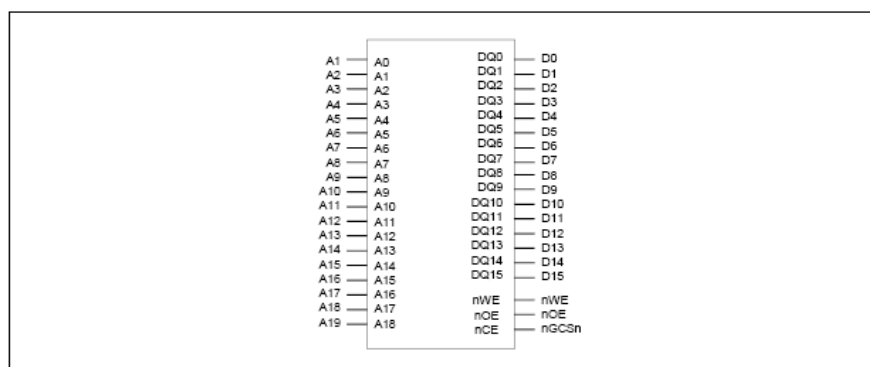


Figure 5-7. Memory Interface with 16-bit ROM

5.3.2 SRAM 存储器接口示例

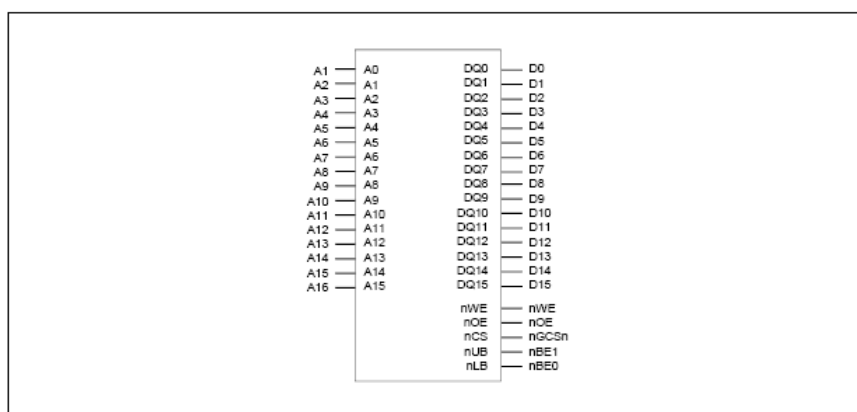


Figure 5-8. Memory Interface with 16-bit SRAM

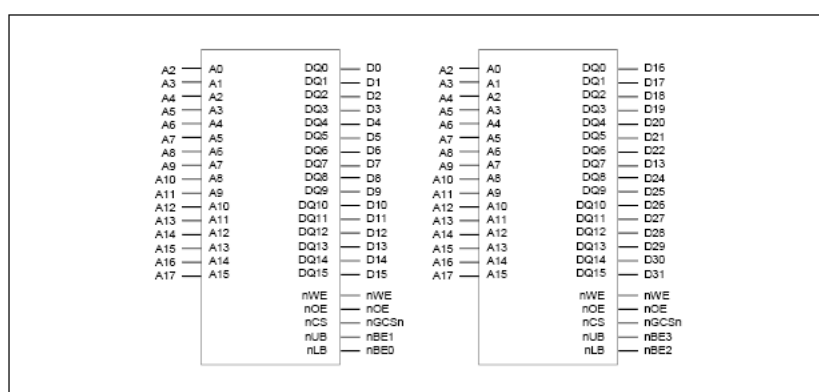


Figure 5-9. Memory Interface with 16-bit SRAM x 2

5.3.3 SDRAM 存储器接口示例

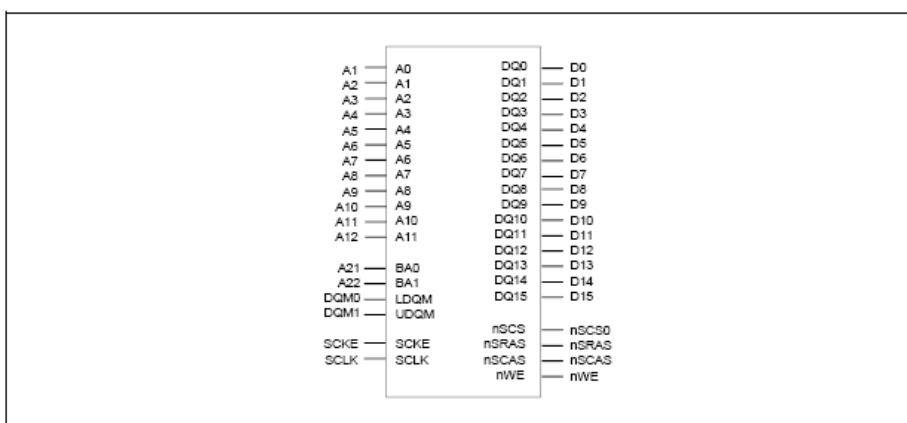


Figure 5-10. Memory Interface with 16-bit SDRAM (4Mx16, 4banks)

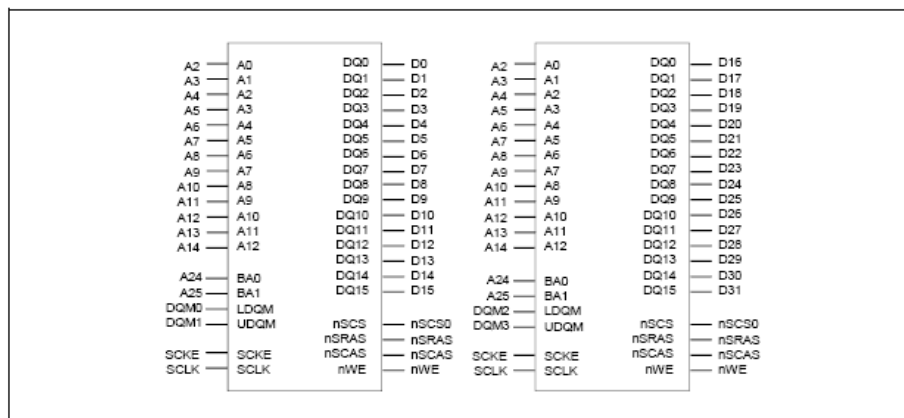


Figure 5-11. Memory Interface with 16-bit SDRAM (4Mx16x4Bank * 2ea)

5.3.4 可编程存储器访问周期

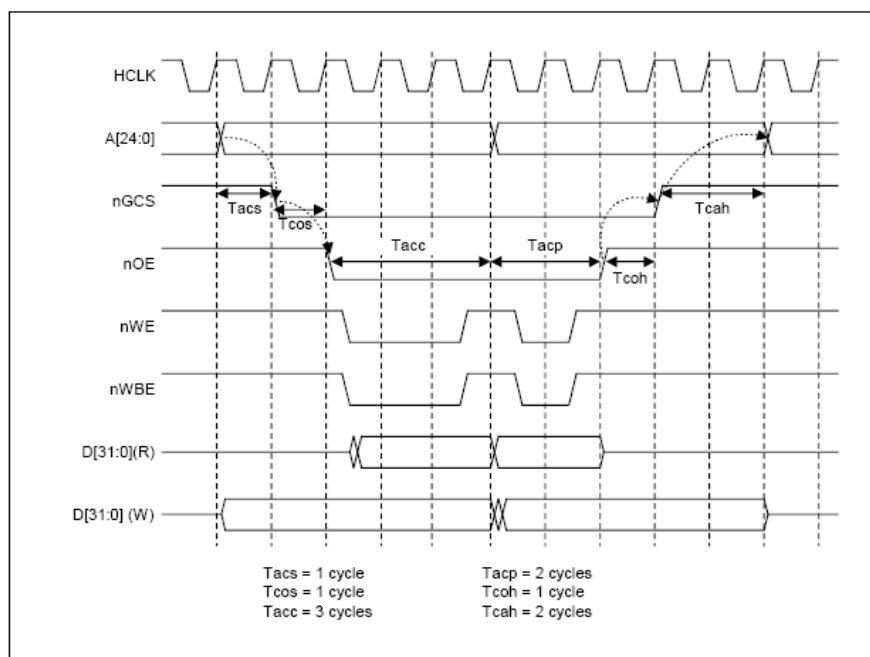


Figure 5-12. S3C2440A nGCS Timing Diagram

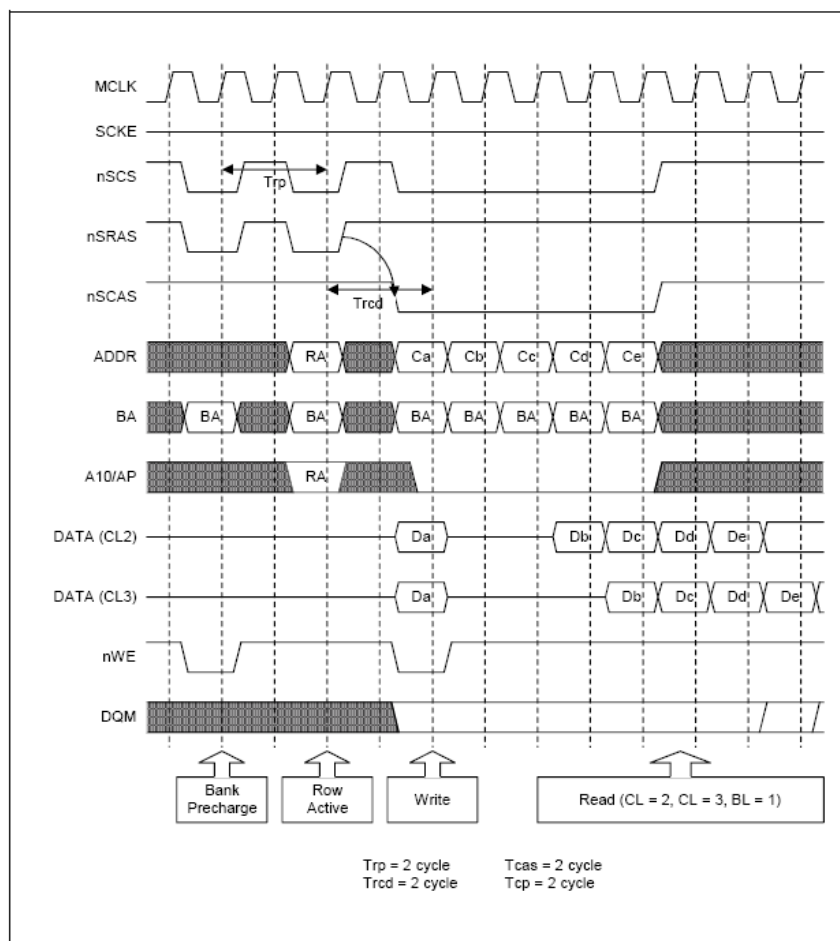


Figure 5-13. S3C2440A SDRAM Timing Diagram

5.4 存储器控制器寄存器

5.4.1 总线宽度&等待控制寄存器

BUS WIDTH & WAIT CONTROL REGISTER (BWSCON)

寄存器	地址	读写	描述	复位值
BWSCON	0x48000000	R/W	总线宽度&等待状态控制寄存器	0x000000

BWSCON	位	描述	初始值
ST7	[31]	决定SRAM对bank7 是否使用UB/LB, 0 = 不使用 UB/LB (引脚对应 nWBE[3:0]) 1 = 使用 UB/LB (引脚对应 nBE[3:0])	0
WS7	[30]	决定对于bank7 的等待状态 0 = WAIT无效, 1 = WAIT使能	0
DW7	[29:28]	决定对于bank7 的数据总线宽度 00 = 8-bit 01 = 16-bit, 10 = 32-bit 11 = reserved	0
ST6	[27]	决定SRAM对bank6 是否使用UB/LB, 0 = 不使用 UB/LB (引脚对应 nWBE[3:0]) 1 = 使用 UB/LB (引脚对应 nBE[3:0])	0
WS6	[26]	决定对于bank6 的等待状态 0 = WAIT无效, 1 = WAIT使能	0
DW6	[25:24]	决定对于bank6 的数据总线宽度 00 = 8-bit 01 = 16-bit, 10 = 32-bit 11 = reserved	0
ST5	[23]	决定SRAM对bank5 是否使用UB/LB, 0 = 不使用 UB/LB (引脚对应 nWBE[3:0]) 1 = 使用 UB/LB (引脚对应 nBE[3:0])	0
WS5	[22]	决定对于bank5 的等待状态 0 = WAIT无效, 1 = WAIT使能	0
DW5	[21:20]	决定对于bank5 的数据总线宽度 00 = 8-bit 01 = 16-bit, 10 = 32-bit 11 = reserved	0
ST4	[19]	决定SRAM对bank4 是否使用UB/LB, 0 = 不使用 UB/LB (引脚对应 nWBE[3:0]) 1 = 使用 UB/LB (引脚对应 nBE[3:0])	0
WS4	[18]	决定对于bank4 的等待状态 0 = WAIT无效, 1 = WAIT使能	0
DW4	[17:16]	决定对于bank4 的数据总线宽度 00 = 8-bit 01 = 16-bit, 10 = 32-bit 11 = reserved	0
ST3	[15]	决定SRAM对bank3 是否使用UB/LB, 0 = 不使用 UB/LB (引脚对应 nWBE[3:0]) 1 = 使用 UB/LB (引脚对应 nBE[3:0])	0
WS3	[14]	决定对于bank3 的等待状态 0 = WAIT无效, 1 = WAIT使能	0
DW3	[13:12]	决定对于bank3 的数据总线宽度 00 = 8-bit 01 = 16-bit, 10 = 32-bit 11 = reserved	0

ST2	[11]	决定SRAM对bank2 是否使用UB/LB, 0 = 不使用 UB/LB (引脚对应 nWBE[3:0]) 1 = 使用 UB/LB (引脚对应 nBE[3:0])	0
WS2	[10]	决定对于bank2 的等待状态 0 = WAIT无效, 1 = WAIT使能	0
DW2	[9: 8]	决定对于bank2 的数据总线宽度 00 = 8-bit 01 = 16-bit, 10 = 32-bit 11 = reserved	0
ST1	[7]	决定SRAM对bank1 是否使用UB/LB, 0 = 不使用 UB/LB (引脚对应 nWBE[3:0]) 1 = 使用 UB/LB (引脚对应 nBE[3:0])	0
WS1	[6]	决定对于bank1 的等待状态 0 = WAIT无效, 1 = WAIT使能	0
DW1	[5: 4]	决定对于bank1 的数据总线宽度 00 = 8-bit 01 = 16-bit, 10 = 32-bit 11 = reserved	0
DW0	[2:1]	决定对于bank0 的数据总线宽度 00 = 8-bit 01 = 16-bit, 10 = 32-bit 11 = reserved	-
保留	[0]	保留为 0	0

注:

1.在存储器控制器里的主时钟都对应总线时钟。例如, SRAM 中的 HCLK 与总线时钟一致, SDRAM 中的 SCLK 与总线时钟一致。在本章(存储器控制器)中, 一个时钟就意味一个总线时钟。

2.nBE[3:0]是 nWBE[3:0]和 nOE 的与信号。

5.4.2 Bank 控制寄存器

BANK CONTROL REGISTER (BANKCONn: nGCS0-nGCS5)

寄存器	地址	读写	描述	复位值
BANKCON0	0x48000004	R/W	Bank0 控制寄存器	0x0700
BANKCON1	0x48000008	R/W	Bank1 控制寄存器	0x0700
BANKCON2	0x4800000C	R/W	Bank2 控制寄存器	0x0700
BANKCON3	0x48000010	R/W	Bank3 控制寄存器	0x0700
BANKCON4	0x48000014	R/W	Bank4 控制寄存器	0x0700
BANKCON5	0x48000018	R/W	Bank5 控制寄存器	0x0700

BANKCONn	位	描述	初始值
Tacs	[14:13]	nGCSn前的地址建立时间 00 = 0 clock 01 = 1 clock 10 = 2 clocks 11 = 4 clocks	00
Tcos	[12:11]	nOE前的片选建立时间 00 = 0 clock 01 = 1 clock 10 = 2 clocks 11 = 4 clocks	00
Tacc	[10:8]	访问周期 000 = 1 clock 001 = 2 clocks 010 = 3 clocks 011 = 4 clocks 100 = 6 clocks 101 = 8 clocks 110 = 10 clocks 111 = 14 clocks 注: nWAIT 信号被使用, Tacc 大于等于 4 clocks.	111
Tcoh	[7:6]	nOE后的片选保持时间 00 = 0 clock 01 = 1 clock 10 = 2 clocks 11 = 4 clocks	00
Tcah	[5:4]	nGCSn后的地址保持时间 00 = 0 clock 01 = 1 clock 10 = 2 clocks 11 = 4 clocks	00
Tacp	[3:2]	Page模式下的访问周期 00 = 2 clocks 01 = 3 clocks 10 = 4 clocks 11 = 6 clocks	00
PMC	[1:0]	Page模式配置 00 = normal (1 data) 01 = 4 data 10 = 8 data 11 = 16 data	00

5.4.3 Bank 控制寄存器

BANK CONTROL REGISTER (BANKCONn: nGCS6-nGCS7)

寄存器	地址	读写	描述	复位值
BANKCON6	0x4800001C	R/W	Bank6 控制寄存器	0x18008
BANKCON7	0x48000020	R/W	Bank7 控制寄存器	0x18008

BANKCONn	位	描述	初始值
MT	[16:15]	Determine the memory type for bank6 and bank7. 00 = ROM or SRAM 01 = Reserved (Do not use) 10 = Reserved (Do not use) 11 = Sync. DRAM	11
Memory Type = ROM or SRAM [MT=00] (15-bit)			
Tacs	[14:13]	nGCSn前的地址建立时间 00 = 0 clock 01 = 1 clock 10 = 2 clocks 11 = 4 clocks	00
Tcos	[12:11]	nOE前的片选建立时间 00 = 0 clock 01 = 1 clock 10 = 2 clocks 11 = 4 clocks	00
Tacc	[10:8]	访问周期 000 = 1 clock 001 = 2 clocks 010 = 3 clocks 011 = 4 clocks 100 = 6 clocks 101 = 8 clocks 110 = 10 clocks 111 = 14 clocks	111
Tcoh	[7:6]	nOE后的片选保持时间 00 = 0 clock 01 = 1 clock 10 = 2 clocks 11 = 4 clocks	00
Tcah	[5:4]	nGCSn后的地址保持时间 00 = 0 clock 01 = 1 clock 10 = 2 clocks 11 = 4 clocks	00
Tacp	[3:2]	Page模式下的访问周期 00 = 2 clocks 01 = 3 clocks 10 = 4 clocks 11 = 6 clocks	00
PMC	[1:0]	Page模式配置 00 = normal (1 data) 01 = 4 data 10 = 8 data 11 = 16 data	00
Memory Type = SDRAM [MT=11] (4-bit)			
Trcd	[3:2]	RAS to CAS 延时 00 = 2 clocks 01 = 3 clocks 10 = 4 clocks	10
SCAN	[1:0]	列地址位数 00 = 8-bit 01 = 9-bit 10 = 10-bit	00

5.4.4 刷新控制寄存器

REFRESH CONTROL REGISTER (REFRESH)

寄存器	地址	读写	描述	复位值
REFRESH	0x48000024	R/W	SDRAM 刷新控制寄存器	0xac0000

REFRESH	位	描述	初始值
REFEN	[23]	SDRAM 刷新使能 0 = 无效 1 = 有效 (自, CBR/自动刷新) 注:CBR (CAS Before RAS, 列提前于行定位) 式刷新	1
TREFMD	[22]	SDRAM刷新模式 0 = CBR/Auto Refresh 1 = Self Refresh 在自刷新模式下, SDRAM 控制信号被置于适当的电平	0
Trp	[21:20]	SDRAM RAS预充电时间 00 = 2 clocks 01 = 3 clocks 10 = 4 clocks 11 = 不支持	10
Tsrc	[19:18]	SDRAM 半行周期时间Tsrc 00 = 4 clocks 01 = 5 clocks 10 = 6 clocks 11 = 7 clocks SDRAM 行周期时间: Trc=Tsrc+Trp 如果 Trp=3clocks & Tsrc=7clocks, Trc=3+7=10clocks.	11
Reserved	[17:16]	Not used	00
Reserved	[15:11]	Not used	0000
Refresh Counter	[10:0]	SDRAM 刷新计数值. 参考第六章SDRAM刷新控制寄存器总线优先级所在节 刷新时间 = $(2^{11}-refresh_count+1)/HCLK$ Ex) 如果刷新时间是 7.8 us并且HCLK是 100 MHz, 刷新计数值如下: 刷新计数值 = $2^{11} + 1 - 100 \times 7.8 = 1269$	0

5.4.5 BANKSIZE 寄存器

BANKSIZE REGISTER (BANKSIZE)

寄存器	地址	读写	描述	复位值
BANKSIZE	0x48000028	R/W	可调的 bank 大小寄存器	0x0

BANKSIZE	位	描述	初始值
BURST_EN	[7]	ARM 内核突发操作使能 0 = 无效突发操作 1 = 使能突发操作	0
Reserved	[6]	没用	0
SCKE_EN	[5]	SCKE 使能SDRAM power down 模式 0 = SDRAM power down 模式无效 1 = SDRAM power down 模式有效	0
SCLK_EN	[4]	SCLK仅在SDRAM访问周期内为减少电源消耗被使能。当 SDRAM没有被访问时, SCLK变成低电平。 0 = SCLK 总是激活 1 = SCLK 仅当访问周期才被激活 (推荐).	0
Reserved	[3]	Not used	0
BK76MAP	[2:0]	BANK6/7 存储分布 010 = 128MB/128MB 001 = 64MB/64MB 000 = 32M/32M 111 = 16M/16M 110 = 8M/8M 101 = 4M/4M 100 = 2M/2M	

5.4.6 SDRAM 模式寄存器集寄存器

SDRAM MODE REGISTER SET REGISTER (MRSR)

寄存器	地址	读写	描述	复位值
MRSRB6	0x4800002C	R/W	模式寄存器集寄存器 bank6	xxx
MRSRB7	0x48000030	R/W	模式寄存器集寄存器 bank7	xxx

BANKSIZE	位	描述	初始值
Reserved	[11:10]	没用	-
WBL	[9]	写突发长度 Write burst length 0: 突发(固定的) 1: Reserved	x
TM	[8:7]	测试模式 00: 模式寄存器集 (固定) 01, 10 and 11: 保留	xx
CL	[6:4]	CAS 反应时间 000 = 1 clock, 010 = 2 clocks, 011=3 clocks Others: 保留	xxx
BT	[3]	突发类型 0: 连续的(固定) 1: 保留	x
BL	[2:0]	Burst length 000: 1 (固定) Others: 保留	

第六章 Nand Flash 控制器

6.1 概述

目前,Nor Flash 价格较高, 而 SDRAM 和 Nand Flash 存储器相对经济, 这样促使一些用户在 NAND Flash 上执行启动代码, 在 SDRAM 上执行主程序。

s3c2440A 的驱动代码可以在外部的 NAND Flash 存储器上被执行。为了支持 NAND Flash 的 boot loader, s3c2440A 配备了一个内部的 SRAM 缓冲器名为“Steppingstone”。启动时, NAND Flash 上的前 4KByte 字节将被装载到 Steppingstone 上并且装载到 Steppingstone 上的启动代码会被执行。

一般情况下, 启动代码会拷贝 NAND Flash 上的内容到 SDRAM。使用硬件的 ECC, NAND Flash 的数据被检查。在完成拷贝的基础上, 主程序将在 SDRAM 上被执行。

6.2 特性

- (1) **自动启动**: 启动代码在重启时被传输到 4kbytes 的 Steppingstone 上。传输后代码会在 Steppingstone 上被执行
- (2) **NAND Flash 存储器接口**: 支持 256 字、512 字节、1000 字和 2000Byte 页
- (3) **软件模式**: 用户可以直接访问 NAND Flash, 例如这个特性可以被用于对 NADN Flash 存储器的读/擦除/编程。
- (4) **接口**: 8/16 微的 NADN Flash 存储器接口总线
- (5) **硬件 ECC 生成**, 检测和指示 (软件纠错)
- (6) **SFR 接口**:支持小端模式, 对于数据和 ECC 数据寄存器的字节/半字/字访问, 对于其他寄存器的字访问。
- (7) **Steppingstone 接口**: 支持大小端, 字节/半字/字访问
- (8) **Steppingstone 4kB 内部 SRAM 缓冲器**可以在 NAND Flash 启动后被用于其他目的。

6.2.1 模块图

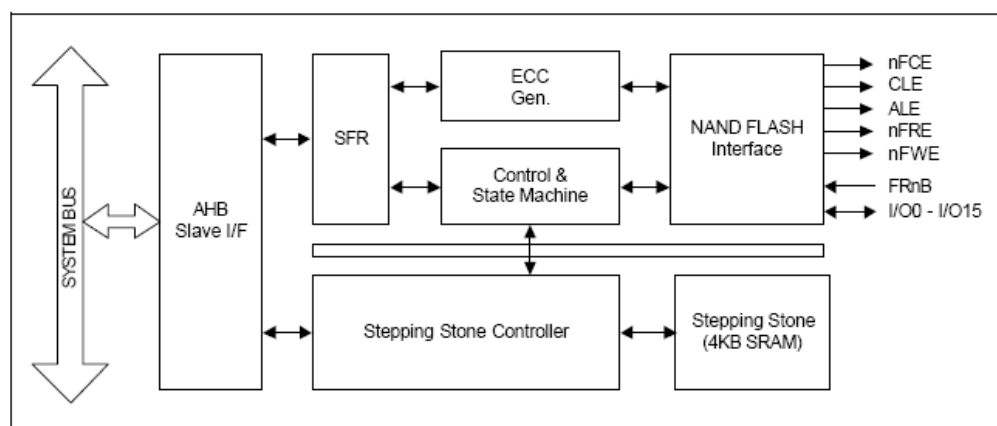


Figure 6-1. NAND Flash Controller Block Diagram

6.2.2 Boot loader 功能

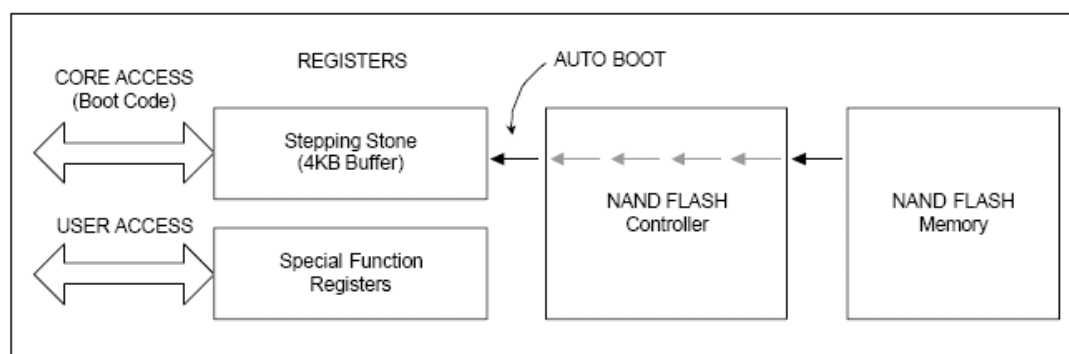


Figure 6-2. NAND Flash Controller Boot Loader Block Diagram

在重启期间，NAND Flash控制器通过引脚状态得到连接NAND Flash的信息 (NCON(Adv flash)，GPG13(页大小)，GPG14(地址周期)，GPG15(总线宽度) - 参考引脚配置)。在上电或重启以后，NAND Flash控制器自动的装载 4KB的boot loader代码。在装载boot loader代码后，其在steppingstone中被执行。

注：在自动重启期间，ECC没有检查，因为NAND Flash的前 4KB数据一般认为没有位错误。

6.2.3 引脚配置

OM[1:0]=00: 使能NAND Flash存储器启动

NCON: NAND Flash存储器选择

0: 普通NAND Flash (256 字/512 字节页大小, 3/4 地址周期)

1: 先进NAND Flash (1K字/2K字节页大小, 4/5 地址周期)

GPG13: NAND Flash存储器页容量选择

0: 页=256 字 (NCON=0) 或页=1K字 (NCON=1)

1: 页=512 字节 (NCON=0) 或页=2K字节(NCON=1)

GPG14: NAND Flash存储器地址周期选择

0: 3 个地址周期 (NCON=0) 或 4 个地址周期 (NCON=1)

1: 4 个地址周期 (NCON=0) 或 5 个地址周期 (NCON=1)

GPG15 NAND Flash存储器总线宽度选择

0: 8 位总线宽度

1: 16 位总线宽度

注: NCON,GPG[15:13]引脚配置在重启期间被?

在通常状态下, 这些引脚必须被设置为输入以至于当通过软件方式进入睡眠模式或异常

状态时, 引脚状态不会被改变。

NAND Flash 存储器配置表

NCON0	GPG13	GPG14	GPG15
0: Normal NAND	0: 256Words	0: 3-Addr	0:8-bit bus width
	1: 512Bytes	1: 4-Addr	
1: Advance NAND	0: 1Kwords	0: 4-Addr	1: 16-bit bus width
	1: 2Kbytes	1: 5-Addr	

注:

例: NAND Flash配置设置

部件	页大小/总大小	NCON0	GPG13	GPG14	GPG15
K9S1208V0M-xxxx	512Byte/512Mbit	0	1	1	0
K9K2G16U0M-xxxx	1KW / 2Gbit	1	0	1	1

6.2.4 NAND Flash 存储器时序

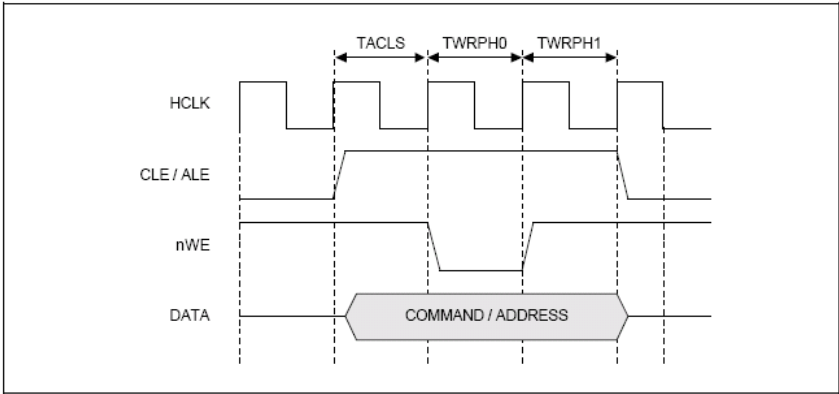


Figure 6-3. CLE & ALE Timing (TACLS=1, TWRPH0=0, TWRPH1=0)

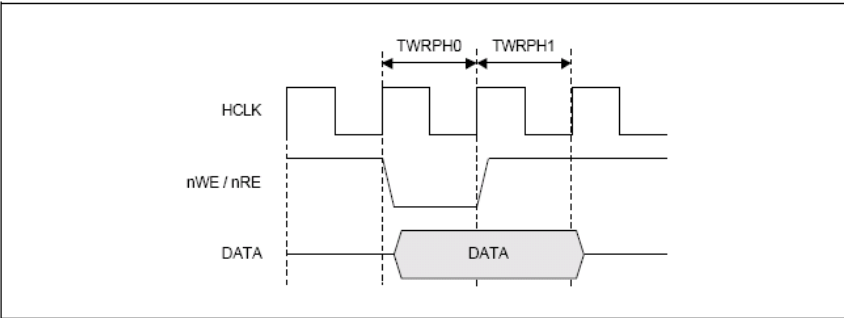


Figure 6-4. nWE & nRE Timing (TWRPH0=0, TWRPH1=0)

6.2.5 软件模式

s3c2440A仅支持软件模式访问。使用此模式，你可以完整的访问NAND Flash。NAND Flash控制器支持NAND Flash存储器的直接访问接口。

- (1) 写命令寄存器=NAND Flash存储器命令周期
- (2) 写地址寄存器=NAND Flash存储器地址周期
- (3) 写数据寄存器=写数据到NAND Flash（写周期）
- (4) 读数据寄存器=从NAND Flash读数据（读周期）
- (5) 读主ECC寄存器和空闲ECC寄存器=从NAND Flash存储器读数据

注：在软件模式中，你必须用查询或中断来检测RnB状态输入引脚。

6.2.6 数据寄存器配置

- (1) 16 位NAND Flash存储器接口

A.字访问

寄存器	大小端	位[31:24]	位[23:16]	位[15:8]	位[7:0]
NFDATA	小端	2 nd I/O[15:8]	2 nd I/O[7:0]	1 st I/O[15:8]	1 st I/O[7:0]
NFDATA	大端	1 st I/O[15:8]	1 st I/O[7:0]	2 nd I/O[15:8]	2 nd I/O[7:0]

B.半字访问

寄存器	大小端	位[31:24]	位[23:16]	位[15:8]	位[7:0]
NFDATA	大/小端	无效值	无效值	1 st I/O[15:8]	1 st I/O[7:0]

- (2) 8 位 NAND Flash 存储器接口

A.字访问

寄存器	大小端	位[31:24]	位[23:16]	位[15:8]	位[7:0]
NFDATA	小端	4 th I/O[7:0]	3 rd I/O[7:0]	2 nd I/O[7:0]	1 st I/O[7:0]
NFDATA	大端	1 st I/O[7:0]	2 nd I/O[7:0]	3 rd I/O[7:0]	4 th I/O[7:0]

B.半字访问

寄存器	大小端	位[31:24]	位[23:16]	位[15:8]	位[7:0]
NFDATA	小端	无效值	无效值	2 nd I/O[7:0]	1 st I/O[7:0]
NFDATA	大端	无效值	无效值	1 st I/O[7:0]	2 nd I/O[7:0]

C.字节访问

寄存器	大小端	位[31:24]	位[23:16]	位[15:8]	位[7:0]
NFDATA	大/小端	无效值	无效值	无效值	1 st I/O[7:0]

6.2.7 Steppingstone (4k 字节 SRAM)

NAND Flash 控制器使用 NAND Flash 作为启动缓冲器且你可以将这个区域用作其他用途。

6.2.8 ECC（错误纠错码）

NAND Flash 控制器包括 4 个 ECC(错误纠错码)模块。两个模块（一个用于 data[7:0],一个用于 data[15:8]）可以被用于（上限）2048bytes 的 ECC 奇偶码的生成，另外两个模块（一个用于 data[7:0],一个用于 data[15:8]）可以被用于（上限）16bytes 的 ECC 奇偶校验码的生

成。

-28 位 ECC 奇偶码=22 位行奇偶+6 位列奇偶

-14 位 ECC 奇偶码=8 位行奇偶+6 位列奇偶

2048 字节 ECC 奇偶码分配表

	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
MECCn_0	P64	P64'	P32	P32'	P16	P16'	P8	P8'
MECCn_1	P1024	P1024'	P512	P512'	P256	P256'	P128	P128'
MECCn_2	P4	P4'	P2	P2'	P1	P1'	P2048	P2048'
MECCn_3	P8192	P8192'	P4096	P4096'	—	—	—	—

16 字节ECC奇偶码分配表

	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
SECCn_0	P16	P16'	P8	P8'	P4	P4'	P2	P2'
SECCn_1	P1	P1'	P64	P64'	P32	P32'	-	-

6.2.8.1 ECC 模块特点

ECC 生成由控制寄存器的 ECC 锁位（主 ECC 锁、空闲 ECC 锁）来控制。

6.2.8.2 ECC 寄存器配置（大小端）

(1) 16 位 NAND Flash 存储器接口

寄存器	位[31:24]	位[23:16]	位[15:8]	位[7:0]
NFMECCD0	2 nd ECC for I/O[15:8]	2 nd ECC for I/O[7:0]	1 st ECC for I/O[15:8]	1 st ECC for I/O[7:0]
NFMECCD1	4 th ECC for I/O[15:8]	4 th ECC for I/O[7:0]	3 rd ECC for I/O[15:8]	3 rd ECC for I/O[7:0]

寄存器	位[31:24]	位[23:16]	位[15:8]	位[7:0]
NFSECCD	2 nd ECC for I/O[15:8]	2 nd ECC for I/O[7:0]	1 st ECC for I/O[15:8]	1 st ECC for I/O[7:0]

(2) 8 位 NAND Flash 存储器接口

寄存器	位[31:24]	位[23:16]	位[15:8]	位[7:0]
NFMECCD0	-	2 nd ECC for I/O[7:0]	-	1 st ECC for I/O[7:0]
NFMECCD1	-	4 th ECC for I/O[7:0]	-	3 rd ECC for I/O[7:0]

寄存器	位[31:24]	位[23:16]	位[15:8]	位[7:0]
NFSECCD	-	2 nd ECC for I/O[7:0]	-	1 st ECC for I/O[7:0]

6.2.8.3 ECC 编程指引

(1) 在软件模式下，ECC 模块为所有的读写数据生成 ECC 奇偶码。因此你通过写 InitECC 位（NFCONT[4]）置 1 来重置 ECC 的值，且在读写数据之前清 0 theMainECCLock 位（NFCONT[15]）。

(2) 不管读还是写数据，ECC 模块在寄存器 NFMECC0/1 上生成 ECC 奇偶码。

(3) 在你完整的读写一页后（不包括空闲区域数据），theMainECCLock 位置 1（锁定）。ECC

奇偶码被锁定且 ECC 状态寄存器的值不会被改变。

(4) 生成空闲区域 ECC 奇偶码，清 0（非锁定）SpareECCLock 位（NFCONT[6]）

- (5) 不管读还是写数据，空闲区域 ECC 模块在寄存器 NFSECC 上生成 ECC 奇偶码
- (6) 在你完整的读写空闲区域后，SpareECCLock 位置 1（锁定）。ECC 奇偶码被锁定且 ECC

状态寄存器的值不会被改变

- (7) 一旦完成你可以使用这些值去记录到空闲区域或检查位错误。
- 注：NFSECCD 是为空闲区域的 ECC 服务（通常，用户写主数据区域的 ECC 值到空闲区域，这些值和 NFMECC0/1 中的值一样）且从主数据区域中生成。

6.2.9 NAND Flash 存储器分布

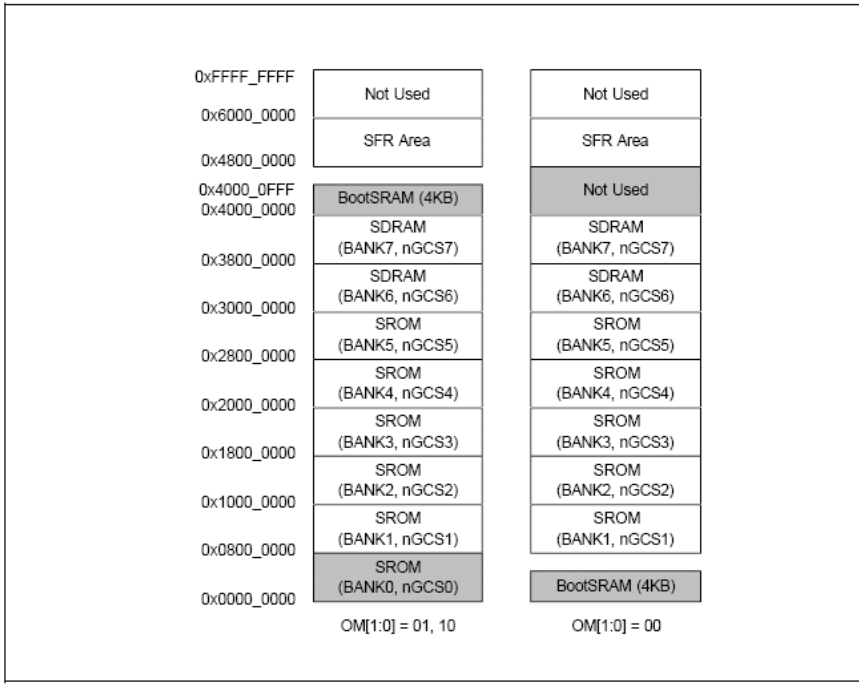


Figure 6-5. NAND Flash Memory Mapping

注：SROM 意为 ROM 和 SRAM 类型存储器。

6.2.10 NAND Flash 存储器配置

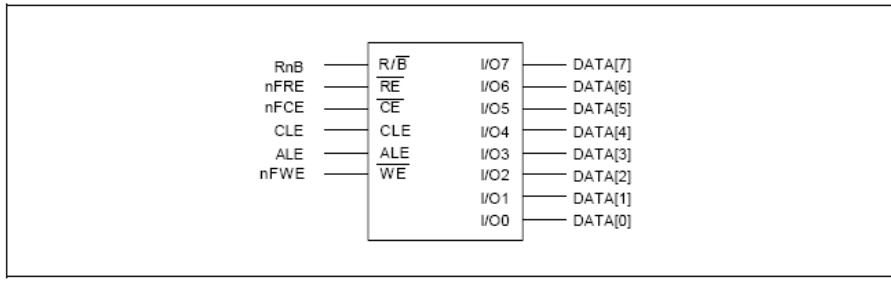


Figure 6-6. A 8-bit NAND Flash Memory Interface

当你写地址时，data[15:8]和 data[7:0]写同样的地址。

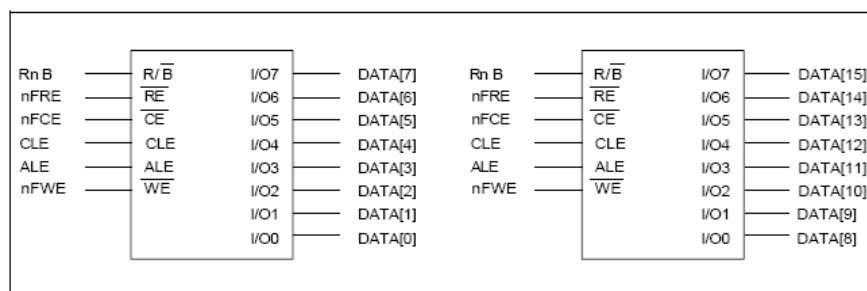


Figure 6-7. Two 8-bit NAND Flash Memory Interface

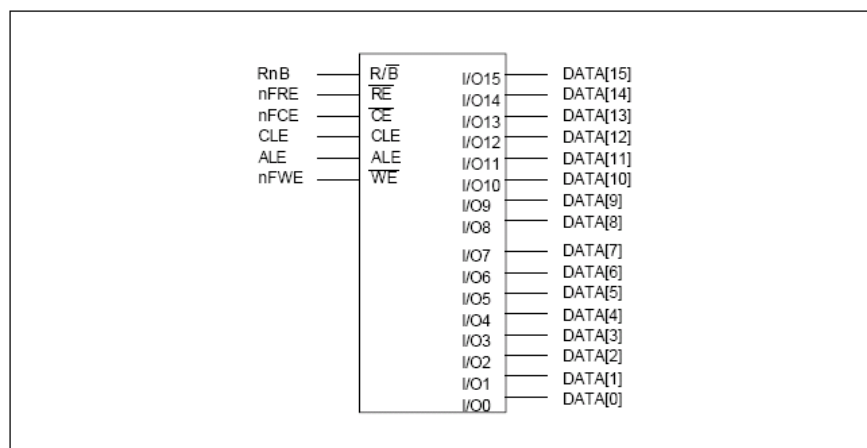


Figure 6-8. A 16-bit NAND Flash Memory Interface

6.3 NADN Flash 寄存器

- (1) NADN Flash 配置寄存器 (NFCONF)
- (2) NADN Flash 控制寄存器 (NFCONT)
- (3) NADN Flash 命令寄存器 (NFCMMD)
- (4) NADN Flash 地址寄存器(NFADDR)
- (5) NADN Flash 数据寄存器 (NFDATA)
- (6) NADN Flash 主数据区域 ECC 寄存器 (NFMECCD0/1)
- (7) NADN Flash 空闲区域 ECC 寄存器 (NFSECCD)
- (8) NADN Flash 操作状态寄存器(NFSTAT)
- (9) NADN Flash ECC0/1 状态寄存器 (NFESTAT0/1)
- (10) NADN Flash 主数据区域ECC状态寄存器 (NFMECC)
- (11) NADN Flash 空闲区域ECC状态寄存器 (NFSECC)
- (12) NADN Flash 块地址寄存器(NFSBLK &NFEBLK)

6.3.1 NADN Flash 配置寄存器

NAND FLASH CONFIGURATION REGISTER (NFCONF)

寄存器	地址	读写	描述	复位值
NFCONF	0x4E000000	R/W	NAND Flash配置寄存器	0x0000100X

NFCNF	位	描述	初始值
保留	[15:14]	保留	-
TACLS	[13:12]	CLE & ALE duration setting value (0~3) Duration = HCLK x TACLS	01
保留	[11]	保留	0
TWRPH0	[10:8]	TWRPH0 duration setting value (0~7) Duration = HCLK x (TWRPH0 + 1)	000
保留	[7]	保留	0
TWRPH1	[6:4]	TWRPH1 duration setting value (0~7) Duration = HCLK x (TWRPH1 + 1)	000
AdvFlash (只读)	[3]	Advance NAND flash memory for auto-booting 0: Support 256 or 512 byte/page NAND flash memory 1: Support 1024 or 2048 byte/page NAND flash memory This bit is determined by NCON0 pin status during reset and wake-up from sleep mode.	H/W Set (NCON0)
PageSize (只读)	[2]	NAND flash memory page size for auto-booting AdvFlash PageSize When AdvFlash is 0, 0: 256 Word/page, 1: 512 Bytes/page When AdvFlash is 1, 0: 1024 Word/page, 1: 2048 Bytes/page This bit is determined by GPG13 pin status during reset and wake-up from sleep mode. After reset, the GPG13 can be used as general I/O port or External interrupt.	H/W Set (GPG13)
AddrCycle (只读)	[1]	NAND flash memory Address cycle for auto-booting AdvFlash AddrCycle When AdvFlash is 0, 0: 3 address cycle 1: 4 address cycle When AdvFlash is 1, 0: 4 address cycle 1: 5 address cycle This bit is determined by GPG14pin status during reset and wake-up from sleep mode. After reset, the GPG14can be used as general I/O port or External interrupt.	H/W Set (GPG14)
BusWidth (读写)	[0]	NAND Flash Memory I/O bus width for auto-booting and general access. 0: 8-bit bus 1: 16-bit bus This bit is determined by GPG15 pin status during reset and wake-up from sleep mode. After reset, the GPG15 can be used as general I/O port or External interrupt. This bit can be changed by software.	H/W Set (GPG15)

6.3.2 控制寄存器

CONTROL REGISTER (NFCNT)

寄存器	地址	读写	描述	复位值
NFCNT	0x4E000004	R/W	NAND Flash控制寄存器	0x0384

NFCNT	位	描述	初始值
保留	[15:14]	保留	0
Lock-tight	[13]	Lock-tight configuration 0: Disable lock-tight 1: Enable lock-tight, Once this bit is set to 1, you cannot clear. Only reset or wake up from sleep mode can make this bit disable (cannot cleared by software). When it is set to 1, the area setting in NFSBLK (0x4E000038) to NFEBLK (0x4E00003C)-1 is unlocked, and except this area, write or erase command will be invalid and only read command is valid. When you try to write or erase locked area, the illegal a	0

第七章 时钟及电源管理模块

时钟电源管理模块包含了 3 部分: Clock 控制、USB 控制、POWER 控制。

时钟控制逻辑单元能够产生 s3c2440 需要的时钟信号, 包括 CPU 使用的主频 FCLK, AHB 总线设备使用的 HCLK, 以及 APB 总线设备使用的 PCLK. 2440 内部有 2 个 PLL (锁相环): 一个对应 FCLK、HCLK、PCLK, 另外一个对应的是 USB 使用 (48MHz)。时钟控制逻辑单元可以在不使用 PLL 情况下降低时钟 CLOCK 的频率, 并且可以通过软件来驱使时钟和各个模块的连接/断开, 以减少电源消耗。

对于电源控制逻辑单元, 2440 有许多钟电源管理方案来针对所给的任务保持最优的电源消耗。S3c2440 中的电源管理模块对应 4 种模式: NORMAL 模式、SLOW 模式, IDLE 模式, SLEEP 模式。

NORMAL 模式: 这个模块给 CPU 时钟以及 2440 相应的外围设备提供时钟。这个模式下, 当所有的外围设备都被打开, 电源消耗被最大化。它允许用户通过软件来控制外部设备的操作。例如, 如果一个定时器不需要时, 那么用户可以通过 CLKCON 寄存器来关闭时钟和定时器的连接, 来降低电源消耗。

SLOW 模式: NON-PLL 模式, 不同于 Normal 模式, 这个模式使用的一个外部时钟 (XT1p11 或 EXTCLK) 来直接作为 2440 的主频 FCLK, 而没有使用 PLL。在这个模式下, 电源的消耗仅依赖于外部时钟频率, 电源同 PLL 有关的消耗可以被排除。

IDLE 模式: 这个模式下 CPU 的时钟 FCLK 被断开, 而还继续提供其他外围设备的时钟。因此空闲模式导致减少了 CPU 核相应的电源消耗。任何中断请求都能够将 CPU 唤醒。

Sleep 模式: 这个模式断开了内部电源。因此在这个模式下 CPU & 内部的逻辑单元都没有电源消耗, 除了一个 wake-up 逻辑单元。激活 sleep 模式需要 2 个独立的电源。一个给 wake-up 逻辑模块提供电源, 另外一个给内部逻辑包括 CPU 提供电源, 并且其是对于 power on/off 可控的。在 Sleep 模式, 提供给内部逻辑 & CPU 的电源模块将被关闭, 而从 Sleep 模式唤醒可以通过 EINT[15:0] & RTC 中断来引发。

7.1 功能描述

7.1.1 时钟结构

如图 7-1 所示时钟结构模块图。主时钟源来自外部晶振 (XTIPLL) 或者是外部时钟 (EXTCLK)。时钟生成器包含了一个振荡器 (振荡放大器), 其连接外部晶振, 并且还有 2 个 PLL, 可以产生满足 s3c2440 所需的高频时钟。

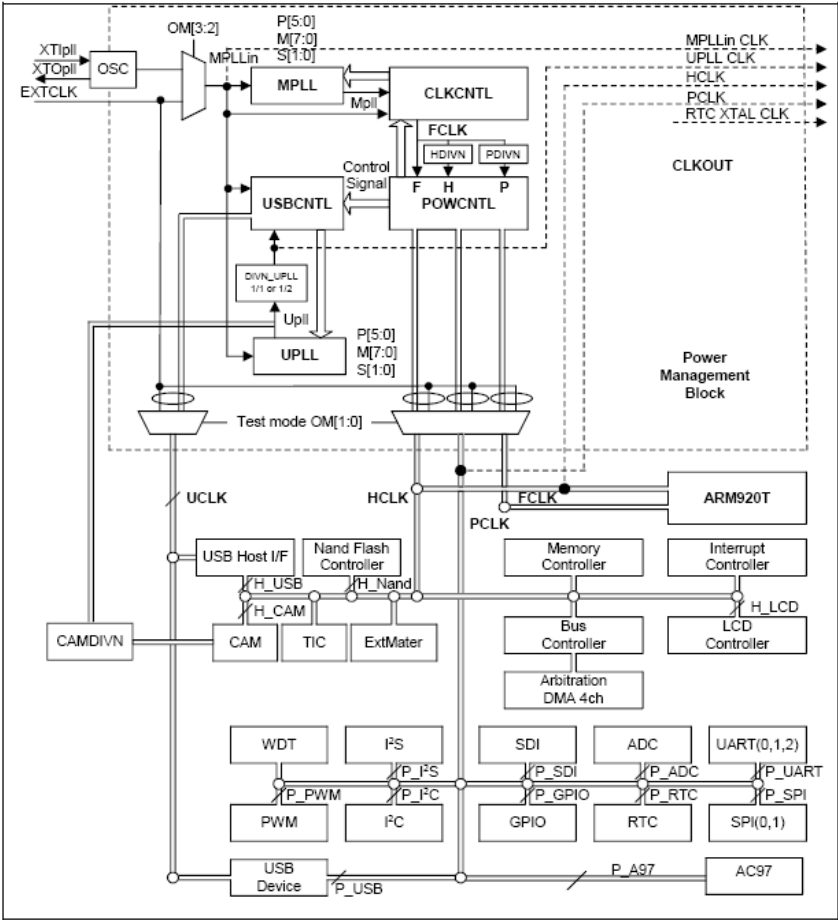


Figure 7-1. Clock Generator Block Diagram

7.1.2 时钟源的选择

如表 7-1 所示模式选择引脚(OM2 & OM3)组合之间的关系以及为 2440 选择时钟源，在 nRESET 上升沿，通过参考引脚 OM3 和 OM2，OM[3:2]的状态被锁存。

Mode OM[3:2]	MPLL State	UPLL State	Main Clock source	USB Clock Source
00	On	On	Crystal	Crystal
01	On	On	Crystal	EXTCLK
10	On	On	EXTCLK	Crystal
11	On	On	EXTCLK	EXTCLK

注意

- 1、虽然 MPLL 在 reset 以后就开始工作，但是 MPLL 输出（Mpll）没有作为系统时钟来使用，直到软件对 MPLLCON 寄存器写入有效设置。有效设置前，来自外部晶振或外部时钟源的时钟将直接被当作系统时钟来使用。就算是用户不想改变 MPLLCON 寄存器的默认值，用户应该再重新写入 MPLLCON 寄存器一次相同的值。
- 2、当 OM[1:0]=11 时，OM[3:2]被用于决定 test 模式。

7.1.3 锁相环

Phase Locked Loop (PLL)

位于时钟发生器中的 MPLL，作为一个集成电路，使得输出信号在频率和相位上同步于一个参考输入信号。应用上其包括如图 7-2 所示以下基本模块：其产生于输入直流电压成比例的输出频率 VCO（电压控制振荡器）；通过 P 值来对输入频率（Fin）来进行分频的分频器 P；通过 m 值来对 VCO 的输出频率进行分频的分频器 M，其输出频率被输入到一个相位频率侦测器（PFD）；通过 s 值来对 VCO 的输出频率进行分频的分频器 S，其分频器 S 的输出是 Mpll(从 Mpll 模块的输出频率)；鉴相器；电荷泵；环路滤波器。

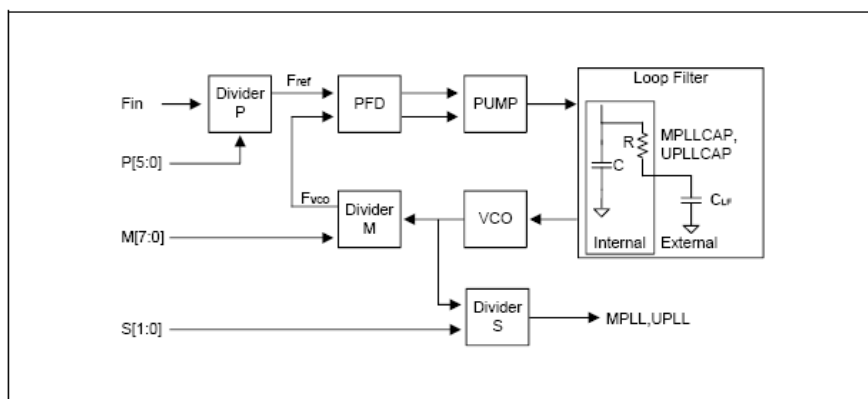


Figure 7-2. PLL (Phase-Locked Loop) Block Diagram

输出时钟频率 Mpll 相对于参考输入时钟频率 Fin 如以下公式所示：

$$M_{pll} = (2 * m * F_{in}) / (p * 2^s)$$

$$m = M (\text{分频器 M 的值}) + 8, \quad p = P (\text{分频器 P 的值}) + 2$$

Upll 时钟发生器的内部也和 Mpll 类似

7.1.3.1 鉴相器

Phase Frequency Detector (PFD)

PFD 的作用是监测 Fref 和 Fvco 的相位差，然后当检测到相位差时产生一个控制信号（跟踪型号）。Fref 意为参考频率，如图 7-2 所示。

7.1.3.2 电荷泵

Charge Pump (PUMP)

电荷泵通过一个由 VCO 驱动的外部滤波器将 PFD 的控制信号转变成一定电压的比例变化。

7.1.3.3 环路滤波器

Loop filter

PFD 为电荷泵生成的控制信号，通过每次 Fvco 与 Fref 比较，可能产生很大的偏移（纹波）。因此为了避免 VCO 过载，低通滤波采样以及滤除控制信号的高频分量。滤波器采用由一个电阻和电容组成的典型单极 RC 滤波器。

7.1.3.4 电压控制振荡器

Voltage Controlled Oscillator (VCO)

VCO 驱动的环路滤波器的输出电压,引起其振荡频率按均值变化线性的增加或减少。当 Fref 和 Fvco 在频率和相位上匹配，PFD 停止发送控制信号到电荷泵，其依次稳定给环路滤波器

的输入电压。VCO 频率保持稳定, PLL 保持固定为系统时钟。

7.1.4 通用条件和时钟发生器

PLL&时钟发生器需要如下硬件满足:

Loop filter capacitance	C _{LF}	MPLLAP: 1.3 nF ± 5% UPLLAP: 700 pF ± 5%
External X-tal frequency	—	12 – 20 MHz (note)
External capacitance used for X-tal	C _{EXT}	15 – 22 pF

注: 1.值可变

2.FCLK 必须大于 200MHz

7.1.5 时钟控制逻辑

时钟控制逻辑决定哪个时钟源被使用, mppll 还是直接使用外部时钟(XTIpll or EXTCLK)。当 PLL 配置新频率值时, 时钟控制逻辑使 Fclk 无效直到 PLL 输出使用 PLL 锁定时间达到稳定。时钟控制逻辑单元在上电 reset 和 power-down 模式被唤醒时被激活。

7.1.5.1 上电重启 Power-on Reset

如图 7-4 所示在电源上电 reset 顺序中的时钟行为。晶振在几毫秒内开始振荡, 当在晶振时钟稳定后 nReset 被释放, PLL 开始参照默认的 PLL 配置运行。但是, PLL 通常在电源启动 reset 后都不稳定, 因此在软件没有配置 PLLCON 之前, Fin 直接代替 MPLL 作为 FCLK。注意, 用户应该通过软件给 PLLCON 寄存器重写一遍相同的值。

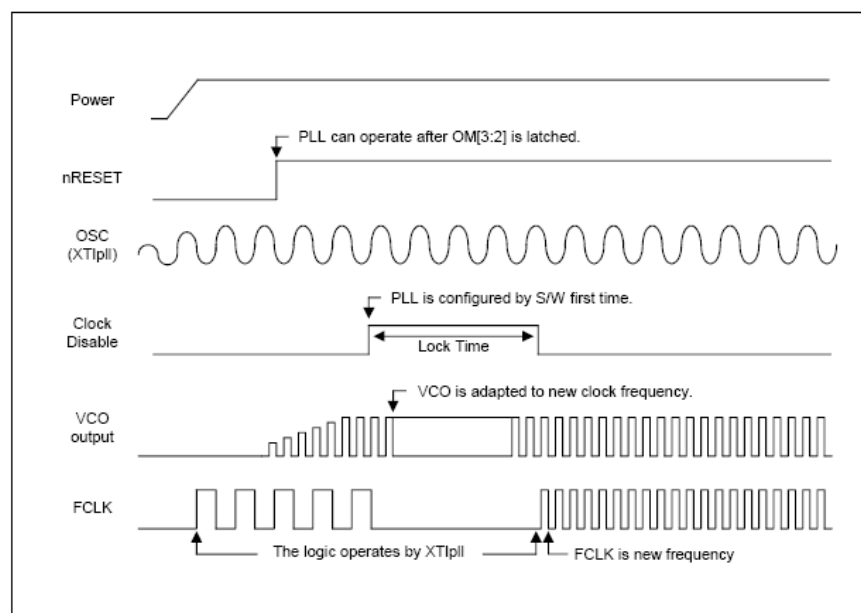


Figure 7-4. Power-On Reset Sequence (when the external clock source is a crystal oscillator)

仅在软件用新频率配置 PLL, PLL 重启一个锁定序列逼近一个新频率。FCLK 能在锁定时间后迅速被配置为 PLL 输出 (MPLL)。

7.1.5.2 正常模式下改变 PLL 设置 Change PLL Settings In Normal Operation Mode

在 Normal 模式下的 s3c2440 操作中, 用户可以改变频率通过写 P M S 三个分频器的值, 在 PLL 锁存时间会被自动的插入。在锁定时间, 时钟没有被提供给 s3c2440A 的内部时钟。图 7-5 所示的时序图。

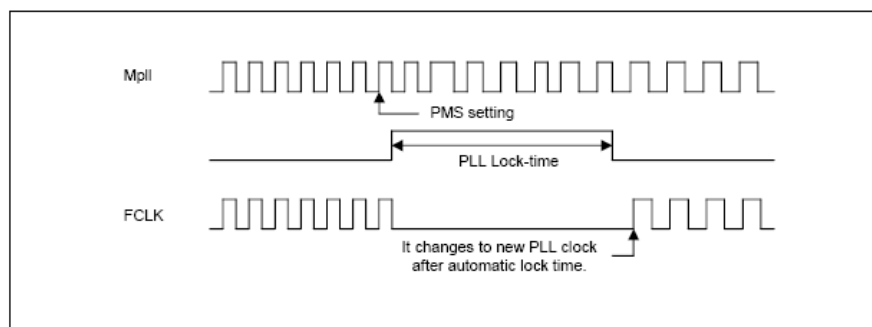


Figure 7-5. Changing Slow Clock by Setting PMS Value

7.1.6 USB 时钟控制

USB 主接口和 USB 从接口都需要 48Mhz 的时钟。在 S3C2440A 内，USB 知明需要 PLL(upll)产生 48Mhz 的时钟给 USB，UCLK 直到 PLL(UPLL)被配置才可以提供。

Condition	UCLK State	UPLL State
After reset	XTIpll or EXTCLK	On
After UPLL configuration	L : during PLL lock time 48MHz: after PLL lock time	On
UPLL is turned off by CLKSLW register	XTIpll or EXTCLK	Off
UPLL is turned on by CLKSLW register	48MHz	On

7.1.7 FCLK HCLK 和 PCLK

- FCLK 用于 ARM920T
- HCLK 用于 AHB 总线，例如用于 ARM920T,内存控制，中断控制，LCD 控制，DMA 以及 USB 主模块。
- PCLK 用于 APB 总线，用于外围设备如看门狗，IIS,I2C,PWM,MMC 接口，ADC UART,GPIO,RTC 以及 SPI。

2440 支持 FCLK,HCLK,PCLK 分频比的选择。比例由 CLKDIVN 控制寄存器中的 HDIVN,PDIVN 位确定。

HDIVN	PDIVN	HCLK3_HALF/ HCLK4_HALF	FLCK	HCLK	PCLK	分频比
0	0	-	FLCK	FLCK	FLCK	1:1:1
0	1	-	FLCK	FLCK	FLCK/2	1:1:2
1	0	-	FLCK	FLCK/2	FLCK/2	1:2:2
1	1	-	FLCK	FLCK/2	FLCK/4	1:2:4
3	0	0/0	FLCK	FLCK/3	FLCK/3	1:3:3
3	1	0/0	FLCK	FLCK/3	FLCK/6	1:3:6
3	0	1/0	FLCK	FLCK/6	FLCK/6	1:6:6
3	1	1/0	FLCK	FLCK/6	FLCK/12	1:6:12
2	0	0/0	FLCK	FLCK/4	FLCK/4	1:4:4
2	1	0/0	FLCK	FLCK/4	FLCK/8	1:4:8
2	0	0/1	FLCK	FLCK/8	FLCK/8	1:8:8
2	1	0/1	FLCK	FLCK/8	FLCK/16	1:8:16

在设置 PMS 3 个值后，需要设置 CLKDIVN 寄存器的值。CLKDIVN 的值设置在 PLL 锁定期后有效。这个值也对复位和改变电源管理模式也是可用的。

设置值在 1.5HCLK 后有效，1 个 HCLK 可以 CLKDIVN 寄存器的值从默认（1：1：1）到其他分频比（1：1：2，1：2：2，1：2：4）改变有效。

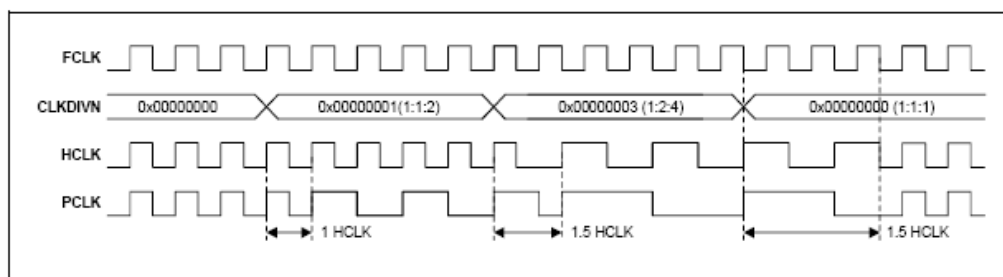


Figure 7-6. Example of internal clock change

注意

- 1、CLKDIVN 的值需要小心设置，不能超过 HCLK,PCLK 的极限值。
- 2、如果 HDIVN 不为 0，根据如下指令，CPU 总线模式从 Fast Bus Mode 变为 Asynchronous（异步总线模式）

MMU_setAsyncBusMode

mrc p15,0,r0,c0,0 ; 协处理器指令

orr r0, r0, #R1_nF : OR: R1_iA

mcr P15, 0, r0, c1, c0, 0

如果 HDIVN 不为 0，且 CPU 总线模式为 Fast Bus mode,CPU 的时钟为 HCLK.,这种方式可以用在将 CPU 频率降低，但是却又不改变 HCLK 和 PCLK.

7.2 电源管理

2440 的电源管理模块通过软件控制系统时钟来降低各个模块的电源消耗。这些方案与 PLL, 时钟控制逻辑(FCLK,HCLK,PCLK),以及唤醒信号相关。如图 7-7 所示系统时钟的结构图。

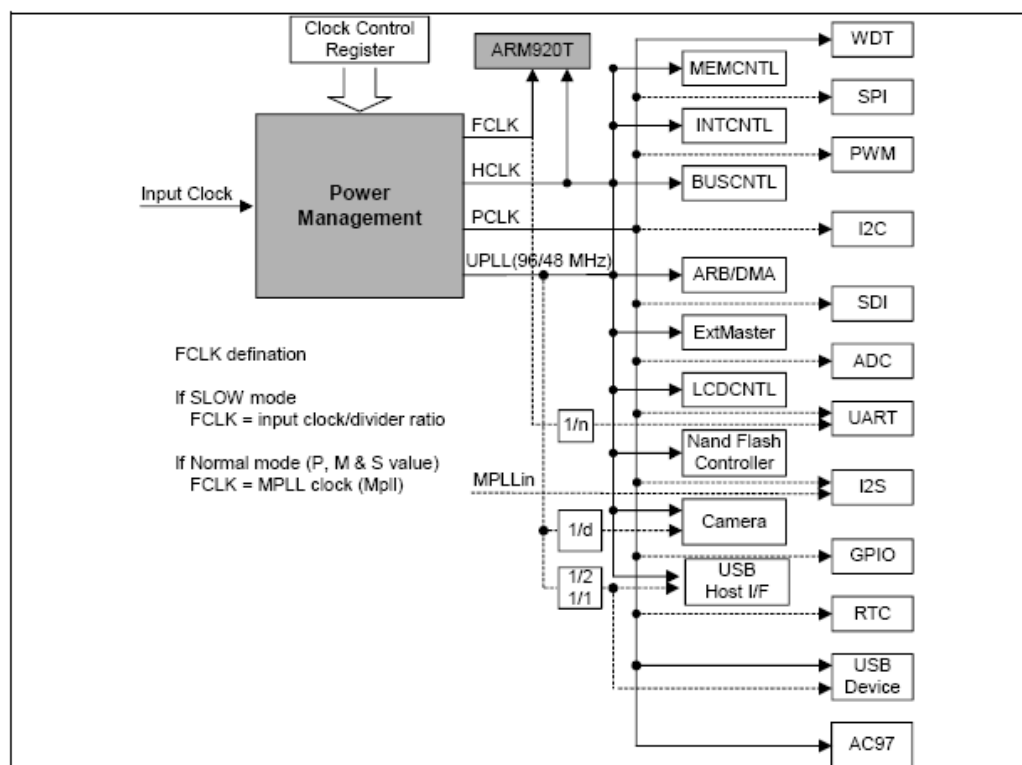


Figure 7-7. The Clock Distribution Block Diagram

2440 有 4 种电源模式，下面描述每个电源管理模式，各个模式并不能自由转换，有条件。查看图 7-8 所示各种模式之间的有效转换。

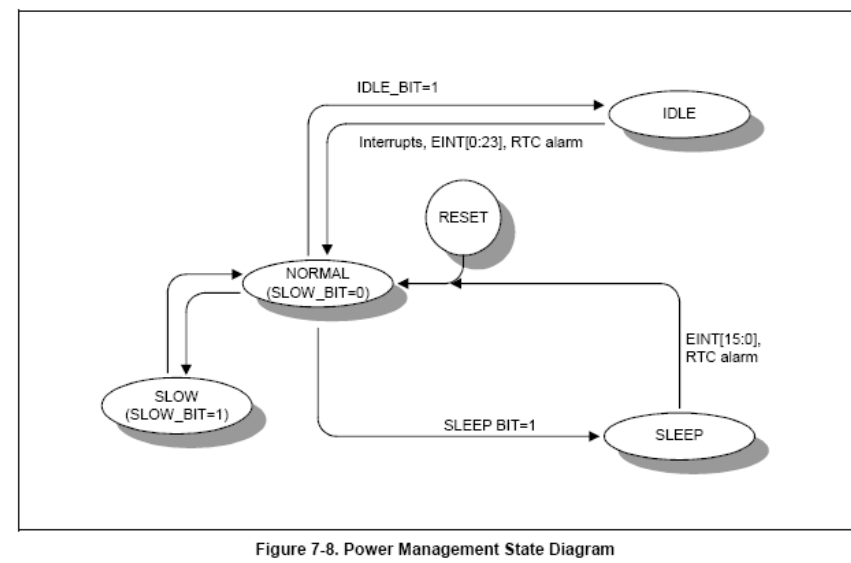


Figure 7-8. Power Management State Diagram

表 7-2 每个模式下的时钟电源状态

模式	ARM920T	AHB 模块/ 看门狗	电源 管理	GPIO	3.2768KHz RTC 时钟	APB 模块& USBH/LCD/NAND
NORMAL	开启	开启	开启	选择	开启	选择
IDLE	关闭	开启	开启	选择	开启	选择
SLOW	开启	开启	开启	选择	开启	选择
SLEEP	关闭	关闭	等待唤 醒事件	先前 状态	开启	关闭

7.2.1 NORMAL Mode

在 Normal 模式,所有的外围设备以及基本模块，包括电源管理模块，CPU 核，总线控制器，内存控制器，中断控制器，DMA，以及外部主控都可以完整的操作。但是，除了基本模块,每个外围设备的时钟都能通过软件来设置关闭以降低电源消耗。

7.2.2 IDLE Mode

在 IDLE 模式中，对 CPU 核的时钟被关闭了，总线控制器，内存控制器，中断控制器和电源管理模块除外。为了退出 IDLE 模式， EINT[23:0]或者 RTC 时钟中断，或者其他的中断应该被激活(EINT 无效直到 GPIO 模块被开启)

7.2.3 SLOW 模式(非 PLL 模式)

这个模式可以减少电源消耗，通过提供一个慢速时钟且关闭了 PLL 模块电源供应。FCLK 不使用 PLL,是输入时钟(XTIp1l 和 EXTCLK)n 分频的频率。分频比率由 CLKSLOW 控制寄存器的 SLOW_VAL 位和 CLKDIVN 控制寄存器来决定。具体如表 7-3 所示：

表 7-3 CLKSLOW 和 CLKDIVN 寄存器对于减慢时钟的例子

SLOW_VAL	FCLK	HCLK		PCLK		UCLK
		1/1 Option (HDIVN=0)	1/2 Option (HDIVN=1)	1/1 Option (PDIVN=0)	1/2 Option (PDIVN=1)	
000	EXTCLK or XTIpll /1	EXTCLK or XTIpll /1	EXTCLK or XTIpll /2	HCLK	HCLK/2	48MHz
001	EXTCLK or XTIpll /2	EXTCLK or XTIpll /2	EXTCLK or XTIpll /4	HCLK	HCLK/2	48MHz
010	EXTCLK or XTIpll /4	EXTCLK or XTIpll /4	EXTCLK or XTIpll /8	HCLK	HCLK/2	48MHz
011	EXTCLK or XTIpll /6	EXTCLK or XTIpll /6	EXTCLK or XTIpll /12	HCLK	HCLK/2	48MHz
100	EXTCLK or XTIpll /8	EXTCLK or XTIpll /8	EXTCLK or XTIpll /16	HCLK	HCLK/2	48MHz
101	EXTCLK or XTIpll /10	EXTCLK or XTIpll /10	EXTCLK or XTIpll /20	HCLK	HCLK/2	48MHz
110	EXTCLK or XTIpll /12	EXTCLK or XTIpll /12	EXTCLK or XTIpll /24	HCLK	HCLK/2	48MHz
111	EXTCLK or XTIpll /14	EXTCLK or XTIpll /14	EXTCLK or XTIpll /28	HCLK	HCLK/2	48MHz

在 SLOW 模式，PLL 将被关闭降低 PLL 的电源消耗。在 SLOW 模式下，PLL 被关闭，用户要从 SLOW 模式变成 NORMAL 模式，PLL 需要时钟稳定时间（PLL Lock time）。PLL 稳定时间可以通过内部逻辑借助锁定时间计数寄存器自动被插入。PLL 稳定时间在 PLL 开启后花费大概 300us 左右。在 PLL 锁定期间内，FCLK 成了 SLOW 时钟。

在 PLL on 态，用户可以使能 CLKSLOW 寄存器中的 SLOW 模式位来改变频率。SLOW 时钟在 SLOW 模式期间被生成，如图 7-9 所示时序图：

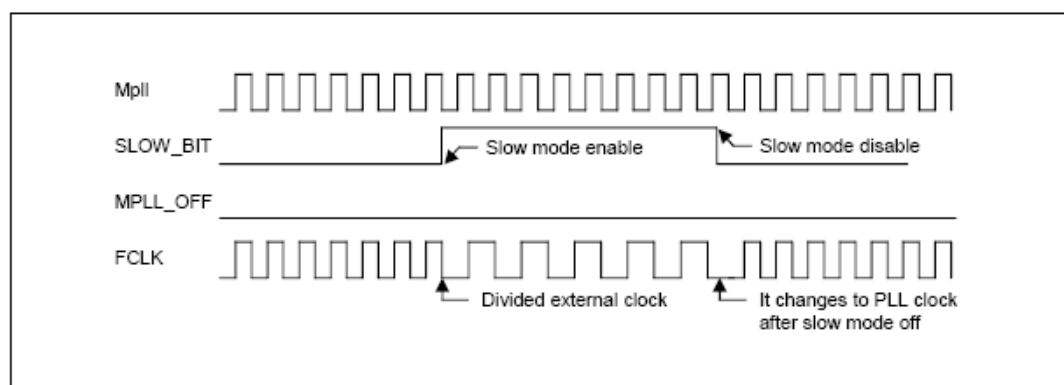


Figure 7-9. Issuing Exit_from_Slow_mode Command in PLL on State

如果用户在 PLL 锁定期后通过使 CLKSLOW 寄存器中的 SLOW_BIT 位无效，从 SLOW 模式切换至 NORMAL 模式，频率在 SLOW 模式无效后立即改变，如图 7-10 所示时序图：

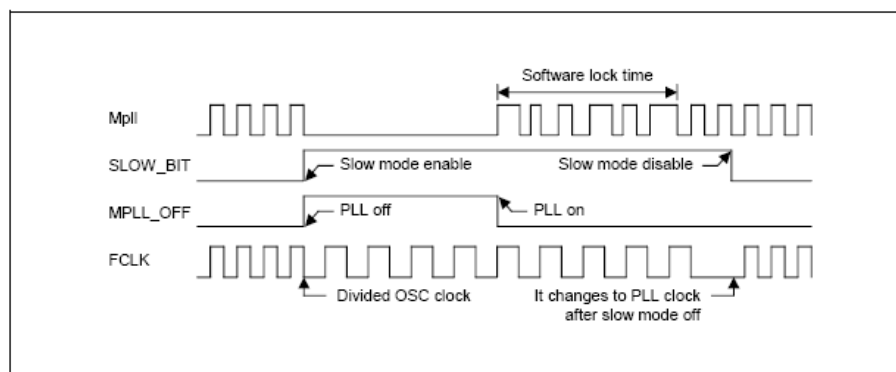


Figure 7-10. Issuing Exit_from_Slow_mode Command After Lock Time

如果用户通过使 CLKSLOW 寄存器中的 SLOW_BIT 和 MPLL_OFF 位同时无效, 从 SLOW 模式切换到 NORMAL 模式, 频率在 PLL 锁定时间后立即改变, 如图 7—13 所示时序图:

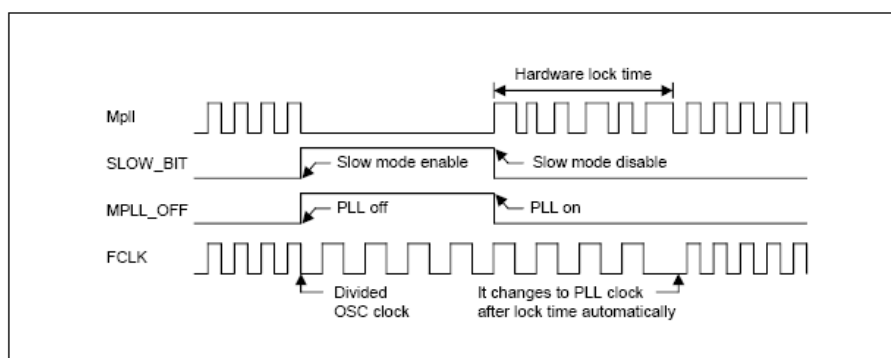


Figure 7-11. Issuing Exit_from_Slow_mode Command and the Instant PLL_on Command Simultaneously

7.2.4 SLEEP 模式

电源管理模块关闭内部电源, 因此, CPU 和内部逻辑模块都没有电源消耗, 除了此模式下的唤醒模块。激活 SLEEP 模式需要 2 个独立的电源, 其中一个给唤醒模块提供电源, 另一个给包括 CPU 的其他逻辑模块提供电源, 并且应该可以由 power on/off 控制。在 SLEEP 模式下, 给 CPU 和内部逻辑单元提供电源的第二个电源被关闭。只有唤醒模块是工作的, 睡眠模式被唤醒可以由 EINT[15:0]和定时器中 RTC(real time control)中断引起。

7.2.4.1 依据以下步骤进入到睡眠模式:

- 1、设置 GPIO 配置满足 SLEEP 模式。ConfigSleepGPIO();
- 2、在 INTMSK 寄存器设置屏蔽所有中断。rINTMSK=BIT_ALLMSK; 屏蔽所有中断;
- 3、正确设置唤醒源包括 RTC alarm (唤醒源的在 EINTMASK 中对应位不必被屏蔽,目的是使得 SRCPND 和 EINTPEND 的相应位 置 1。尽管一个唤醒源引发, 而 EINTMASK 相应位被屏蔽, 唤醒也将出现但 SRCPND 和 EINTPEND 的相应位不会被置 1)
`rEINTMASK = rEINTMASK & ~(1<<11); //SRCPND:EINT8_23 will be set by EINT11 after wake-up.`
- 4、设置 USB 挂起模式(MISCCR[13:12]=11b)
`rMISCCR |= (1<<12); //USB port0 = suspend`
`rMISCCR |= (1<<13); //USB port1 = suspend`
- 5、保存有含义的值到 GSTATUS[4:3]寄存器。这些寄存器在 SLEEP 期间被保留。
`rGSTATUS3=(U32)StartPointAfterSleepWakeUp; //复位后起始点的地址`
- 6、为数据总线上的上拉电阻配置 MISCCR[1:0]。如果有外部总线保持器, 如 74LVCH162245, 关闭上拉电阻; 如果没有, 开启上拉电阻。另外内存相关的引脚设置成

两态，一个是高阻，另一个是其他非活动状态。

```
*****rMISCCR=rMISCCR| (3<<0);
```

```
*****rMSLCON
```

=

```
(1<<11)|(0<<10)|(0<<9)|(1<<8)|(1<<7)|(1<<6)|(1<<5)|(1<<4)|(1<<3)|(1<<2)|(1<<1)|(1<<0);
```

查看 P284 寄存器可以看此设置

7、清除 LCDCON1.ENVID 位来使 LCD 停止工作。

```
*****rLCDCON1 =rLCDCON1 & 0x3ffe;
```

8、读取 rREFRESH 和 rCLKCON 寄存器以填充 TLB 位

9、设置 REFRESH[22]=1b，使能 SDRAM 进入自刷新模式。

10、等待直到 SDRAM 自刷新有效。

11、设置 MISCCR[19:17]=111b 使得 SDRAM 的信号(SCLK0,SCLK1,SCLKE)在睡眠模式期间被保护。

12、设置 CLKCON 寄存器的 SLEEP 位。

注意：当系统运行在 NAND boot 模式启动时，硬件引脚配置 EINT[23:21]必须，为 Sleep 模式唤醒的启动，设置为输入。

7.2.4.2 根据下面过程唤醒 SLEEP 模式

1、如果有唤醒源被触发，内部的复位信号就会动作。这和外部的 nReset 引脚触发非常相似。reset 复位持续时间由内部的 16 位计数器逻辑决定，reset 复位决断时间可以计算 $t_{RST}=(65535/XTAL_frequency)$

2、检测 GSTATUS2[2]来判断是否是由 sleep 模式唤醒引起的电源开启。

3、释放 SDRAM 的信号保护，通过设置 MISCCR[19:17]=000b

4、配置 SDRAM 内存控制器

5、等待直到 SDRAM 自刷新释放。大部分 SDRAM 需要所有 SDRAM 行的自刷新周期。

6、GSTATUS[3:4]的信息可用于用户自己目的，因为在 GSTATUS[3:4]中的值在睡眠模式下被保留。

7、-对 EINT[3:0]，检查 SRCPND 寄存器

-对 EINT[15:4]，查看 EINTPEND 寄存器而不是 SRCPND 寄存器。（尽管 EINTPEND 寄存器的一些位被置位，SRCPND 寄存器不会被置位）

表 7-4 在睡眠模式下引脚配置表

引脚条件		引脚配置指引
GPIO 引脚	配置为输入	上拉有效
	配置为输出	上拉无效输出低电平
输入引脚，无内部上拉控制	外部设备不能一直驱动引脚电平	上拉有效，通过外部上拉电阻
输出引脚，连接外部设备	外部设备电源关闭	输出低电平
	外部设备电源开启	高或低（根据外部设备状况）
数据总线	内存电源关闭	输出低电平
	内存电源开启	且外部缓存存在
	且无外部缓存	上拉有效，控制信号先前值

7.2.4.3 关于 VDDi&VDDiarm 的电源控制

在睡眠模式下，VDDi,VDDiarm,VDDMPLL 以及 VDDUPLL 会被关闭，其由 PWREN 引脚来控制，如 PWREN 信号被置位，VDDi 和 VDDiarm 由一个外部变压器供电。

当 $PWREN=0$ 时, $VDDi$ 和 $VDDiarm$ 被关闭

注意:

即使 $VDDi, VDDiarm, VDDMPLL$ 以及 $VDDUPLL$ 可能被关闭, 部分其他电源脚也可供电。
如图 7-12 所示

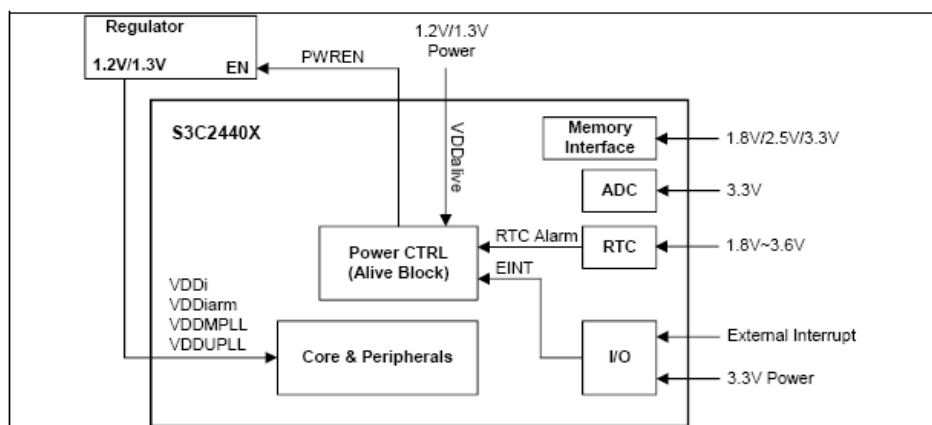


Figure 7-12. SLEEP Mode

在睡眠模式, 如果你不使用触摸屏, 4 个触摸接口必须悬空, 不能接地, 在睡眠模式下, XP, YP 都是高电平。

7.2.4.4 唤醒中断信号 EINT[15:0]

S3c2440 只有满足下面条件, 才能从 Sleep 模式中被唤醒

- a) $EINTn$ 引脚上的电平信号或跳变信号。
- b) GPIO 控制寄存器中配置 $EINTn$ 引脚为 EINT 外部中断功能。
- c) $nBATT_FLT$ 引脚必须为高电平, 这个在设置 GPIO 控制寄存器设置外部中断时候是非常重要的。

当唤醒后, 这些中断引脚 $EINTn$ 就不用作唤醒功能, 也就是说这些引脚又可以当作外部中断请求引脚。

7.2.4.5 Entering IDLE Mode

如果 $CLKCON[2]$ 设置为 1 进入空闲模式, 经过一段时间延时 2440 就进入空闲模式 (直到电源控制逻辑模块从 CPU 处收到一个 ACK 应答信号)。

7.2.4.6 PLL On/OFF

PLL 可以在 SLOW 模式下为降低电源功耗被关闭。如果 PLL 在其他任何模式下关闭, MCU 操作不能被保证。

当处理器在 SLOW 模式下且试图改变它的状态到其他态, PLL 为开启, 在 PLL 稳定后, $SLOW_BIT$ 应该被清除以转换到其他态。

7.2.4.7 PULL-up Resistors on the Data Bus and SLEEP Mode

在睡眠模式下, 数据总线($D[31:0]$ 或 $D[15:0]$)可以选择高阻或者输出低电平。

通过打开 pull-up 寄存器, 数据总线可以设置为高阻状态; 或通过关闭 pull-up 寄存器, 数据总线可以设置为输出低电平, 目的是在 SLEEP 模式下降低电源消。 $D[31:0]$ 引脚上拉电阻

能够由 GPIO 控制寄存器 (MISCCR) 控制。但是, 如果在数据总线上有一个外部总线保持器, 例如 74LVCH162245, 用户选择两态中的一个, 一个是关闭上拉, 输出低电平, 另一个是高阻, 上拉关闭, 这样可以消耗最少的电源。

7.2.4.8 Output Port State and SLEEP mode

输出引脚在 power off 模式下应该有一个适合逻辑电平, 使得电流消耗最小。如果输出引脚没有负载, 高电平比较好。如果输出是低电平, 电流会通过内部的寄生电阻被消耗。如果输出是高电平, 电流就不会被消耗。对于一个输出引脚, 如果输出态是高电平, 电流消耗可以被减少。

推荐把所有输出引脚都设为高电平态, 以减少在 SLEEP 模式下的电流消耗。

但是如果输出要低, 那么需要通过设置 DSC0,DSC1 来设置驱动电流, 举例如下:

```
*****rDSC0 = 0x7ffffff;  
*****rDSC0 &= ~(1<<31); //enable    调节内部的电阻阻值  
*****rDSC1 = (3<<28)|(3<<26)|(3<<24)|(3<<22)|(3<<20)|(3<<18);
```

7.2.4.9 Battery Fault Signal (nBATT_FLT)

nBATT_FLT 脚有 2 个功能, 如下

1、当 CPU 不是在 SLEEP 模式时 (大多数工作时候), nBATT_FLT 此脚将引起中断请求, 通过设置 BATT_FUNC(MISCCR[22:20])=10xb。其中断为低电平触发。

2、当 CPU 为 SLEEP 模式时, 设置此脚可以禁止从 SLEEP 模式的唤醒, 通过设置 BATT_FUNC(MISCCR[22:20])=11xb。因此, 任何一个唤醒源将被屏蔽如果 nBATT_FLT 被设置, 保护在低电池能力下系统故障。

7.3 时钟发生器及电源管理特殊寄存器

(Clock Generator & Power Management Special Register)

1. 锁定时间计数寄存器
LOCK TIME COUNT REGISTER (LOCKTIME): PLL lock time count register
2. 锁相环控制寄存器
PLL CONTROL REGISTER (MPLLCON & UPLLCON)
MPLLCON: MPLL configuration register
UPLLCON: UPLL configuration register
3. 时钟控制寄存器
CLOCK CONTROL REGISTER (CLKCON): Clock generator control register
4. 时钟减慢控制寄存器
CLOCK SLOW CONTROL REGISTER (CLKSLOW): Slow clock control register
5. 时钟分频控制寄存器
CLOCK DIVIDER CONTROL REGISTER (CLKDIVN): Clock divider control register
6. 摄像头时钟分频寄存器
CAMERA CLOCK DIVIDER REGISTER (CAMDIVN): Camera clock divider register

7.3.1 锁定时间计数寄存器

LOCK TIME COUNT REGISTER (LOCKTIME)

寄存器	地址	读写	描述	复位值
LOCKTIME PLL	0x4C000000	R/W	锁定时间计数寄存器	0xFFFFFFFF

LOCKTIME	位	描述	初始值
U_LTIME	[31:16]	UPLL 对于 UCLK 的锁定时间计数值. (U_LTIME: 300uS)	0xFFFF
M_LTIME	[15:0]	MPLL对于FCLK、HCLK、PCLK的锁定时间计数值 (M_LTIME: 300uS)	0xFFFF

7.3.2 锁相环控制寄存器

PLL CONTROL REGISTER (MPLLCON & UPLLCON)

MPLL Control Register

$$M_{pll} = (2 * m * F_{in}) / (p * 2^s)$$
$$m = (MDIV + 8), p = (PDIV + 2), s = SDIV$$

UPLL Control Register

$$U_{pll} = (m * F_{in}) / (p * 2^s)$$
$$m = (MDIV + 8), p = (PDIV + 2), s = SDIV$$

锁相环值选择指引 (MPLLCON)

1. $F_{out} = 2 * m * F_{in} / (p * 2^s)$, $F_{vco} = m * F_{in} / p$, $m = MDIV + 8$, $p = PDIV + 2$, $s = SDIV$
2. $600MHz \leq F_{VCO} \leq 1.2GHz$
3. $200MHz \leq F_{CLKOUT} \leq 600MHz$
4. 不能设置P和M值为零, 因为P=000000, M=00000000 会引起PLL故障。
5. 合适的P、M值范围是: $1 \leq P \leq 62$, $1 \leq M \leq 248$

寄存器	地址	读写	描述	复位值
MPLLCON	0x4C000004	R/W	MPLL 配置寄存器	0x00096030
UPLLCON	0x4C000008	R/W	UPLL 配置寄存器	0x0004d030

PLLCON	位	描述	初始值
MDIV	[19:12]	主分频器控制	0x96 / 0x4d
PDIV	[9:4]	预分频器控制	0x03 / 0x03
SDIV	[1:0]	后分频器控制	0x0 / 0x0

注: 当你设置MPLL和UPLL值的时候, 需要先设置UPLL再设置MPLL的值

PLL值选择表 PLL VALUE SELECTION TABLE

一般较难找到一个合适的PLL值, 因此推荐参考以下PLL推荐值表:

输入频率	输出频率	MDIV	PDIV	SDIV
12.0000MHz	48.00 MHz 注	56 (0x38)	2	2
12.0000MHz	96.00 MHz 注	56 (0x38)	2	1
12.0000MHz	271.50 MHz	173 (0xad)	2	2
12.0000MHz	304.00 MHz	68 (0x44)	1	1
12.0000MHz	405.00 MHz	127 (0x7f)	2	1
12.0000MHz	532.00 MHz	125 (0x7d)	1	1
12.0000MHz	47.98 MHz 注	60 (0x3c)	4	2
16.9344MHz	95.96 MHz 注	60 (0x3c)	4	1
16.9344MHz	266.72 MHz	118 (0x76)	2	2
16.9344MHz	296.35 MHz	97 (0x61)	1	2
16.9344MHz	399.65 MHz	110 (0x6e)	3	1
16.9344MHz	530.61 MHz	86 (0x56)	1	1
16.9344MHz	533.43 MHz	118 (0x76)	1	1

注: 48MHz和 96MHz输出是用于UPLL。

7.3.3 时钟控制寄存器

CLOCK CONTROL REGISTER (CLKCON)

寄存器	地址	读写	描述	复位值
CLKCON	0x4C00000C	R/W	时钟发生器控制寄存器	0xFFFFF0

CLKCON	位	描述	初始值
AC97	[20]	控制进入 AC97 模块的 PCLK 0: 无效, 1: 有效	1
Camera	[19]	控制进入 Camera 模块的 HCLK 0: 无效, 1: 有效	1
SPI	[18]	控制进入 SPI 模块的 PCLK 0: 无效, 1: 有效	1
IIS	[17]	控制进入 IIS 模块的 PCLK 0: 无效, 1: 有效	1
IIC	[16]	控制进入 IIC 模块的 PCLK 0: 无效, 1: 有效	1
ADC	[15]	控制进入 AC97 模块的 PCLK 0: 无效, 1: 有效	1
RTC	[14]	控制进入 AC97 模块的 PCLK 0: 无效, 1: 有效	1
GPIO	[13]	控制进入 AC97 模块的 PCLK 0: 无效, 1: 有效	1
UART2	[12]	控制进入 UART2 模块的 PCLK 0: 无效, 1: 有效	1
UART1	[11]	控制进入 UART1 模块的 PCLK 0: 无效, 1: 有效	1
UART0	[10]	控制进入 UART0 模块的 PCLK 0: 无效, 1: 有效	1
SDI	[9]	控制进入 SDI 模块的 PCLK 0: 无效, 1: 有效	1
PWMTIMER	[8]	控制进入 PWMTIMER 模块的 PCLK 0: 无效, 1: 有效	1
USB Device	[7]	控制进入 USB Device 模块的 PCLK 0: 无效, 1: 有效	1
USB Host	[6]	控制进入 USB Host 模块的 HCLK 0: 无效, 1: 有效	1
LCDC	[5]	控制进入 LCDC 模块的 HCLK 0: 无效, 1: 有效	1
NAND Flash Controller	[4]	控制进入 NAND Flash Controller 模块的 HCLK 0: 无效, 1: 有效	1
SLEEP	[3]	控制s3c2440 的睡眠模式 0: 无效, 1: 进入睡眠模式	0
IDLE BIT	[2]	进入空闲模式 0: 无效, 1: 进入空闲模式	0
Reserved	[1:0]	保留	0

7.3.4 时钟减慢控制寄存器

CLOCK SLOW CONTROL REGISTER (CLKSLOW)

寄存器	地址	读写	描述	复位值
CLKSLOW	0x4C000010	R/W	减慢时钟控制寄存器	0x00000004

CLKSLOW	位	描述	初始值
UCLK_ON	[7]	0: UCLK ON (UPLL 被打开并且自动插入UPLL锁定时间.) 1: UCLK OFF (UPLL 也被关闭.)	0
保留	[6]	保留	-
MPLL_OFF	[5]	0: Turn on PLL. 在PLL稳定时间（至少 300us）后，SLOW_BIT位可以被清零 1: Turn off PLL. 仅当SLOW_BIT是 1，PLL才被关闭	0
SLOW_BIT	[4]	0 : FCLK = Mpll (MPLL output) 1: SLOW 模式 -FCLK = input clock/(2xSLOW_VAL), 当SLOW_VAL>0 -FCLK = input clock, 当 SLOW_VAL=0. -Input clock = XTIpil or EXTCLK	0
保留	[3]	保留	-
SLOW_VAL	[2:0]	当SLOW_BIT位开启，该位表示减慢时钟的分频器值	0x4

7.3.5 时钟分频器控制寄存器

CLOCK DIVIDER CONTROL REGISTER (CLKDIVN)

寄存器	地址	读写	描述	复位值
CLKDIVN	0x4C000014	R/W	时钟分频器控制寄存器	0x00000000

CLKDIVN	位	描述	初始值
DIVN_UPLL	[3]	UCLK 选择寄存器(UCLK必须对USB提供 48MHz) 0: UCLK = UPLL clock 1: UCLK = UPLL clock / 2 设置为 0，当UPLL时钟被设置为 48Mhz。 设置为 1，当 UPLL 时钟被设置为 96Mhz。	0
HDIVN	[2:1]	00: HCLK = FCLK/1. 01: HCLK = FCLK/2. 10: HCLK = FCLK/4, 当 CAMDIVN[9] = 0. HCLK = FCLK/8, 当 CAMDIVN[9] = 1. 11: HCLK = FCLK/3, 当 CAMDIVN[8] = 0. HCLK = FCLK/6, 当 CAMDIVN[8] = 1.	00
PDIVN	[0]	0: PCLK 是和HCLK/1 相同的时钟。 1: PCLK 是和HCLK/2 相同的时钟。	0

7.3.6 摄像头时钟分频器寄存器

CAMERA CLOCK DIVIDER REGISTER (CAMDIVN)

寄存器	地址	读写	描述	复位值
CAMDIVN	0x4C000018	R/W	摄像头时钟分频器寄存器	0x00000000

CAMDIVN	位	描述	初始值
DIVN_UPLL	[12]	0:DVS OFF (FCLK = MPLL clock) 1:DVS ON (FCLK = HCLK)	0
保留	[11]	保留	0
保留	[10]	保留	0
HCLK4_HALF	[9]	HDIVN分频比率改变位, 当CLKDIVN[2:1]=10b. 0: HCLK = FCLK/4 1: HCLK= FCLK/8 参考CLKDIV寄存器	0
HCLK3_HALF	[8]	HDIVN分频比率改变位, 当CLKDIVN[2:1]=11b. 0: HCLK = FCLK/3 1: HCLK= FCLK/6 参考CLKDIV寄存器	0
CAMCLK_SEL	[4]	0:用UPLL output作为CAMCLK (CAMCLK=UPLL output). 1:CAMCLK 用CAMCLK_DIV的值分频.	0
CAMCLK_DIV	[3:0]	CAMCLK 分频参数设置寄存器(0-15). Camera clock = UPLL / [(CAMCLK_DIV + 1)x2]. 这个位有效, 当CAMCLK_SEL=1.	0

第八章 DMA

8.1 概述

S3c2440A 支持位于系统总线和外设总线之间的 4 个通道的控制器。每个 DMA 控制器通道无限制地执行系统总线上的设备或外设总线上的设备之间数据搬移。换句话说，就是每个通道都操作一下四种情况：

- (1) 源和设备都在系统总线上
- (2) 源设备在系统总线上，目的设备在外设总线上
- (3) 源设备在外设总线上，目的设备在系统总线上
- (4) 源设备和目的设备都在外设总线上

DMA 的主要有点就是其传输数据不需要 CPU 的干涉。DMA 操作可由软件或来自内设或外部请求引脚来初始化。

8.1.1 DMA 请求源

如果通过 DCON 寄存器 H/W DMA 的请求模式被选择，每个 DMA 控制器通道可以从四个 DMA 源中选择一个 DMA 请求源（如果 S/W 请求模式被选中，DMA 请求源就完全没有任何意义）。如表 8-1 所示对于每个通道的 4 个 DMA 源：

	Source0	Source1	Source2	Source3	Source4	Source5	Source6
Ch-0	nXDREQ0	UART0	SDI	Timer	USB device EP1	I2SSDO	PCMIN
Ch-1	nXDREQ1	UART1	I2SSDI	SPI0	USB device EP2	PCMOUT	SDI
Ch-2	I2SSDO	I2SSDI	SDI	Timer	USB device EP3	PCMIN	MICIN
Ch-3	UART2	SDI	SPI1	Timer	USB device EP4	MICIN	PCMOUT

这里 nXDREQ0 和 nXDREQ1 表示两个外部源（外部设备），I2SSDO 和 I2SSDI 分别表示 IIS 传输和接收。

8.1.2 DMA 操作

DMA 使用三态 FSM（有限状态机）进行操作，其由一下三个步骤来描述：

状态 1： 作为初始状态，DMA 等待 DMA 请求。一旦请求到达，其进入状态 2。在此状态，DMA ACK 和 INT REQ 都为 0。

状态 2： 在此状态，DMA ACK 变成 1 且计数器（CURR_TC）从 DCONN[19:0]装载。注意 DMA ACK 直到后面被清除一直保持 1。

状态 3： 在此状态，进行 DMA 原子操作的子状态机被初始化。此子状态机从源地址读取数据并写入目的地址。此操作应该考虑数据大小和传输大小（single or burst）。在全服务模式 Whole service mode 中，此操作一直被重复直到计数器（CURR_TC）变成 1；而在单服务模式 Single service mode 中，此操作仅被执行一次。主有限机对计数器（CURR_TC）减少计数。此外，当计数器（CURR_TC）变成 0 且外部中断设置 DCON[29]寄存器被置 1，主状态机发出中断请求信号（INT REQ）。另外，遇到以下情况，其清除 DMA ACK。

- (1) 在全服务模式下计数器（CURR_TC）变成 0
- (2) 在单个服务模式下计数器原子操作完成

注意：在单个服务模式下，主状态机的三个状态执行然后停止，再等待其他的 DMA REQ。如果 DMA REQ 到来，将重复这样的三个状态。因此，每个原子传输过程中 DMA ACK 总是先有效再无效。相反，在全服务模式中，主状态机一直在状态 3 等待，直到计数器（CURR_TC）变成 0。所以 DMA ACK 在这个传输过程中有效，然后当计数器（CURR_TC）为 0 时无效。

但是不管服务模式如何，一旦计数器（CURR_TC）变成 0，中断请求信号 INT REQ 发出。

8.1.3 外部 DMA 请求/应答协议

有三种类型的外部 DMA 请求/应答协议（单服务请求，单服务握手和全服务握手模式）。每种模式都定义了像 DMA 请求和应答信号和这些协议如何相关。

8.1.4 基本 DMA 时序

DMA 服务意味着在 DMA 操作中执行一对读写周期，其形成了一个 DMA 操作。如图 8-1 所示 s3c2440A 的 DMA 操作的基本时序。

- 在所有模式中，XnXDREQ 和 XnXDACK 的建立时间和延迟时间是一致的。
- 如果 XnXDREQ 满足其建立时间，其将被同步两个时钟周期且 XnXDACK 被引发。
- 在 XnXDACK 被引发后，DMA 请求总线，如果其得到总线控制权，他将执行操作。当 DMA 操作完成后，XnXDACK 被设无效。

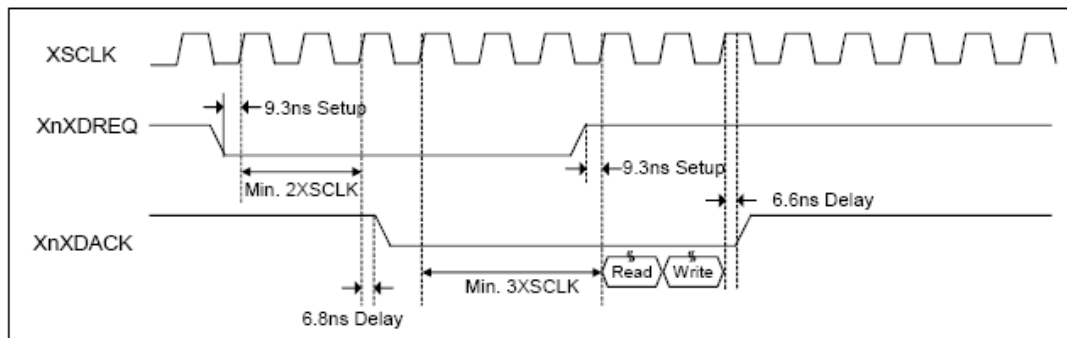


Figure 8-1. Basic DMA Timing Diagram

8.1.5 请求/握手模式对比

请求和握手模式与 XnXDREQ 和 XnXDACK 之间的协议有关。如图 8-2 所示两种模式的不同。

在一个传输的末尾（Single/Burst transfer），DMA 检测两次同步的 XnXDREQ 的状态。

请求模式

- 如果 XnXDREQ 保持有效，下个传输马上开始，否则它会一直等到 XnXDREQ 有效。

握手模式

- 如果 XnXDREQ 无效，DMA 在两个周期内将 XnXDACK 设无效。否则他会一直等 XnXDREQ 无效。

警告：只有在 XnXDACK 无效（高电平）以后，XnXDREQ 才能为有效（低电平）。

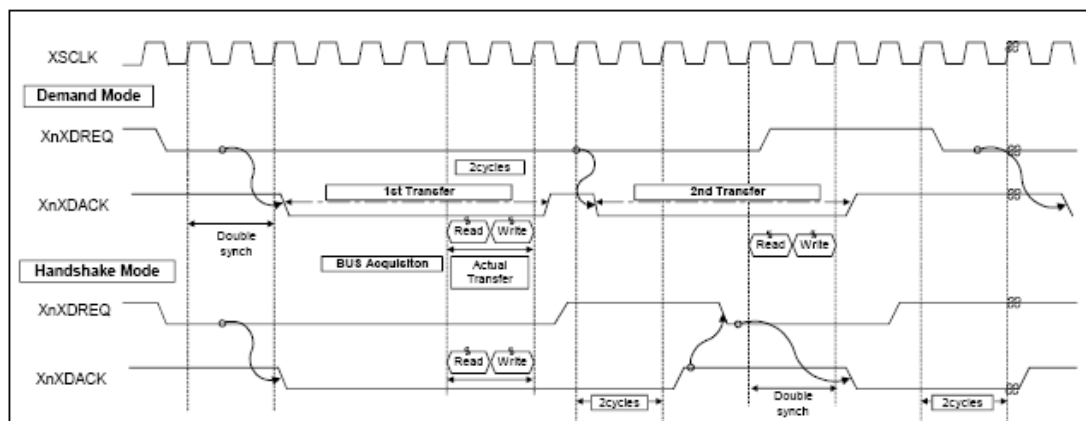


Figure 8-2. Demand/Handshake Mode Comparison

8.1.6 传输大小

-两种不同的传输大小: unit 和 burst4

-在传输大块数据时 DMA 掌握总线。这样其他的总线主设备就不能得到总线控制权。

Burst4 的传输大小

在 Burst4 传输中有 4 个连续的读写操作被分别执行。

注意: Unit 传输大小: 一个读和一个写被执行。

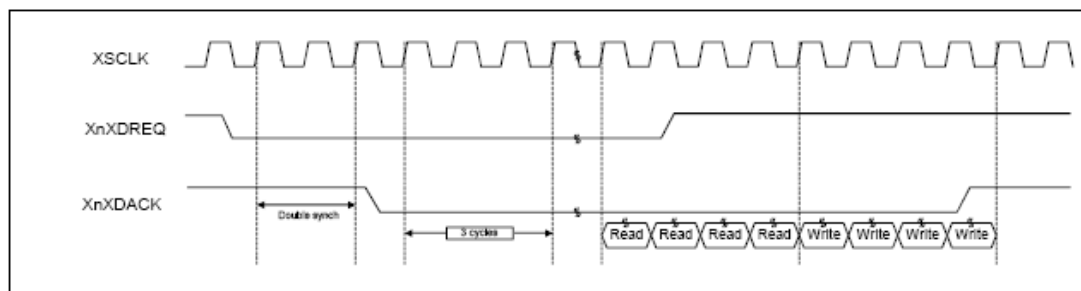


Figure 8-3. Burst 4 Transfer Size

8.1.7 举例

Unit 传输大小请求模式下的单服务

对于每个 unit 传输 (单服务模式) XnXDREQ 都需要为有效。当 XnXDREQ 为有效时 (请求模式) 操作将继续, 一对读写操作 (单传输大小) 被执行。

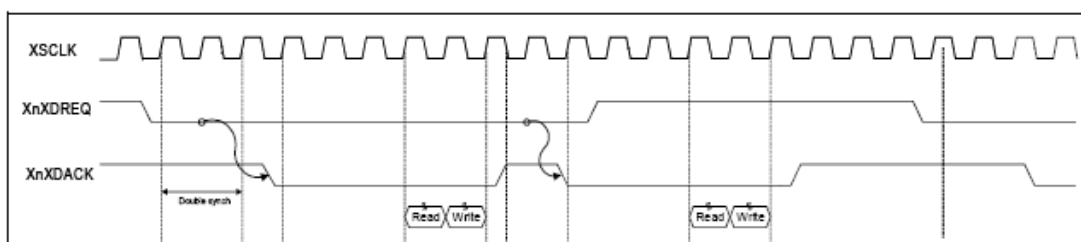


Figure 8-4. Single service in Demand Mode with Unit Transfer Size

Unit 传输大小握手模式下的单服务

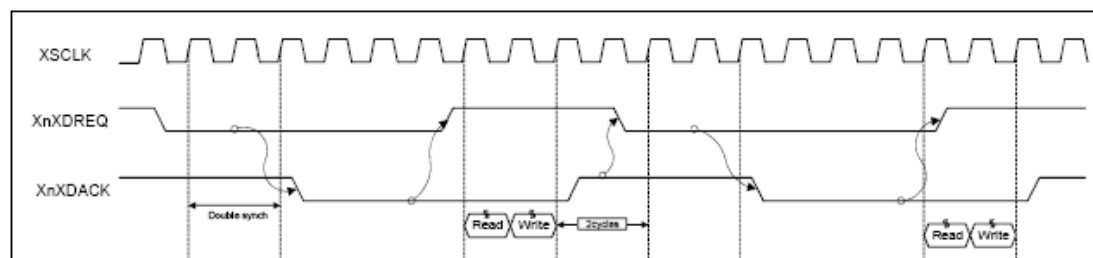


Figure 8-5. Single service in Handshake Mode with Unit Transfer Size

Unit 传输大小握手模式下的全服务

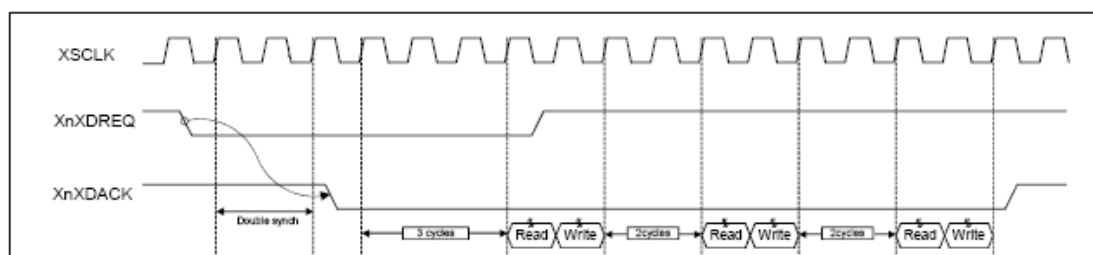


Figure 8-6. Whole service in Handshake Mode with Unit Transfer Size

8.2 DMA 特殊寄存器

每个 DMA 通道都有 9 个控制寄存器（4 个通道 DMA 控制器共计 36 个寄存器）。6 个寄存器用来控制 DMA 传输，其他三个监视 DMA 控制器的状态。这些寄存器的详细情况如下：

- (1) DMA 初始源寄存器(DISRC)
- (2) DMA初始源控制寄存器(DISRCC)
- (3) DMA初始目的寄存器(DIDST)
- (4) DMA初始目的控制寄存器(DIDSTC)
- (5) DMA控制寄存器(DCON)
- (6) DMA状态寄存器（DSTAT）
- (7) DMA当前源寄存器(DCSRC)
- (8) DMA当前目的寄存器(DCDST)
- (9) DMA屏蔽触发寄存器(DMASKTRIG)

8.2.1DMA 初始源寄存器

DMA INITIAL SOURCE REGISTER (DISRC)

寄存器	地址	读写	描述	复位值
DISRC0	0x4B000000	R/W	DMA0 初始源寄存器	0x00000000
DISRC1	0x4B000040	R/W	DMA1 初始源寄存器	0x00000000
DISRC2	0x4B000080	R/W	DMA2 初始源寄存器	0x00000000
DISRC3	0x4B0000C0	R/W	DMA3 初始源寄存器	0x00000000

DISRCn	位	描述	初始值
S_ADDR	[30:0]	用于传输的源数据基址（开始地址）。如果 CURR_SRC 为 0 且 DMA ACK 为 1，该位值将被装载到 CURR_SRC。	0x00000000

8.2.2 DMA 初始源控制寄存器

DMA INITIAL SOURCE CONTROL REGISTER (DISRCC)

寄存器	地址	读写	描述	复位值
DISRCC0	0x4B000004	R/W	DMA0 初始源控制寄存器	0x00000000
DISRCC1	0x4B000044	R/W	DMA1 初始源控制寄存器	0x00000000
DISRCC2	0x4B000084	R/W	DMA2 初始源控制寄存器	0x00000000
DISRCC3	0x4B0000C4	R/W	DMA3 初始源控制寄存器	0x00000000

DISRCCn	位	描述	初始值
LOC	[1]	位 1 被用于选择本地源。 0: 源在系统总线(AHB)上。 1: 源在外设总线(APB)上。	0
INC	[0]	位 0 被用于选择地址增加。 0: 增加 1: 固定 如果为 0, 每次传输以后, 地址增加相应的数据大小 如果为 1, 每次传输以后, 地址保持不变。(burst 模式下, 地址在burst传输中增加, 但是在传输后被初始值覆盖)	0

8.2.3 DMA 初始目的寄存器

DMA INITIAL DESTINATION REGISTER (DIDST)

寄存器	地址	读写	描述	复位值
DIDST0	0x4B000008	R/W	DMA0 初始目的控制寄存器	0x00000000
DIDST1	0x4B000048	R/W	DMA1 初始目的控制寄存器	0x00000000
DIDST2	0x4B000088	R/W	DMA2 初始目的控制寄存器	0x00000000
DIDST3	0x4B0000C8	R/W	DMA3 初始目的控制寄存器	0x00000000

DIDSTn	位	描述	初始值
D_ADDR	[30:0]	用于传输的目的数据基址 (开始地址)。如果 CURR_SRC 为 0 且 DMA ACK 为 1, 该位值将被装载到 CURR_SRC。	0x00000000

8.2.4 DMA 初始目的控制寄存器

DMA INITIAL DESTINATION CONTROL REGISTER (DIDSTC)

寄存器	地址	读写	描述	复位值
DIDSTC0	0x4B00000C	R/W	DMA0 初始目的控制寄存器	0x00000000
DIDSTC1	0x4B00004C	R/W	DMA1 初始目的控制寄存器	0x00000000
DIDSTC2	0x4B00008C	R/W	DMA2 初始目的控制寄存器	0x00000000
DIDSTC3	0x4B0000CC	R/W	DMA3 初始目的控制寄存器	0x00000000

DISRCCn	位	描述	初始值
CHK_INT	[2]	当自动重载被设置, 选择中断出现时间。 0: 当TC为 0, 中断出现 1: 自动重载被执行后, 中断出现	0
LOC	[1]	位 1 被用于选择本地目的。 0: 源在系统总线(AHB)上。 1: 源在外设总线(APB)上。	0
INC	[0]	位 0 被用于选择地址增加。 0: 增加 1: 固定 如果为 0, 每次传输以后, 地址增加相应的数据大小 如果为 1, 每次传输以后, 地址保持不变。 (burst模式下, 地址在burst传输中增加, 但是在传输后被初始值覆盖)	0

8.2.5 DMA 控制寄存器

DMA CONTROL REGISTER (DCON)

寄存器	地址	读写	描述	复位值
DCON0	0x4B000010	R/W	DMA0 控制寄存器	0x00000000
DCON1	0x4B000050	R/W	DMA1 控制寄存器	0x00000000
DCON2	0x4B000090	R/W	DMA2 控制寄存器	0x00000000
DCON3	0x4B0000D0	R/W	DMA3 控制寄存器	0x00000000

DISRCCn	位	描述	初始值
DMD_HS	[31]	在请求模式和握手模式中选 1。 0: 选择请求模式 1: 选择握手模式 在两种模式下, DMA 控制器开始传输且对于一个 DREQ 有效, 使得 DACK 有效。两种模式的差异其是否等待 DACK 无效。在握手模式下, DMA 控制器在开始一个新传输前等待无效 DREQ。如果 DREQ 无效, 其使得 DACK 无效并等待另外有效的 DREQ。相对比, 在请求模式下, DMA 控制器不会等到 DREQ 无效, 其仅将 DACK 置无效且如果 DREQ 有效则开始另外一个传输。我们推荐对于外部 DMA 请求源使用握手模式以避免不经意的开始新的传输。	0
SYNC	[30]	选择 DREQ/DACK 同步。 0: DREQ and DACK 与 PCLK (APB 时钟) 同步。 1: DREQ and DACK 与 HCLK (AHB 时钟) 同步。 因此对于连接在 AHB 总线上的设备, 该位应该置 1; 对于连接在 APB 总线上的设备, 该位应该置 0 对于连接在外部系统上的设备, 该位的设置应该取决于其外部系统同步于 AHB 系统还是 APB 系统。	0
INT	[29]	对于 CURR_TC 中断设置使能或无效 0: CURR_TC 中断无效. 用户必须查看状态寄存器中的传输计数	0

		器（例如轮询）。 1: 当所有的传输结束，中断请求生成（CURR_TC变为0）。	
TSZ	[28]	选择原子传输的传输大小（例如在释放总线之前，一旦DMA拥有总线控制权，传输被执行）。 0: 执行单元传输 1: 执行四数据长度的突发传输	
SERVMODE	[27]	在单服务模式 and 全服务模式中选择服务模式 0: 单服务模式被选定，在此模式下每个原子传输（单数据或4数据长度的突发传输）后DMA停止且等待其他的DMA请求。 1: 全服务模式被选定，在此模式下，DMA请求引起原子传输一直重复，直到传输计数器为0。此模式下不需要附加的请求。 注意：在全模式下,在每个原子传输后DMA释放总线，又试图重新得到总线以避免其他总线主设备得到总线控制。	0
HWSRCSEL	[26:24]	各DMA通道请求源选择： DCON0: 000:nXDREQ0 001:UART0 010:SDI 011:Timer 100:USB device EP1 DCON1: 000:nXDREQ1 001:UART1 010:I2SSDI 011:SPI 100:USB device EP2 DCON2: 000:I2SSDO 001:I2SSDI 010:SDI 011:Timer 100:USB device EP3 DCON3: 000:UART2 001:SDI 010:SPI 011:Timer 100:USB device EP4 DCON0: 101:I2SSDO 110:PCMIN DCON1: 101:PCMOUT 110:SDI DCON2: 101:PCMIN 110:MICIN DCON3: 101:MICIN 110:PCMOUT 这些位控制一个四选一的多路器来为每个DMA选择请求源。如果硬件请求模式通过DCONn[23]被选定，这些位才有意义。	00
SWHW_SEL	[23]	在软件（软件请求模式）和硬件（硬件请求模式）间选择DMA源 0: 选择软件请求模式且DMA通过DMASKTRIG 控制寄存器的SW_TRIG位被触发。 1: 由位[26:24]选定的DMA源触发DMA操作	0
RELOAD	[22]	设定重装载开关选项 0: 当传输计数器的当前值为0，自动重载被执行(例如所有请求的传输都被执行)。 1: 当传输计数器的当前值为0，DMA通道（DMA REQ）被关闭。DMA通道开关位 (DMASKTRIGn[1]) 被清零(DREQ off) 以防止不经意的开始一个新的DMA操作。	0
DSZ	[21:20]	传输数据大小单位 00 = Byte 01 = Half word 10 = Word 11 = reserved	00
TC	[19:0]	初始化传输计数器 注意：传输的实际字节数由以下公式计算：DSZ x TSZ x TC。 DSZ, TSZ (1 或 4)和TC分别代表数据大小DCONn[21:20]、	00000

		传输大小DCONn[28]和初始传输计数器。仅当CURR_TC为 0 且 DMA ACK 为 1, 该值将被重载入CURR_TC。	
--	--	---	--

8.2.6 DMA 状态寄存器

DMA STATUS REGISTER (DSTAT)

寄存器	地址	读写	描述	复位值
DSTAT0	0x4B000014	R/W	DMA0 状态寄存器	0x00000000
DSTAT1	0x4B000054	R/W	DMA1 状态寄存器	0x00000000
DSTAT2	0x4B000094	R/W	DMA2 状态寄存器	0x00000000
DSTAT3	0x4B0000D4	R/W	DMA3 状态寄存器	0x00000000

DSTATn	位	描述	初始值
STAT	[21:20]	DMA控制器的状态 00: 指出DMA控制器准备好接收其他DMA请求 01: 指出DMA控制器忙于传输	00b
CURR_TC	[19:0]	传输个数的当前值 注: 传输计数值被DCONn[19:0]初始化, 在每一次原子传输结束后该值减一。	00000b

8.2.7 DMA 当前源寄存器

DMA CURRENT SOURCE REGISTER (DCSRC)

寄存器	地址	读写	描述	复位值
DCSRC0	0x4B000018	R/W	DMA0 当前源寄存器	0x00000000
DCSRC1	0x4B000058	R/W	DMA1 当前源寄存器	0x00000000
DCSRC2	0x4B000098	R/W	DMA2 当前源寄存器	0x00000000
DCSRC3	0x4B0000D8	R/W	DMA3 当前源寄存器	0x00000000

DCSRCn	位	描述	初始值
CURR_SRC	[30:0]	DMA _n 当前源地址	0x00000000

8.2.8 DMA 当前目的寄存器

CURRENT DESTINATION REGISTER (DCDST)

寄存器	地址	读写	描述	复位值
DCDST0	0x4B00001C	R/W	DMA0 当前目的寄存器	0x00000000
DCDST1	0x4B00005C	R/W	DMA1 当前目的寄存器	0x00000000
DCDST2	0x4B00009C	R/W	DMA2 当前目的寄存器	0x00000000
DCDST3	0x4B0000DC	R/W	DMA3 当前目的寄存器	0x00000000

DCDSTn	位	描述	初始值
CURR_DST	[30:0]	DMA _n 当前目的地址	0x00000000

8.2.9 DMA 屏蔽触发寄存器

DMA MASK TRIGGER REGISTER (DMASKTRIG)

寄存器	地址	读写	描述	复位值
DMASKTRIG0	0x4B000020	R/W	DMA0 屏蔽触发寄存器	0x00000000
DMASKTRIG1	0x4B000060	R/W	DMA1 屏蔽触发寄存器	0x00000000
DMASKTRIG2	0x4B0000A0	R/W	DMA2 屏蔽触发寄存器	0x00000000
DMASKTRIG3	0x4B0000E0	R/W	DMA3 屏蔽触发寄存器	0x00000000

DMASKTRIGn	位	描述	初始值
STOP	[2]	<p>停止DMA 操作。</p> <p>1: 在当前原子操作结束后立即停止DMA操作。如果没有当前原子操作，DMA会立即停止。</p> <p>CURR_TC, CURR_SRC, and CURR_DST 将为 0。</p> <p>注：由于可能的当前原子传输，停止操作可能会花费几个周期。一旦DMA通道开关位（DMAKSTRIG[1]）被设关闭，才能检测出其操作结束（例如，真实停止时间）。此位又名“真正停止”</p>	0
ON_OFF	[1]	<p>DMA通道开关位</p> <p>0: DMA 通道被关闭（对此通道的DMA请求被忽略）</p> <p>1: DMA 通道被开启，DMA请求被处理</p> <p>如果我们设置DCONn[22]位到“非自动装载”且/或停止位DMASKTRIGn设为停止，此位将自动关闭。</p> <p>注意当DCONn[22]位为“非自动装载”，CURR_TC为零则此位清零。如果停止位是 1，一旦当前原子传输完成，此位立即清零。</p> <p>注：此位在DMA操作期间不应该被手工修改（例如通过使用DCON[22]或停止位来修改此位）</p>	0
SW_TRIG	[0]	<p>在软件模式下触发DMA通道。</p> <p>1: 请求对于该控制器的DMA操作。</p> <p>注意：在软件请求模式被选定（DCONn[23]）且通道ON_OFF位被置 1（通道开启）后，此触发器有效。</p> <p>当DMA操作开始后，该位自动清零。</p>	0

第十章 PWM 及定时器

10.1 概述

S3c2440A 有 5 个 16 位的定时器。定时器 0、1、2、3 有脉宽调制功能（PWM）。定时器 4 有一个没有输出引脚的内部定时器。定时器 0 有一个用于大电流设备的死区生成器。

定时器 0 和 1 共享一个 8 位的预分频器（预定标器），定时器 2、3、4 共享另一个 8 位预分频器（预定标器）。每个定时器有一个时钟分频器，其可以生成 5 种不同的分频信号（1/2, 1/4, 1/8, 1/16 和 TCLK）。每个定时器模块从时钟分频器接收其自己的时钟信号，其分频器从相应的 8 位预分频器（预定标器）接收时钟。8 位的预标器是可编程的且根据装载的值来分频 PCLK，其值存储在 TCFG0 和 TCFG1 寄存器中。

当定时器使能，定时器计数缓存寄存器（TCNTBn）得到一个被装载到递减计数器中的初始值。定时器比较缓存寄存器（TCMPBn）有一个被装载到比较寄存器中用来和递减计数器的值作比较的初始值。TCNTBn 和 TCMPBn 双缓存特点使得当频率和负荷比发生改变时，定时器生成一个稳定的输出。

每个定时器有一个自己的由定时器时钟驱动的 16 位递减计数器。当递减计数器为零时，定时器中断请求生成通知 CPU 定时器操作已经完成。当定时器计数器达到 0，相应的 TCNTBn 的值也知道装载到递减计数器中以继续下一个操作。但是如果定时器停止了，例如在定时器运行模式下通过对 TCONn 的定时器使能位清零，则 TCNTBn 的值不会装载到计数器中。

TCMPBn 的值用于脉宽调制。当递减计数器的值和定时器控制逻辑中的比较寄存器的值匹配时，定时器控制逻辑改变输出电平。因此，比较寄存器决定了 PWM 输出的开启时间。

10.2 特性

- (1) 5 个 16 位定时器
- (2) 两个 8 位预分频器（预定标器）和 2 个 4 位分频器
- (3) 输出波形的可编程任务控制
- (4) 自动重载模式或单脉冲模式
- (5) 死区生成器

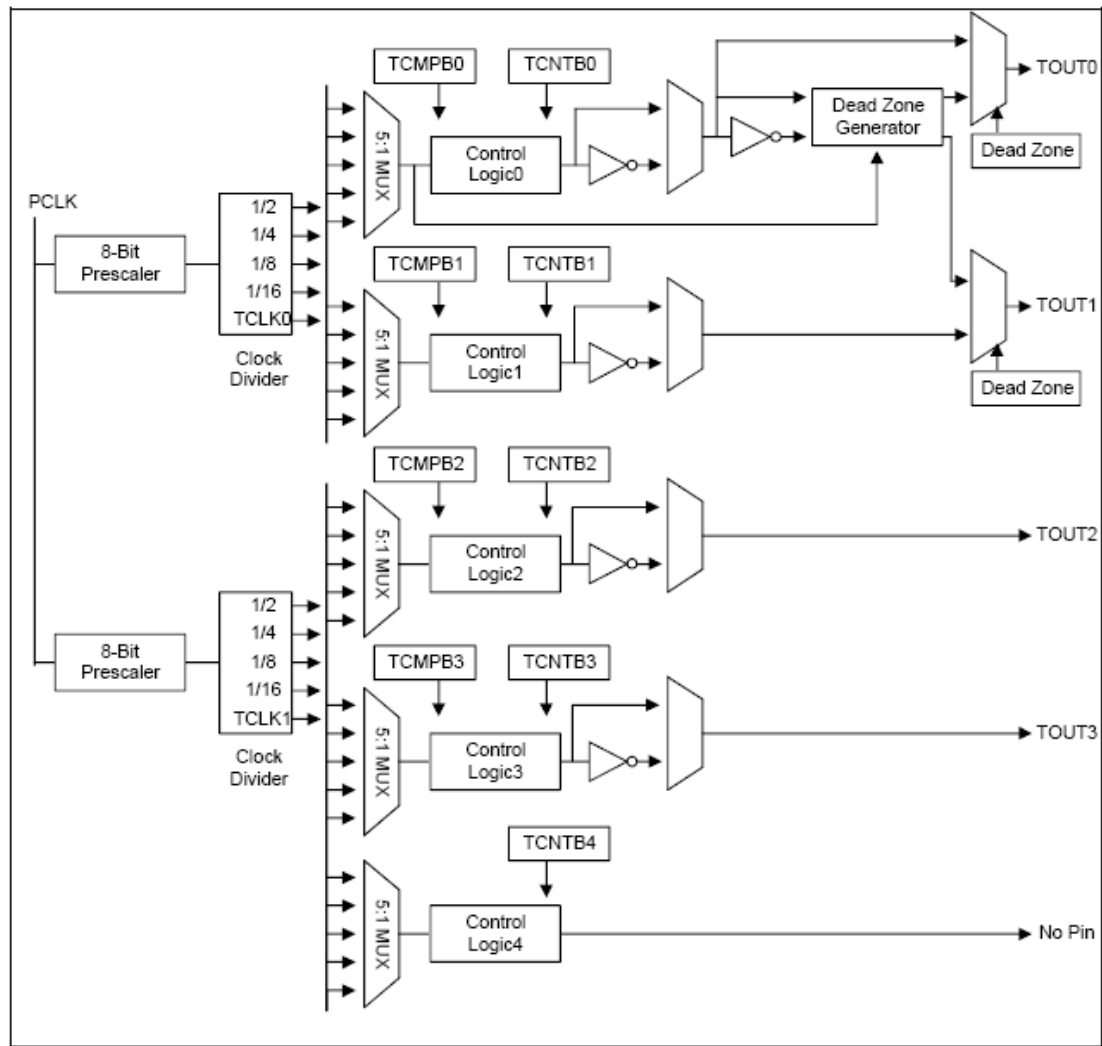


Figure 10-1. 16-bit PWM Timer Block Diagram

10.2.1 PWM 定时器操作

10.2.1.1 预分频器（预定标器）和分频器

一个 8 位的预分频器（预定标器）和一个 4 位的分频器得到以下输出频率：

4 位分频器设置	最小分辨率 (prescaler = 0)	最大分辨率 (prescaler = 255)	最大间隔 (TCNTBn = 65535)
1/2 (PCLK = 50MHz)	0.0400 us (25.0000 MHz)	10.2400 us (97.6562 MHz)	0.6710 sec
1/4 (PCLK = 50MHz)	0.0800 us (12.5000 MHz)	20.4800 us (48.8281 KHz)	1.3421 sec
1/8 (PCLK = 50MHz)	0.1600 us (6.2500 MHz)	40.9601 us (24.4140 KHz)	2.6843 sec
1/16 (PCLK = 50 MHz)	0.3200 us (3.1250 MHz)	81.9188 us (12.2070 KHz)	5.3686 sec

10.2.1.2 基本定时器操作

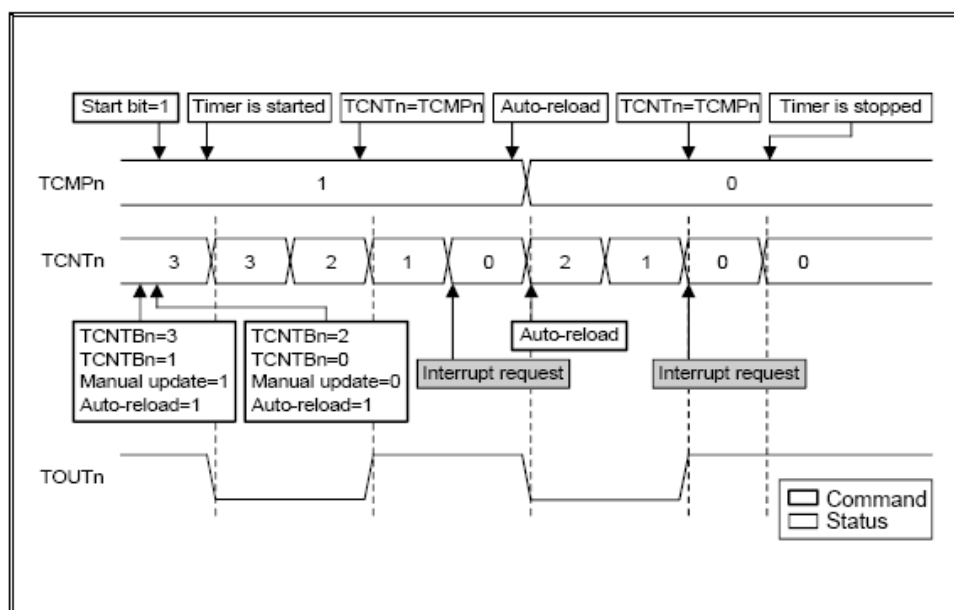


Figure 10-2. Timer Operations

定时器（除定时器通道 5）有 TCNTBn、TCNTn、TCMPBn 和 TCMPn。（TCNTn 和 TCMPn 是内部寄存器的名字。TCNTn 寄存器可以从 TCNTOn 寄存器读取）当定时器为 0，TCNTBn 和 TCMPBn 可以被装载到 TCNTn 和 TCMPn 中。当 TCNTn 为 0，如果中断使能则一个中断请求将出现。

10.2.1.3 自动重载和双缓冲

S3c2440A 的 PWM 定时器有一个双缓冲功能，在不停止当前定时器操作的情况下对于下个定时器操作使能重载值改变。所以，尽管一个新定时器值被设置，当前定时器操作还能成功完成。

定时器值可以写到定时器计数缓存寄存器 TCNTBn，定时器的当前计数器值可以从定时器计数观察寄存器 TCNTOn 中读取。如果 TCNTBn 被读取，读取值不指示计数器的当前状态而是下个定时器期间的重载值。

当 TCNTn 为 0，自动重载操作拷贝 TCNTBn 到 TCNTn。仅当 TCNTn 到达 0 且自动重载使能，写入 TCNTBn 的值被装载到 TCNTn 中。如果 TCNTn 变成 0 且自动重载位为 0，TCNTn 不再进一步操作。

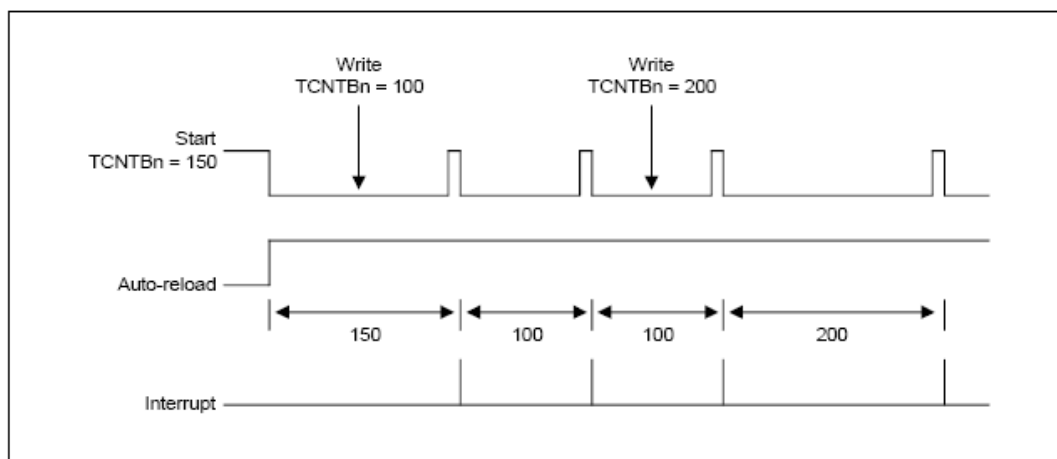


Figure 10-3. Example of Double Buffering Function

10.2.1.4 使用手动更新位和反相器位的定时器初始化

当递减计数器为 0 将出现定时器自动重载操作。因此 TNCTn 的初始值必须由用户预选定义。在这种情况下，初始值必须由手动更新位来装载。以下步骤描述了如何开始一个定时器：

- (1) 写初始值到 TCNTBn 和 TCMPBn
- (2) 设置相应定时器的手动更新位。推荐配置反相器开关为（不论是否使用反相器）。
- (3) 设置相应定时器的开始位开启定时器（且清除手动更新位）

如果定时器被强行停止，TCNTn 保留计数器值且不从 TCNTBn 重载。如果必须设置新值，执行手动更新。

注：不论 TOUT 反相器开关位是否改变，如果定时器在运行 TOUTn 逻辑值也不会改变。因此其描述了反相器开关位由手动更新位来配置。

10.2.2 定时器操作

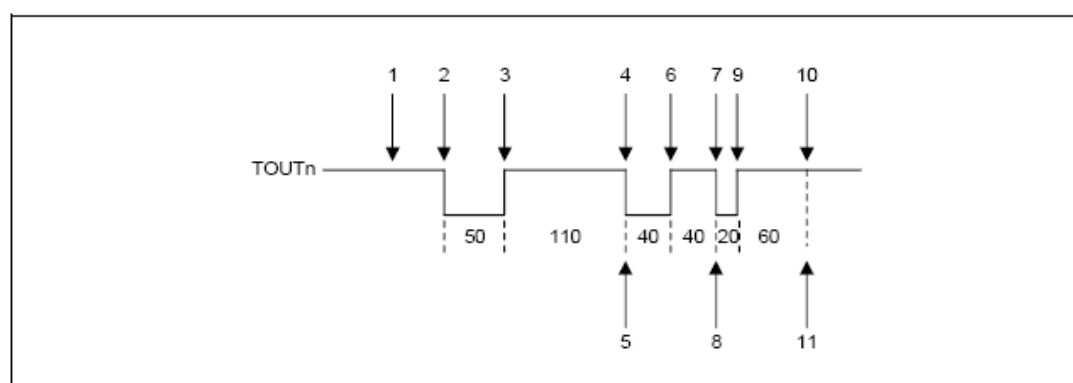


Figure 10-4. Example of a Timer Operation

如图 10-4 所示以下过程的结果：

- (1) 使能自动重载功能。设置 TCNTBn 为 160 (50+110)，TCMPBn 为 110。设置手动更新位且配置反相器位（开/关）。手动更新位将 TCNTn 和 TCMPn 的值分别置给 TCNTBn 和 TCMPBn。然后分别设置 TCNTBn 和 TCMPBn 的值为 80 (40+40) 和 40 以决定下一个重载值。
- (2) 设置开始位，手动更新位为 0，反相器关闭且自动重载开启。在等待时间后定时器开始倒数
- (3) 当 TCNTn 的值和 TCMPn 的值相同，TOUTn 的逻辑电平从低变为高。
- (4) 当 TCNTn 为 0，中断请求生成且 TCNTBn 的值装载到一个临时寄存器中。在下一个定时器周期，TCNTn 会用该临时寄存器的值重载。

- (5) 在中断服务程序中, TCNTBn 和 TCMPBn 为下个周期被分别设置为 80 (60+20) 和 60
- (6) 当 TCNTn 的值和 TCMPn 的值相同, TOUTn 的逻辑电平从低变成高。
- (7) 当 TCNTn 为 0, TCNTn 用 TCNTBn 的值自动重载, 并触发中断请求。
- (8) 在中断服务程序中, 自动重载和自动请求被设无效以停止定时器
- (9) 当 TCNTn 的值和 TCMPn 的值相同, TOUTn 的逻辑电平从低变成高。
- (10) 就算 TCNTn 为 0, 因为自动重载被设为无效, TCNTn 不再被重载且定时器停止。
- (11) 没有其他的自动请求生成

10.2.3 脉宽调制 (PWM)

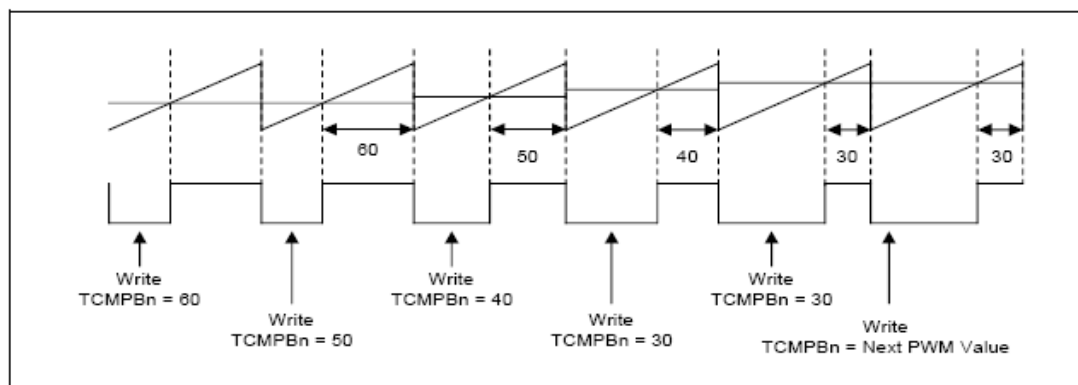


Figure 10-5. Example of PWM

通过使用 TCMPBn 来执行 PWM 功能。PWM 的频率由 TCNTBn 来决定。如图 10-5 所示 TCMPBn 的值来决定 PWM 的值。

要得到一个更高的 PWM 的值, 则要减少 TCMPBn 的值。要得到一个更低的 PWM 的值, 则要增加 TCMPBn 的值。如果使用了反相器, 则增加和减少可以相反。

双缓冲功能允许对于下个 PWM 周期在当前 PWM 周期任意时间点由 ISR 或其他程序改写 TCMPBn

10.2.4 输出电平控制

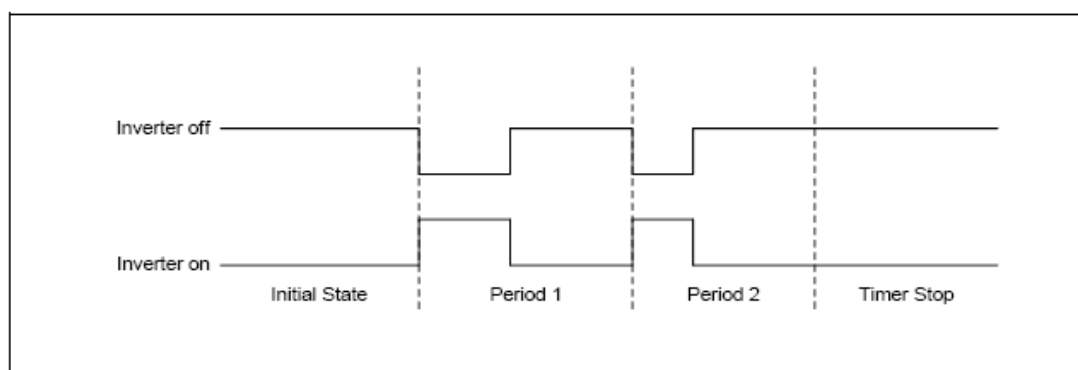


Figure 10-6. Inverter On/Off

以下过程描述如何保持 TOUT 高电平或低电平 (假定反相器关闭)

- (1) 关闭中断重载位。然后 TOUTn 变成高电平, 在 TCNTn 为 0 后定时器停止
- (2) 对定时器的开始停止位清零停止定时器。如果 TCNTn ≤ TCMPn, 输出为高电平。如果 TCNTn > TCMPn, 输出为低电平。
- (3) TOUTn 可以由 TCON 中的反相器开启关闭位来翻转。反相器删除了用于调节输出电平的附加电路。

10.2.5 死区生成器

死区用于电源设备的 PWM 控制。该功能是使能在关闭一个开关设备和开启另一个开关设备之间插入一个时间间隙。该时间间隙禁止两个开关设备同时开启即使是在极短的时间内。

TOUT0 是 PWM 输出。nTOUT0 是 TOUT0 的倒置。如果死区使能，TOUT0 和 TOUT1 的输出波形分别是 TOUT0_DZ 和 nTOUT0_DZ。nTOUT0_DZ 被发生到 TOUT1 引脚。

在死区间隙，TOUT0_DZ 和 nTOUT0_DZ 不能同时开启。

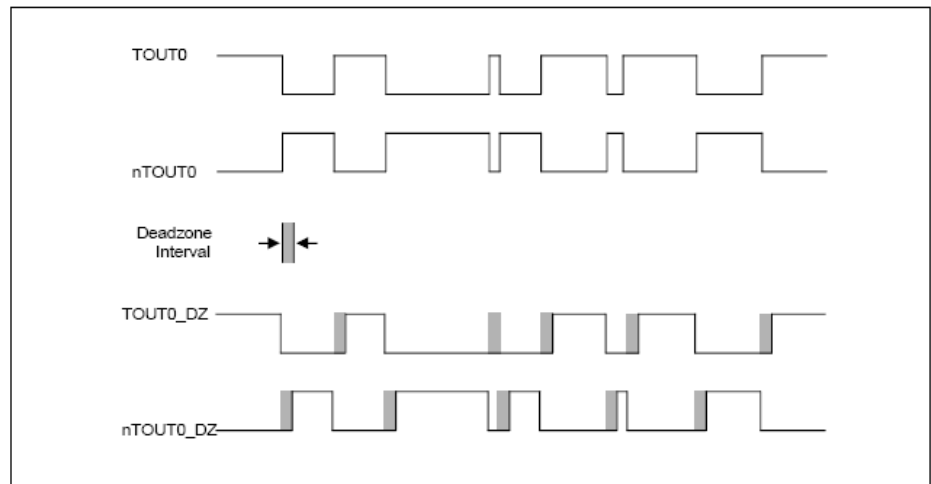


Figure 10-7. The Wave Form When a Dead Zone Feature is Enabled

10.2.6 DMA 请求模式

PWM 定时器可以在每个特定时间生成 DMA 请求。定时器保持 DMA 请求信号（nDMA_REQ）为低电平直到定时器收到一个 ACK 信号。当定时器收到一个 ACK 信号，其使得请求信号无效。生成 DMA 请求的定时器可以通过 DMA 模式位（在 TCFG 寄存器中）来决定。如果一个定时器配置为 DMA 请求模式，定时器不会生成中断请求，其他定时器还可以正常的生成中断。

DMA 模式配置和 DMA/中断操作

DMA 模式	DMA 请求	Timer0 INT	Timer1 INT	Timer2 INT	Timer3 INT	Timer4 INT
0000	Not select	ON	ON	ON	ON	ON
0001	Timer0	OFF	ON	ON	ON	ON
0010	Timer1	ON	OFF	ON	ON	ON
0011	Timer2	ON	ON	OFF	ON	ON
0100	Timer3	ON	ON	ON	OFF	ON
0101	Timer4	ON	ON	ON	ON	OFF
0110	Not select	ON	ON	ON	ON	ON

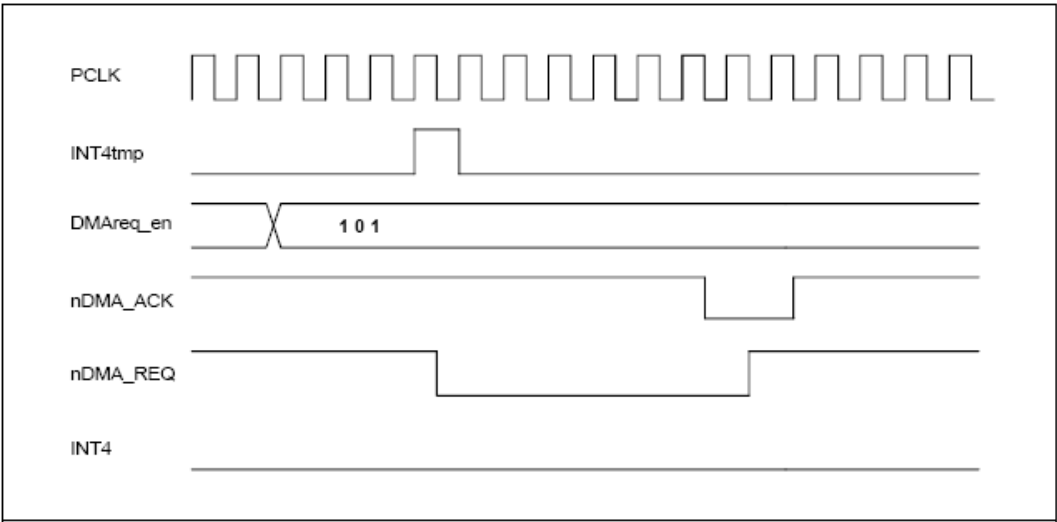


Figure 10-8. Timer4 DMA Mode Operation

10.3 脉宽调制定时器特殊寄存器

- (1) 定时器配置寄存器(TCFGn)
- (2) 定时器控制寄存器 (TCON)
- (3) 定时器 n 计数缓存寄存器 (TCNTBn)，定时器 n 比较缓存寄存器 (TCMPBn)
- (4) 定时器 n 计数观察计数器 (TCNTOn)

10.3.1 定时器配置寄存器 0

TIMER CONFIGURATION REGISTER0 (TCFG0)

定时器输出时钟频率 = PCLK / {prescaler value+1} / {divider value}

{prescaler value} = 0~255

{divider value} = 2, 4, 8, 16

寄存器	地址	读写	描述	复位值
TCFG0	0x51000000	R/W	配置两个 8 位预分频器（预定标器）	0x00000000

TCFG0	位	描述	初始值
保留	[31:24]		0x00
Dead zone length	[23:16]	此 8 位决定死区长度。 死区长度的单位时间等于定时器 0 的单位时间。	0x00
Prescaler 1	[15:8]	此 8 位决定定时器 2, 3, 4 的预标器值	0x00
Prescaler 0	[7:0]	此 8 位决定定时器 0, 1 的预标器值	0x00

10.3.2 定时器配置寄存器 1

TIMER CONFIGURATION REGISTER1 (TCFG1)

寄存器	地址	读写	描述	复位值
TCFG1	0x51000004	R/W	5 路多路器&DMA模式选择寄存器	0x00000000

TCFG1	位	描述	初始值
保留	[31:24]		00000000
DMA mode	[23:20]	Select DMA request channel 0000 = No select (all interrupt), 0001 = Timer0 0010 = Timer1, 0011 = Timer2, 0100 = Timer3 , 0101 = Timer4, 0110 = Reserved	0000
MUX4	[19:16]	Select MUX input for PWM Timer4. 0000 = 1/2 0001 = 1/4 0010 = 1/8 0011 = 1/16 01xx = External TCLK1	0000
MUX3	[15:12]	Select MUX input for PWM Timer3. 0000 = 1/2 0001 = 1/4 0010 = 1/8 0011 = 1/16 01xx = External TCLK1	0000
MUX2	[11:8]	Select MUX input for PWM Timer2. 0000 = 1/2 0001 = 1/4 0010 = 1/8 0011 = 1/16 01xx = External TCLK1	0000
MUX1	[7:4]	Select MUX input for PWM Timer1. 0000 = 1/2 0001 = 1/4 0010 = 1/8 0011 = 1/16 01xx = External TCLK0	0000
MUX0	[3:0]	Select MUX input for PWM Timer0. 0000 = 1/2 0001 = 1/4 0010 = 1/8 0011 = 1/16 01xx = External TCLK0	0000

10.3.3 定时器控制寄存器

TIMER CONTROL REGISTER (TCON)

寄存器	地址	读写	描述	复位值
TCON	0x51000008	R/W	定时器控制寄存器	0x00000000

TCON	位	描述	初始值
Timer 4 auto reload on/off	[22]	Determine auto reload on/off for Timer 4. 0 = One-shot 1 = Interval mode (auto reload)	0
Timer 4 manual update (note)	[21]	Determine the manual update for Timer 4. 0 = No operation 1 = Update TCNTB4	0
Timer 4 start/stop	[20]	Determine start/stop for Timer 4. 0 = Stop 1 = Start for Timer 4	0
Timer 3 auto reload on/off	[19]	Determine auto reload on/off for Timer 3. 0 = One-shot 1 = Interval mode (auto reload)	0
Timer 3 output inverter on/off	[18]	Determine output inverter on/off for Timer 3. 0 = Inverter off 1 = Inverter on for TOUT3	0
Timer 3 manual update (note)	[17]	Determine manual update for Timer 3. 0 = No operation 1 = Update TCNTB3 & TCMPB3	0
Timer 3 start/stop	[16]	Determine start/stop for Timer 3. 0 = Stop 1 = Start for Timer 3	0
Timer 2 auto reload on/off	[15]	Determine auto reload on/off for Timer 2. 0 = One-shot 1 = Interval mode (auto reload)	0
Timer 2 output inverter on/off	[14]	Determine output inverter on/off for Timer 2. 0 = Inverter off 1 = Inverter on for TOUT2	0
Timer 2 manual update (note)	[13]	Determine the manual update for Timer 2. 0 = No operation 1 = Update TCNTB2 & TCMPB2	0
Timer 2 start/stop	[12]	Determine start/stop for Timer 2. 0 = Stop 1 = Start for Timer 2	0
Timer 1 auto reload on/off	[11]	Determine the auto reload on/off for Timer1. 0 = One-shot 1 = Interval mode (auto reload)	0
Timer 1 output inverter on/off	[10]	Determine the output inverter on/off for timer1. 0 = Inverter off 1 = Inverter on for TOUT1	0
Timer 1 manual update (note)	[9]	Determine the manual update for Timer 1. 0 = No operation 1 = Update TCNTB1 & TCMPB1	0
Timer 1 start/stop	[8]	Determine start/stop for Timer 1. 0 = Stop 1 = Start for Timer 1	0
Reserved	[7:5]	Reserved	
Dead zone enable	[4]	Determine the dead zone operation. 0 = Disable 1 = Enable	0
Timer 0 auto reload on/off	[3]	Determine auto reload on/off for Timer 0. 0 = One-shot 1 = Interval mode(auto reload)	0
Timer 0 output inverter on/off	[2]	Determine the output inverter on/off for Timer 0. 0 = Inverter off 1 = Inverter on for TOUT0	0
Timer 0 manual update (note)	[1]	Determine the manual update for Timer 0. 0 = No operation 1 = Update TCNTB0 &	0

		TCMPB0	
Timer 0 start/stop	[0]	Determine start/stop for Timer 0. 0 = Stop 1 = Start for Timer 0	0

10.3.4 定时器 0 计数缓存寄存器&比较缓存寄存器

TIMER0 COUNT BUFFER REGISTER & COMPARE BUFFER REGISTER (TCNTB0/TCMPB0)

寄存器	地址	读写	描述	复位值
TCNTB0	0x5100000C	R/W	定时器 0 计数缓存寄存器	0x00000000
TCMPB0	0x51000010	R/W	定时器 0 比较缓存寄存器	0x00000000

TCMPB0	位	描述	初始值
Timer 0 compare buffer register	[15:0]	Set compare buffer value for Timer0	0x00000000

TCNTB0	位	描述	初始值
Timer 0 count buffer register	[15:0]	Set count buffer value for Timer 0	0x00000000

10.3.5 定时器 0 计数观察寄存器

TIMER0 COUNT OBSERVATION REGISTER (TCNTO0)

寄存器	地址	读写	描述	复位值
TCNTO0	0x51000014	R	定时器 0 计数观察寄存器	0x00000000

TCNTO0	位	描述	初始值
Timer 0 observation register	[15:0]	Set count observation value for Timer 0	0x00000000

10.3.6 定时器 1 计数缓存寄存器&比较缓存寄存器

TIMER1 COUNT BUFFER REGISTER & COMPARE BUFFER REGISTER (TCNTB1/TCMPB1)

寄存器	地址	读写	描述	复位值
TCNTB1	0x51000018	R/W	定时器 1 计数缓存寄存器	0x00000000
TCMPB1	0x5100001C	R/W	定时器 1 比较缓存寄存器	0x00000000

TCMPB1	位	描述	初始值
Timer1 compare buffer register	[15:0]	Set compare buffer value for Timer1	0x00000000

TCNTB1	位	描述	初始值
Timer 1 count buffer register	[15:0]	Set count buffer value for Timer 1	0x00000000

10.3.7 定时器 1 计数观察寄存器

TIMER1 COUNT OBSERVATION REGISTER (TCNTO1)

寄存器	地址	读写	描述	复位值
TCNTO1	0x51000020	R	定时器 1 计数观察寄存器	0x00000000

TCNTO1	位	描述	初始值
Timer 1 observation register	[15:0]	Set count observation value for Timer1	0x00000000

10.3.8 定时器 2 计数缓存寄存器&比较缓存寄存器

TIMER2 COUNT BUFFER REGISTER & COMPARE BUFFER REGISTER (TCNTB2/TCMPB2)

寄存器	地址	读写	描述	复位值
TCNTB2	0x51000024	R/W	定时器 2 计数缓存寄存器	0x00000000
TCMPB2	0x51000028	R/W	定时器 2 比较缓存寄存器	0x00000000

TCMPB2	位	描述	初始值
Timer 2 compare buffer register	[15:0]	Set compare buffer value for Timer2	0x00000000

TCNTB2	位	描述	初始值
Timer 2 count buffer register	[15:0]	Set count buffer value for Timer 2	0x00000000

10.3.9 定时器 2 计数观察寄存器

TIMER2 COUNT OBSERVATION REGISTER (TCNTO2)

寄存器	地址	读写	描述	复位值
TCNTO2	0x5100002C	R	定时器 2 计数观察寄存器	0x00000000

TCNTO2	位	描述	初始值
Timer 2 observation register	[15:0]	Set count observation value for Timer2	0x00000000

10.3.10 定时器 3 计数缓存寄存器&比较缓存寄存器

TIMER3 COUNT BUFFER REGISTER & COMPARE BUFFER REGISTER (TCNTB3/TCMPB3)

寄存器	地址	读写	描述	复位值
TCNTB3	0x51000030	R/W	定时器 3 计数缓存寄存器	0x00000000
TCMPB3	0x51000034	R/W	定时器 3 比较缓存寄存器	0x00000000

TCMPB3	位	描述	初始值
Timer 3 compare buffer register	[15:0]	Set compare buffer value for Timer3	0x00000000

TCNTB3	位	描述	初始值
Timer 3 count buffer register	[15:0]	Set count buffer value for Timer 3	0x00000000

10.3.11 定时器 3 计数观察寄存器

TIMER3 COUNT OBSERVATION REGISTER (TCNTO3)

寄存器	地址	读写	描述	复位值
TCNTO3	0x51000038	R	定时器 3 计数观察寄存器	0x00000000

TCNTO3	位	描述	初始值
Timer 3 observation register	[15:0]	Set count observation value for Timer3	0x00000000

10.3.12 定时器 4 计数缓存寄存器&比较缓存寄存器

TIMER4 COUNT BUFFER REGISTER & COMPARE BUFFER REGISTER (TCNTB4/TCMPB4)

寄存器	地址	读写	描述	复位值
TCNTB4	0x5100003C	R/W	定时器 2 计数缓存寄存器	0x00000000

TCNTB4	位	描述	初始值
Timer 4 count buffer register	[15:0]	Set count buffer value for Timer 4	0x00000000

10.3.13 定时器 4 计数观察寄存器

TIMER4 COUNT OBSERVATION REGISTER (TCNTO4)

寄存器	地址	读写	描述	复位值
TCNTO4	0x51000040	R	定时器 4 计数观察寄存器	0x00000000

TCNTO4	位	描述	初始值
Timer 4 observation register	[15:0]	Set count observation value for Timer 4	0x00000000

第十一章 UART

10.1 概述

s3c2440A 通用异步接收器和发送器 (UART) 提供了三个独立的异步串行 I/O (SIO) 端口, 每个端口都可以在中断模式或 DMA 模式下操作。换言之, UART 可以生成一个中断或 DMA 请求用于 CPU 和 UART 之间的数据传输。UART 使用系统时钟可以支持最高 115.2K bps 的波特率。如果一个外部设备提供 UEXTCLK 给 UART, UART 可以在更高的速度下工作。每个 UART 通道对于接收器和发送器包括了 2 个 64 位的 FIFO。

s3c2440A UART 包括了可编程波特率, 红外传输接收, 一个或两个停止位, 5 位 6 位 7 位 8 位数据长度和奇偶校验。

每个 UART 包含一个波特率发送器, 发送器, 计数器和一个控制单元, 如图 11-1 所示。其波特率发生器可由 PCLK, FCLK/n 或 UEXTCLK (外部输入时钟) 来锁定。发送器和接收器包含了 64 位 FIFO 和数据移位器。数据写到 FIFO 然后在被传送前拷贝到发送移位器。数据通过发送数据引脚 (Tx_{Dn}) 被发出。同时, 接收数据通过接收数据引脚 (Rx_{Dn}) 移入, 然后从移位器拷贝到 FIFO。

10.2 特点

- 基于 DMA 或中断操作的 Rx_{D0}, Tx_{D0}, Rx_{D1}, Tx_{D1}, Rx_{D2} 和 Tx_{D2}
- 有红外和 64 位 FIFO 的 UART 通道 0 和 1
- 有 nRTS₀, nCTS₀, nRTS₁ 和 nCTS₁ 的 UART 通道 0 和 1
- 支持握手的发送和接收

10.3 模块图

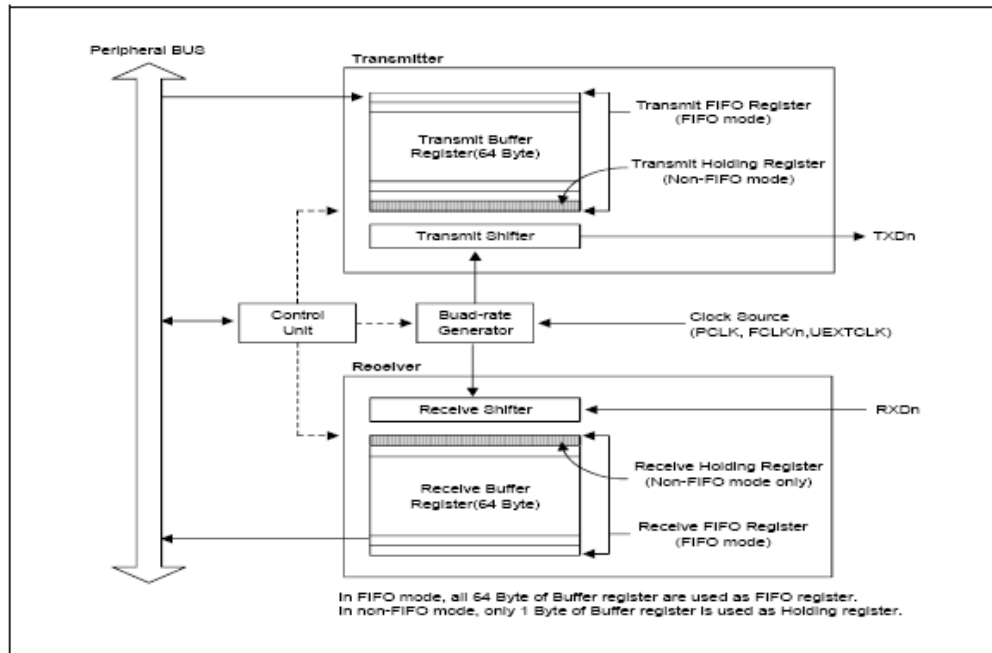


Figure 11-1 UART Block Diagram (with FIFO)

10.4 UART 操作

以下章节描述了 UART 操作，包括数据传输，数据接收，中断生成，波特率生，回送模式，红外模式自动流控制。

10.4.1 数据发送

发送的数据帧是可编程的。其包括一个开始位，5~8 个数据位，一个可选的奇偶校验位和 1~2 个停止位，其可由线性控制寄存器 ULCONn 来设置。发送器也可以产生一个终止条件，其可以对一个帧发送时间强制串行输出为逻辑 0。在当前发送字被完全传输完以后，该模块发送一个终止信号。在终止信号发送后，其串行发送数据到 Tx FIFO。

10.4.2 数据接收

如数据发送，接收的数据帧是可编程的。其包括一个开始位，5~8 个数据位，一个可选的奇偶校验位和 1~2 个停止位，其可由线性控制寄存器 ULCONn 来设置。接收器可以侦测溢出错误，奇偶校验错误，帧错误和终止条件，每个错误都可以设置一个错误标志。

- 溢出错误是指在旧数据被读取前新数据覆盖了旧数据。
- 奇偶校验错误是指接收器侦测到一个不希望的奇偶条件。
- 帧错误是指接收到的数据没有一个有效的停止位。
- 终止条件是指 Rx Dn 输入保持逻辑 0 状态长于一个帧的传输时间。

当接收器在三个字的时间（其间隔根据字长位的设置）没有收到任何数据且 Rx FIFO 在 FIFO 模式下不为空，接收超时条件出现。

10.4.3 自动流控制 (AFC)

s3c2440A 的 UART0 和 UART1 支持有 nRTS 和 nCTS 的自动流。这种情况下, 其可以连接到外部 UART。如果用户想连接 UART 到 Modem, 应使 UMCONn 寄存器中的自动流控制位无效且通过软件控制 nRTS。

在 AFC 中, nRTS 依赖于接收器的条件, nCTS 信号控制发送器的操作。UART 的发送器传输数据到 FIFO 仅当 nCTS 信号被激活 (在 AFC 中, nCTS 意思是对方的 UART FIFO 准备好接收数据)。在 UART 接收数据之前, 当其接收 FIFO 有大于 32 个字节的空闲空间, nRTS 必须被激活; 当其接收 FIFO 有小于 32 个字节的空闲空间, nRTS 必须置非激活。

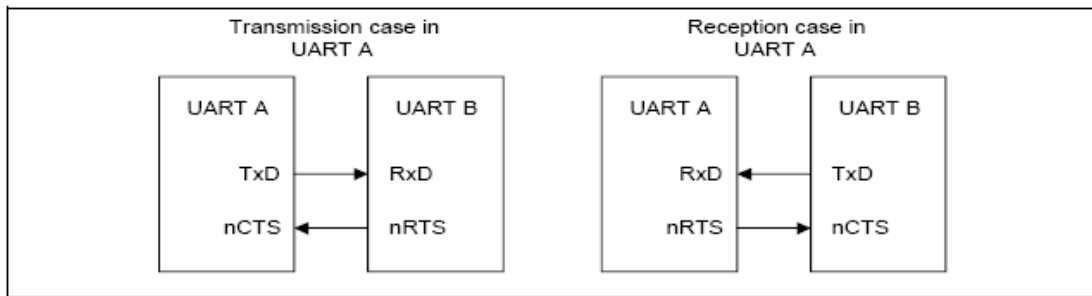


Figure 11-2 UART AFC interface

UART2 不支持 AFC 功能, 因为 s3c2440A 没有 nRTS2 和 nCTS2。

非自动流控制举例 (用软件控制 nRTS 和 nCTS)

使用 FIFO 的接收操作

- (1) 选择接收模式 (中断或 DMA 模式)
- (2) 检查 UFSTATn 寄存器中的 Rx FIFO 计数器。如果其值小于 32, 用户应该将 UMCONn[0] 的值置 1 (激活 nRTS), 如果其等于或大于 32, 用户应该将其置 0。
- (3) 重复步骤 2

有 FIFO 的发送操作

- (1) 选择发送模式 (中断或 DMA 模式)
- (2) 检查 UMSTATn[0] 的值, 如果其值为 1, 用户写数据到发送 FIFO 寄存器。

10.4.4 RS-232C 接口

如果用户想连接 UART 到 Modem 接口上, 就需要 nRTS、nCTS、nDSR、nDTR、DCD 和 nRI。在这种情况下, 用户可通过软件使用通用 IO 端口来控制这些信号, 因为 AFC 不支持 RS-232C 接口。

10.4.5 中断/DMA 请求生成

s3c2440A 的每个 UART 有七个状态 (Tx/Rx/Error) 信号: 溢出错误, 奇偶校验错误, 帧错误, 终止, 接收缓存数据准备好, 发送缓存空和发送移位器空, 其由相应 UART 的状态寄存器 (UTRSTATn/UERSTATn) 指出。

溢出错误, 奇偶校验错误, 帧错误, 终止条件是作为接收错误状态来参考, 每种错误都可以引起接收错误状态中断请求, 如果在控制寄存器 UCONn 中的接收错误状态中断使能位置 1。

当接收错误状态中断请求被检测到, 该信号引起的请求可以通过读 UERSTSTn 寄存器的值

来鉴别。

当接收器传输接收移位器上的数据到在FIFO模式下的接收FIFO寄存器上且接收到的数据数量达到接收FIFO触发水平，接收中断产生，如果在控制寄存器UCONn中接收模式被置1（中断请求或查询模式）。在非FIFO模式下，传输接收移位器的数据到接收保持寄存器将引起在中断请求或查询模式下的接收中断。

当发送器传输接收移位器上的数据到发送 FIFO 寄存器上且在在发送 FIFO 中剩余的数据数量达到发送 FIFO 触发水平，发送中断产生，如果在控制寄存器 UCONn 中的接收模式被选定为中断请求或查询模式。在非 FIFO 模式下，传输发送保持寄存器的数据到发送移位器将引起在中断请求或查询模式下的发送中断。

如果在控制寄存器中发送模式和接收模式被选定为 DMA_n 请求模式，这样 DMA_n 请求将取代上述情形下的接收或发送中断出现。

类型	FIFO 模式	非 FIFO 模式
Rx interrupt	只要接收数据到达接收 FIFO 的触发等级，中断产生。 当 FIFO 中的数据数量没有达到接收 FIFO 的触发等级且在三个字时间内没有接收到任何数据（接收超时），产生中断。该时间根据字长位设置。	只要接收缓存为空，由接收保持寄存器生成。
Tx interrupt	只要发送数据达到发送 FIFO 触发等级，中断产生。	只要发送缓存为空，由接收发送寄存器生成。
Error interrupt	当帧错误，奇偶错误或终止信号被检测到，中断产生。 当达到接收 FIFO 的上限没有读出数据，中断产生。	由错误生成。但是如果其他错误同时产生，仅一个中断产生。

10.4.6 UART 错误状态 FIFO

除了接收FIFO寄存器，UART有一个错误状态FIFO。错误状态FIFO是指出FIFO寄存器中有错的数据。当有错的数据准备读出，错误中断就产生。为了清除错误状态FIFO，有错误的URXH_n和UERSTAT_n必须被读出。

举例

假定UART接收FIFO顺序接收到A、B、C、D、E且在接收B时出现帧错误，在接收D时出现奇偶校验错误。因为错误接收的字符没有被读取，所以UART接收错误没有产生错误中断。一旦字符被读取，错误中断产生。

如图 11-3 所示UART接收 4 个字符包括 2 个错误。

时间	顺序流	错误中断	注
#0	当没有字符被读出	-	-
#1	接收到A、B、C、D、E	帧错误（B中）中断出现	‘B’必须被读出
#2	在A被读出后	-	-
#3	在 B 被读出后	-	-
#4	在 C 被读出后	奇偶校验错误（D中）中断出现	‘D’必须被读出
#5	在 D 被读出后	-	-
#6	在 E 被读出后	-	-

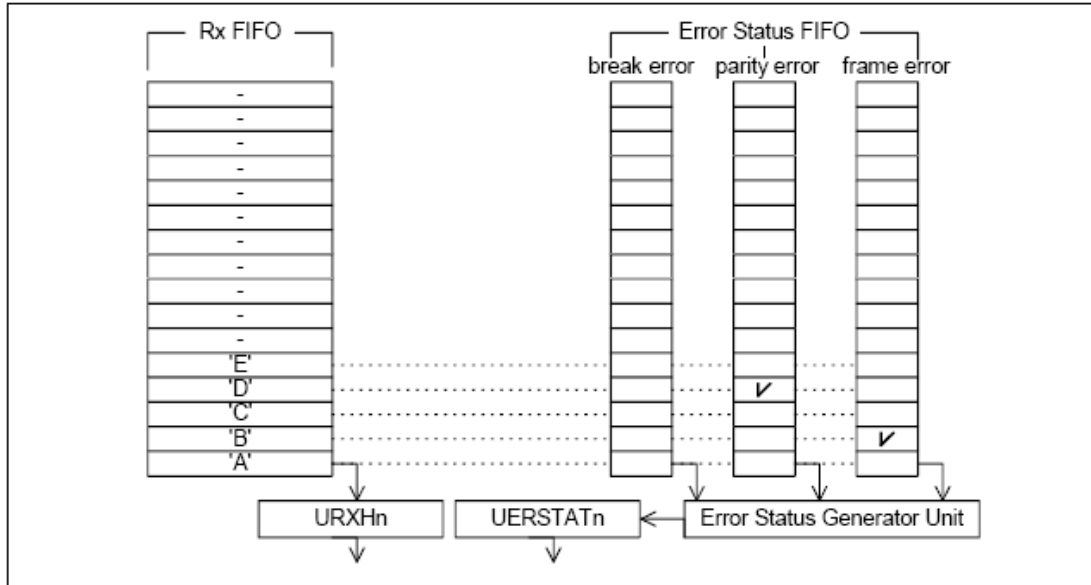


Figure 11-3. Example showing UART Receiving 5 Characters with 2 Errors

10.4.7 波特率产生

每个UART波特率发生器为发送器和接收器提供串行时钟。波特率发生器的源时钟可以选择s3c2440A的内部系统时钟或UEXTCLK。换言之，被除数可以设定UCONn中的时钟选择来选择。波特率时钟通过 16 和由UART波特率除数寄存器（UBRDIVn）中定义的 16 位除数来分频源时钟（PCLK, FCLK/n or UEXTCLK）产生。UBRDIVn由以下表达式来决定。

$$\text{UBRDIVn} = (\text{int}) (\text{UART clock} / (\text{baud rate} \times 16)) - 1$$

(UART clock: PCLK, FCLK/n or UEXTCLK)

UBRDIVn应该是从 1 到 $2^{16}-1$ ，当使用小于PCLK的外部时钟UEXTCLK时应该设置 0。

例如如果波特率是 115200bps且UART时钟是 40MHz, UBRDIVn是

$$\begin{aligned} \text{UBRDIVn} &= (\text{int})(40000000 / (115200 \times 16)) - 1 \\ &= (\text{int})(21.7) - 1 \text{ [round to the nearest whole number]} \\ &= 22 - 1 = 21 \end{aligned}$$

波特率错误公差 Baud-Rate Error Tolerance

UART帧错误应该少于 1.87%(3/160).

$$t_{\text{UPCLK}} = (\text{UBRDIVn} + 1) \times 16 \times 1\text{Frame} / \text{PCLK} \quad t_{\text{UPCLK}} : \text{Real UART Clock}$$

$$t_{\text{UEXACT}} = 1\text{Frame} / \text{baud-rate} \quad t_{\text{UEXACT}} : \text{Ideal UART Clock}$$

$$\text{UART error} = (t_{\text{UPCLK}} - t_{\text{UEXACT}}) / t_{\text{UEXACT}} \times 100\%$$

注:

- 1Frame = start bit + data bit + parity bit + stop bit.
- 在特定条件下，我们支持波特率最高达到 921.6K bps。例如，当PCLK为 60MHz, 你可以在 1.69UART错误公差下使用 921.6K bps under UART error of 1.69%.

10.4.8 回送模式

s3c2440A UART提供了一个参考作为回送模式的测试模式，以帮助孤立通讯连接中的错误。这个模式结构上使能在UART中的RXD和TXD连接。在此模式下发送的数据通过RXD接收到接收器。该特点是允许处理器验证内部传输和每个SIO通道的接收数据通道。该模式通过设置UART控制寄存器UCONn的回送位来选定。

10.4.9 红外模式

S3c2440A UART模块支持红外发送和接收，其通过设置UART线性控制寄存器ULCONn的红外模式位来选定。如图 11-4 所示如何执行一个红外模式。

在红外发送模式，发送脉冲是 $3/16$ 的正常串行发送率（当发送数据位为 0 时）。在红外接收模式下，接收器必须检测 $3/16$ 的脉冲期间来识别一个 0 值（看如图 11-7 和 11-6 所示的帧时序图）。

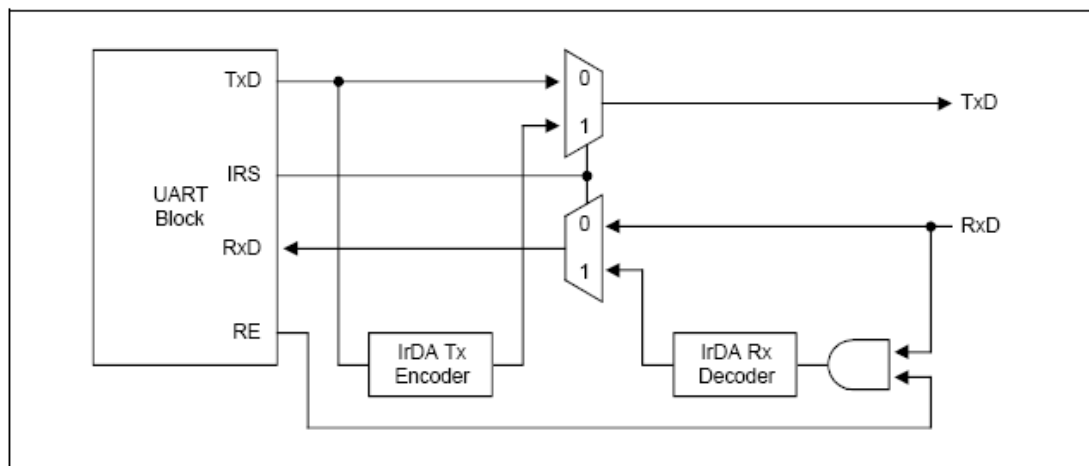


Figure 11-3. IrDA Function Block Diagram

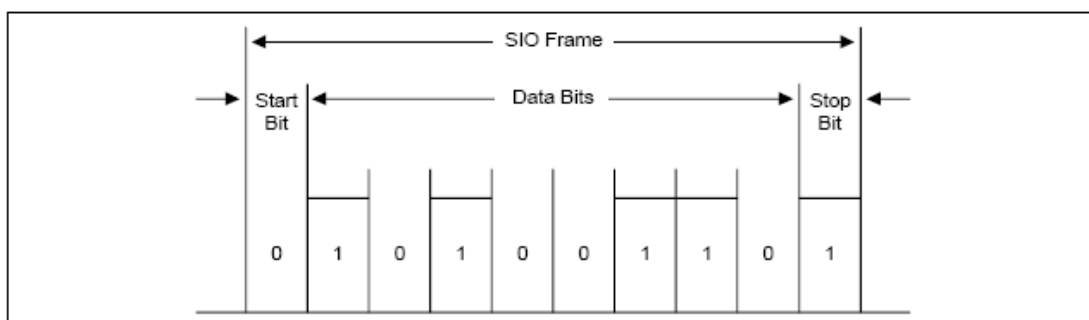


Figure 11-4. Serial I/O Frame Timing Diagram (Normal UART)

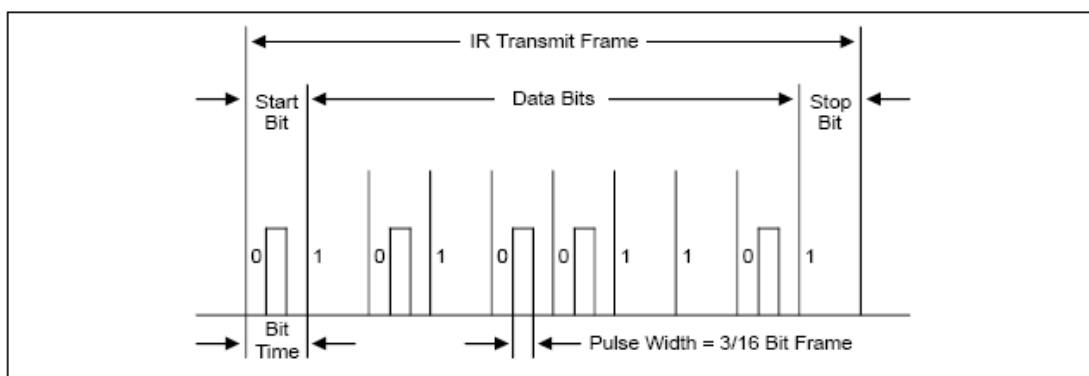


Figure 11-5. Infrared Transmit Mode Frame Timing Diagram

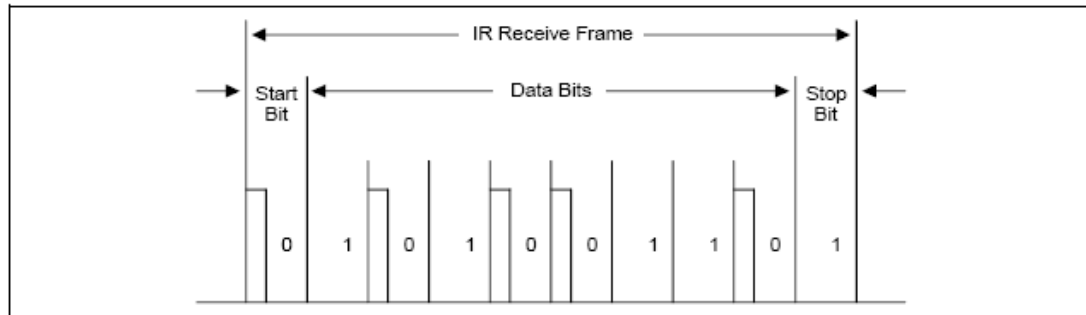


Figure 11-6. Infrared Receive Mode Frame Timing Diagram

10.5 UART 特殊寄存器

- (1) UART 线性控制寄存器 (ULCONn)
- (2) UART 控制寄存器 (UCONn)
- (3) UART FIFO 控制寄存器(UFCONn)
- (4) UART MODEM 控制寄存器(UMCONn)
- (5) UART 接收发送状态寄存器(UTRSTATn)
- (6) UART 错误状态寄存器(UERSTATn)
- (7) UART FIFO 状态寄存器(UFSTATn)
- (8) UART MODEM 状态寄存器(UMSTATn)
- (9) UART 发送缓存寄存器(UTXHn)
- (10) UART 接收缓存寄存器(URXHn)
- (11) UART 波特率除数寄存器 (UBRDIVn)

10.5.1 UART 线性控制寄存器

UART LINE CONTROL REGISTER (ULCONn)

寄存器	地址	读写	描述	复位值
ULCON0	0x50000000	R/W	UART通道 0 线性控制寄存器	0x00
ULCON1	0x50004000	R/W	UART通道 1 线性控制寄存器	0x00
ULCON2	0x50008000	R/W	UART通道 2 线性控制寄存器	0x00

ULCONn	位	描述	初始值
Reserved	[7]		0
Infrared Mode	[6]	决定是否使用红外模式 0 = 正常模式操作 1 = 红外接收发送模式	0
Parity Mode	[5:3]	在UART发送接收操作中定义奇偶码的生成和检验类型 0xx = No parity 100 = Odd parity 101 = Even parity 110 = Parity forced/checked as 1 111 = Parity forced/checked as 0	000
Number of Stop Bit	[2]	定义度搜按个停止位用于帧末信号 0 = 每帧一个停止位 1 = 每帧两个停止位	0
Word Length	[1:0]	指出发送接收每帧的数据位数 00 = 5-bits 01 = 6-bits 10 = 7-bits 11 = 8-bits	00

10.5.2 UART 控制寄存器

UART CONTROL REGISTER (UCONn)

寄存器	地址	读写	描述	复位值
UCON0	0x50000004	R/W	UART通道 0 控制寄存器	0x00
UCON1	0x50004004	R/W	UART通道 1 控制寄存器	0x00
UCON2	0x50008004	R/W	UART通道 2 控制寄存器	0x00

UCONn	位	描述	初始值
FCLK divider	[15:12]	<p>当UART时钟源选为FCLK/n时的分频器值。 n由UCON0[15:12], UCON1[15:12], UCON2[14:12]来决定。</p> <p>UCON2[15]是FCLK/n 时钟使能位。 设置n从 7 到 21, 使用UCON0[15:12], 设置n从 22 到 36, 使用UCON1[15:12], 设置n从 37 到 43, 使用UCON2[14:12],</p> <p>UCON2[15]: 0 = 无效FCLK/n时钟 1 = 使能FCLK/n时钟 对于UCON0, UART时钟= FCLK / (divider+6), 其 divider>0, UCON1, UCON2 必须为 0。 ex) 1: UART clock = FCLK/7, 2: UART clock = FCLK/8 3: UART clock = FCLK/9, ... , 15: UART clock = FCLK/21 对于UCON1, UART时钟= FCLK / (divider+21), 其 divider>0, UCON0, UCON2 必须为 0 ex) 1: UART clock = FCLK/22, 2: UART clock = FCLK/23 3: UART clock = FCLK/24, ... , 15: UART clock = FCLK/36 对于UCON2, UART时钟= FCLK / (divider+36), 其 divider>0, UCON0, UCON1 必须为 0。 ex) 1: UART clock = FCLK/37, 2: UART clock = FCLK/38 3: UART clock = FCLK/39, ... , 7: UART clock = FCLK/43 如果UCON0/1[15:12]和UCON2[14:12] 都是 0, 分频器为 44, 则UART时钟= FCLK/44 总的除数范围是从 7 到 44。</p>	0000
Clock Selection	[11:10]	<p>为波特率选择PCLK, UEXTCLK或FCLK/n。 $UBRDIVn = (int)(selected\ clock / (baudrate \times 16)) - 1$ 00, 10 = PCLK, 01 = UEXTCLK, 11 = FCLK/n (如果你选择FCLK/n,你应该在选择或取消选择FCLK/n加上 注释代码)</p>	0
Tx Interrupt Type	[9]	<p>发送中断请求类型 0 = Pulse (在非FIFO模式下一旦发送缓存变空或在FIFO模式 下发送缓存达到发送FIFO触发水平, 则中断请求) 1 = Level (当非FIFO模式下发送缓存变空或在FIFO模式下发 送缓存达到发送FIFO触发水平, 则中断请求)</p>	0
Rx Interrupt Type	[8]	<p>接收中断请求类型 0 = Pulse (在非FIFO模式下一旦接收缓存接收到数据, 或在</p>	0

		FIFO模式下, 发送缓存达到发送FIFO触发水平, 则中断请求) 1 = Level (在非FIFO模式下接收缓存接收数据, 或在FIFO模式下发送缓存达到发送FIFO触发水平, 则中断请求)	
Rx Time Out Enable	[7]	在UART FIFO有效时, 使能接收超时中断。该中断是一个接收中断。 0 = 无效 1 = 有效	0
Rx Error Status Interrupt Enable	[6]	使能UART对异常产生中断, 例如在接收期间终止信号, 帧错误, 奇偶校验错误和溢出错误。 0 = 不产生接收错误状态中断 1 = 产生接收错误状态中断。	0
Loopback Mode	[5]	对该位置位将引起UART进入回送模式。该模式仅用于测试目的。 0 = 正常操作 1 = 回送模式	0
Send Break Signal	[4]	对该位置位将引起UART在一个帧时间内发送一个终止信号。在发送终止信号后该位自动清零。 0 = 正常发送 1 = 发终止信号	0
Transmit Mode	[3:2]	决定哪种功能用来写数据到UART发送缓存寄存器 (UART Tx Enable/Disable)。 00 = 无效 01 = 中断请求或查询模式 10 = DMA0 请求 (仅对UART0), DMA3 请求 (仅对UART2) 11 = DMA1 请求 (仅对UART1)	00
Receive Mode	[1:0]	决定哪种功能用来读取UART接收缓存寄存器中的数据 (UART Rx Enable/Disable)。 00 = 无效 01 = 中断请求或查询模式 10 = DMA0 请求 (Only for UART0), DMA3 请求 (Only for UART2) 11 = DMA1 请求 (Only for UART1)	00

10.5.3 UART FIFO 控制寄存器

UART FIFO CONTROL REGISTER (UFCONn)

寄存器	地址	读写	描述	复位值
UFCON0	0x50000008	R/W	UART通道 0FIFO控制寄存器	0x00
UFCON1	0x50004008	R/W	UART通道 1FIFO控制寄存器	0x00
UFCON2	0x50008008	R/W	UART通道 2FIFO控制寄存器	0x00

UFCONn	位	描述	初始值
Tx FIFO Trigger Level	[7:6]	决定发送FIFO的触发等级 00 = Empty 01 = 16-byte 10 = 32-byte 11 = 48-byte	00
Rx FIFO Trigger Level	[5:4]	决定接收FIFO的触发等级 00 = 1-byte 01 = 8-byte 10 = 16-byte 11 = 32-byte	00
Reserved	[3]		0
Tx FIFO Reset	[2]	在重置FIFO后自动清除 0 = Normal 1= Tx FIFO reset	0
Rx FIFO Reset	[1]	在重置FIFO后自动清除 0 = Normal 1= Rx FIFO reset	0
FIFO Enable	[0]	0 = Disable 1 = Enable	0

10.5.4 UART MODEM 控制寄存器

UART MODEM CONTROL REGISTER (UMCONn)

寄存器	地址	读写	描述	复位值
UMCON0	0x5000000C	R/W	UART通道 0MODEM控制寄存器	0x0
UMCON1	0x5000400C	R/W	UART通道 1MODEM控制寄存器	0x0
保留	0x5000800C	-	-	-

UMCONn	位	描述	初始值
Reserved	[7:5]	这些位必须为 0	000
Auto Flow Control (AFC)	[4]	0 = 无效 1 = 有效	0
Reserved	[3:1]	这些位必须为 0	000
Request to Send	[0]	如果AFC位有效, 则该值被忽略。在这种情况下 S3C2440A将自动控制nRTS。 如果AFC位无效, nRTS必须由软件控制。 0 = 'H' level (去激活nRTS) 1 = 'L' level (激活nRTS)	0

10.5.5 UART 接收发送状态寄存器

UART TX/RX STATUS REGISTER (UTRSTATn)

寄存器	地址	读写	描述	复位值
UTRSTAT0	0x50000010	R/W	UART通道 0 接收发送状态寄存器	0x00
UTRSTAT1	0x50004010	R/W	UART通道 1 接收发送状态寄存器	0x00
UTRSTAT2	0x50008010	R/W	UART通道 2 接收发送状态寄存器	0x00

UTRSTATn	位	描述	初始值
Transmitter empty	[2]	当发送缓存寄存器中没有有效值且发送移位寄存器空, 则自动置 1。 0 = 非空 1 = 发送器空(发送缓存和移位寄存器)	1
Transmit buffer empty	[1]	当发送缓存寄存器为空, 则自动置于 1 0 = 发送缓存寄存器不为空 1 = 发送缓存寄存器 为空 (在非FIFO模式下, 中断或DMA被请求。在FIFO模式下当发送FIFO触发等级设为 00 (空) 时, 中断或DMA被请求) 如果UART使用FIFO, 用户应该检查寄存器UFSTAT中的 Tx FIFO Coun位和Tx FIFO Full 位取代对此位的检查。	1
Receive buffer data ready	[0]	只要接收缓存寄存器保留通过RXDn端口接收的有效值, 则自动置 1。 0 = 缓存寄存器为空 1 = 缓存寄存器接收到数据 (在非FIFO模式下, 请求中断或DMA) 如果UART使用FIFO, 用户应该在UFSTAT中的Rx FIFO Count位和Rx FIFO Full位取代对此位的检查。	0

10.5.6 UART 错误状态寄存器

UART ERROR STATUS REGISTER (UERSTATn)

寄存器	地址	读写	描述	复位值
UERSTAT0	0x50000014	R/W	UART通道 0 错误状态寄存器	0x00
UERSTAT1	0x50004014	R/W	UART通道 1 错误状态寄存器	0x00
UERSTAT2	0x50008014	R/W	UART通道 2 错误状态寄存器	0x00

UTRSTATn	位	描述	初始值
Break Detect	[3]	自动置 1 来指出一个终止信号已发出 0 = No break receive 1 = Break receive (已请求中断)	0
Frame Error	[2]	只要在接收操作中帧错误出现则自动置 1 0 = 接收过程中无帧错误 1 = 帧错误(已请求中断)	0
Parity Error	[1]	只要在接收操作中出现奇偶校验错误则自动置 1 0 = 接受过程中无奇偶校验错误 1 = 奇偶校验错误 (已请求中断)	0
Overrun Error	[0]	只要在接收过程中出现溢出错误则自动置 1 0 = 接收过程中无溢出错误 1 = 溢出错误 (已请求中断)	0

10.5.7 UART FIFO 状态寄存器

UART FIFO STATUS REGISTER (UFSTATn)

寄存器	地址	读写	描述	复位值
UFSTAT0	0x50000018	R/W	UART通道 0FIFO状态寄存器	0x00
UFSTAT1	0x50004018	R/W	UART通道 1FIFO状态寄存器	0x00
UFSTAT2	0x50008018	R/W	UART通道 2FIFO状态寄存器	0x00

UFSTATn	位	描述	初始值
Reserved	[15]		0
Tx FIFO Full	[14]	只要在接收操作中接收FIFO满, 则自动置 1。 0 = 0-byte ≤ Tx FIFO data ≤ 63-byte 1 = Full	0
Tx FIFO Count	[13:8]	发送FIFO中的数据数量	0
Reserved	[7]		0
Rx FIFO Full	[6]	只要在接收操作中接收FIFO满, 则自动置 1。 0 = 0-byte ≤ Rx FIFO data ≤ 63-byte 1 = Full	0
Rx FIFO Count	[5:0]	接收FIFO中的数据数量	0

10.5.8 UART MODEM 状态寄存器

UART MODEM STATUS REGISTER (UMSTATn)

寄存器	地址	读写	描述	复位值
UFSTAT0	0x5000001C	R/W	UART通道 0MODEM状态寄存器	0x00
UFSTAT1	0x5000401C	R/W	UART通道 1MODEM状态寄存器	0x00
保留	0x5000801C	R/W	保留	-

UMSTATn	位	描述	初始值
Delta CTS	[4]	Indicate that the nCTS input to the S3C2440A has changed state since the last time it was read by CPU. (参考图 11-8.) 0 = 未变 1 = 已变	0
Reserved	[3:1]		0
Clear to Send	[0]	0 = CTS信号未激活 (nCTS引脚为高电平) 1 = CTS信号激活 (nCTS引脚为低电平)	0

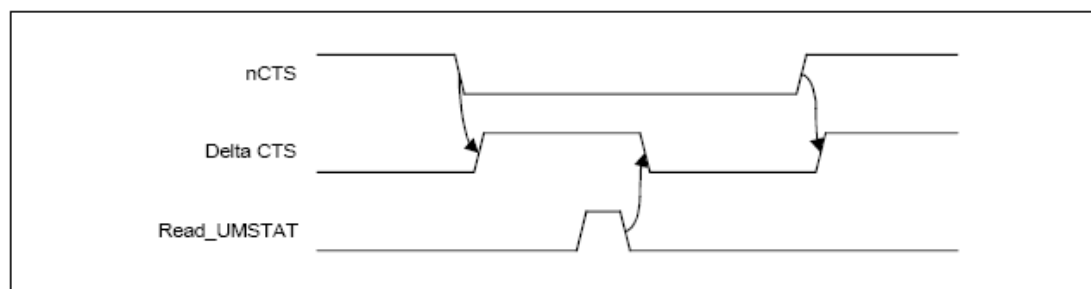


Figure 11-7. nCTS and Delta CTS Timing Diagram

10.5.9 UART 发送缓存寄存器

UART TRANSMIT BUFFER REGISTER (HOLDING REGISTER & FIFO REGISTER)

寄存器	地址	读写	描述	复位值
UTXH0	0x50000020(L) 0x50000023(B)	W (byte)	UART通道 0 发送缓存寄存器	-
UTXH1	0x50004020(L) 0x50004023(B)	W (byte)	UART通道 1 发送缓存寄存器	-
UTXH2	0x50008020(L) 0x50008023(B)	W (byte)	UART通道 2 发送缓存寄存器	-

UTXHn	位	描述	初始值
TXDATAn	[7:0]	UARTn的发送数据	-

10.5.10 UART 接收缓存寄存器

UART RECEIVE BUFFER REGISTER (HOLDING REGISTER & FIFO REGISTER)

寄存器	地址	读写	描述	复位值
URXH0	0x50000024(L) 0x50000027(B)	R (byte)	UART通道 0 接收缓存寄存器	-
URXH1	0x50004024(L) 0x50004027(B)	R (byte)	UART通道 1 接收缓存寄存器	-
URXH2	0x50008024(L) 0x50008027(B)	R (byte)	UART通道 2 接收缓存寄存器	-

URXHn	位	描述	初始值
RXDATAn	[7:0]	UARTn的接收数据	-

10.5.11 UART 波特率除数寄存器

UART BAUD RATE DIVISOR REGISTER (UBRDIVn)

在 UART 模块中有三个 UART 波特率除数寄存器 UBRDIV0, UBRDIV1 和 UBRDIV2。存储在波特率除数寄存器 UBRDIVn 中的值用于决定串行发送接收时钟率（波特率）如下：

$$\text{UBRDIVn} = (\text{int}) (\text{UART clock} / (\text{baud rate} \times 16)) - 1$$

(UART clock: PCLK, FCLK/n or UEXTCLK)

UBRDIVn 应该从 1 到 $(2^{16}-1)$ ，仅当使用小于 PCLK 的 UEXTCLK 时可以设为 0。

例如如果波特率为 115200bps 且 UART 时钟为 40MHz, UBRDIVn 是：

$$\text{UBRDIVn} = (\text{int}) (40000000 / (115200 \times 16)) - 1$$

$$= (\text{int}) (21.7) - 1 \text{ [round to the nearest whole number]}$$

$$= 22 - 1 = 21$$

寄存器	地址	读写	描述	复位值
UBRDIV0	0x50000028	R/W	波特率除数寄存器 0	-
UBRDIV1	0x50004028	R/W	波特率除数寄存器 1	-
UBRDIV2	0x50008028	R/W	波特率除数寄存器 2	-

UBRDIVn	位	描述	初始值
UBRDIV	[15:0]	波特率分频值 UBRDIVn > 0 使用 UEXTCLK 作为输入时钟, UBRDIVn 可以置 0	-

第十四章 中断控制器

S3C2440A 中的中断控制器可以从 60 个中断源接收中断请求。这些中断源由内部外设提供，例如 DMA 控制器、UART、IIC 等。在这些中断源中，UARTn、AC97 和外部中断 EINTn 对于中断控制器来说是或逻辑。

当接收来自内部外设和外部中断请求引脚的多个中断请求时，在仲裁过程后中断控制器请求 ARM920T 的 FIR 或 IRQ 中断。

仲裁过程依赖于硬件优先级逻辑且其结果写入中断未决寄存器，其帮助用户通报那些由不同中断源生成的中断。

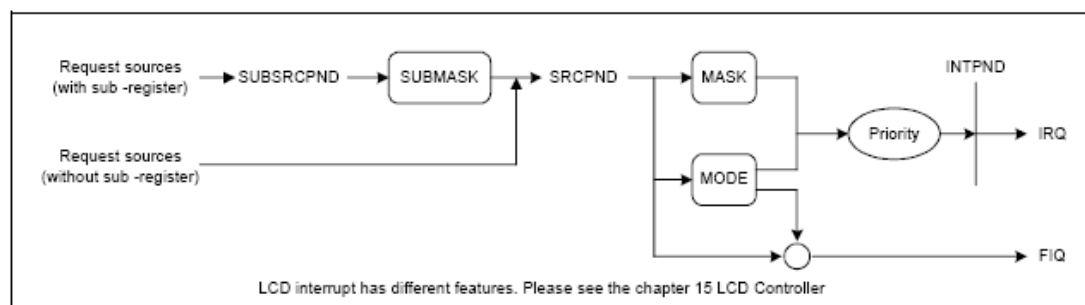


Figure 14-1. Interrupt Process Diagram

14.1 中断控制器操作

14.1.1 程序状态寄存器（PSR）的 F 位和 L 位

如果在 ARM920T 的 CPU 中的程序状态寄存器的 F 位置 1，CPU 不能接收来自中断控制器的 FIR，如果程序状态寄存器的 L 位被置 1，则 CPU 不能接收来自中断控制器的 IRQ。所以，通过清 0 程序状态寄存器中的 F 位和 L 位且清 0INTMSK 寄存器中的相应位，中断控制器可以接收中断。

14.1.2 中断模式

ARM920T 有两个类型的中断模式：FIR 和 IRQ。所有的中断源决定了哪种模式的中断请求。

14.1.3 中断未决寄存器

S3C2440A 有两个中断未决寄存器：源未决寄存器（SRCPND）和中断未决寄存器（INTPND）。这些未决寄存器指出了是否中断请求未决。当中断源请求中断服务，中断源未决寄存器（SRCPND）的相应位置 1，同时在仲裁过程后中断未决寄存器（INTPND）的一个位自动的置 1。如果中断被屏蔽，则 SRCPND 寄存器的相应位被置 1。但不引起 INTPND 的位改变。如果 INTPND 的未决位置位，只要 L 和 F 标志为 0，中断服务程序开始运行。SRCPND 和 INTPND 寄存器可以被读写，因此中断服务程序可以通过对 SRCPND 寄存器相应位写 1 来清除未决条件，对于 INTPND 寄存器清除未决条件也采用一样的方法。

14.1.4 中断屏蔽寄存器

如果相应的屏蔽位置 1，寄存器指出某个中断无效。如果 INTMSK 的一个中断屏蔽位是 0，中断将被正常的服务。如果相应的屏蔽位置 1 且中断产生，则源未决位被置位。

14.1.5 中断源

中断控制器支持 60 个中断源如下表所示

源	描述	仲裁组
INT_ADC	ADC EOC and Touch interrupt (INT_ADC_S/INT_TC)	ARB5
INT_RTC	RTC alarm interrupt	ARB5
INT_SPI1	SPI1 interrupt	ARB5
INT_UART0	UART0 Interrupt (ERR, RXD, and TXD)	ARB5
INT_IIC	IIC interrupt	ARB4
INT_USBH	USB Host interrupt	ARB4
INT_USBD	USB Device interrupt	ARB4
INT_NFCON	Nand Flash Control Interrupt	ARB4
INT_UART1	UART1 Interrupt (ERR, RXD, and TXD)	ARB4
INT_SPI0	SPI0 interrupt	ARB4
INT_SDI	SDI interrupt	ARB3
INT_DMA3	DMA channel 3 interrupt	ARB3
INT_DMA2	DMA channel 2 interrupt	ARB3
INT_DMA1	DMA channel 1 interrupt	ARB3
INT_DMA0	DMA channel 0 interrupt	ARB3
INT_LCD	LCD interrupt (INT_FrSyn and INT_FiCnt)	ARB3
INT_UART2	Interrupt (ERR, RXD, and TXD) UART2	ARB2
INT_TIMER4	Timer4 interrupt	ARB2
INT_TIMER3	Timer3 interrupt	ARB2
INT_TIMER2	Timer2 interrupt	ARB2
INT_TIMER1	Timer1 interrupt	ARB2
INT_TIMER0	Timer0 interrupt	ARB2
INT_WDT_AC97	Watch-Dog timer interrupt(INT_WDT, INT_AC97)	ARB1
INT_TICK	RTC Time tick interrupt	ARB1
nBATT_FLT	Battery Fault interrupt	ARB1
INT_CAM	Camera Interface (INT_CAM_C, INT_CAM_P)	ARB1
EINT8_23	External interrupt 8 – 23	ARB1
EINT4_7	External interrupt 4 – 7	ARB1
EINT3	External interrupt 3	ARB0
EINT2	External interrupt 2	ARB0
EINT1	External interrupt 1	ARB0
EINT0	External interrupt 0	ARB0

14.1.6 子中断源

子源	描述	源
INT_AC97	AC97 interrupt	INT_WDT_AC97
INT_WDT	Watchdog interrupt	INT_WDT_AC97
INT_CAM_P	P-port capture interrupt in camera interface	INT_CAM
INT_CAM_C	C-port capture interrupt in camera interface	INT_CAM
INT_ADC_S	ADC interrupt	INT_ADC
INT_TC	Touch screen interrupt (pen up/down)	INT_ADC
INT_ERR2	UART2 error interrupt	INT_UART2
INT_TXD2	UART2 transmit interrupt	INT_UART2
INT_RXD2	UART2 receive interrupt	INT_UART2
INT_ERR1	UART1 error interrupt	INT_UART1
INT_TXD1	UART1 transmit interrupt	INT_UART1
INT_RXD1	UART1 receive interrupt	INT_UART1
INT_ERR0	UART0 error interrupt	INT_UART0
INT_TXD0	UART0 transmit interrupt	INT_UART0
INT_RXD0	UART0 receive interrupt	INT_UART0

14.1.7 中断优先级生成模块

32 个中断请求优先级逻辑包括基于仲裁器的 7 个翻转：如图 14-1 所示的 6 个一级仲裁器和一个 2 级仲裁器

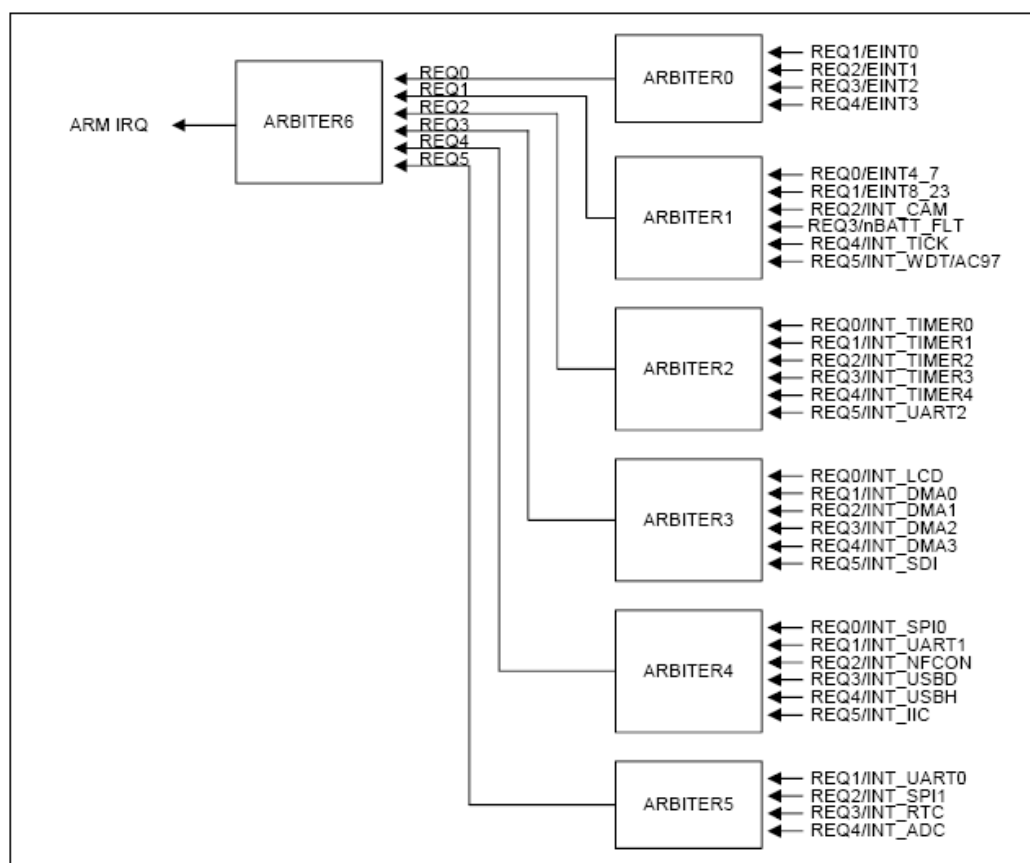


Figure 14.2. Priority Generating Block

14.1.8 中断优先级

每个仲裁器基于一个位仲裁器模式控制 (ARB_MODE) 和选择控制信号 (ARB_SEL) 的两位来处理 6 个中断请求。

如果 ARB_SEL 位是 00b, 优先级是 REQ0, REQ1, REQ2, REQ3, REQ4, 和 REQ5.

如果 ARB_SEL 位是 01b, 优先级是 REQ0, REQ2, REQ3, REQ4, REQ1, 和 REQ5.

如果 ARB_SEL 位是 10b, 优先级是 REQ0, REQ3, REQ4, REQ1, REQ2, 和 REQ5.

如果 ARB_SEL 位是 11b, 优先级是 REQ0, REQ4, REQ1, REQ2, REQ3, 和 REQ5.

注意仲裁器的 REQ0 总是有最高优先级, REQ5 总是有最低优先级。此外通过改变 ARB_SEL 位, 我们可以翻转 REQ1 到 REQ4 的优先级。

如果 ARB_MODE 位置 0, ARB_SEL 位不会自动改变, 使得仲裁器在一个固定优先级的模式下操作 (注意在此模式下, 我们通过手工改变 ARB_SEL 位来配置优先级)。另外, 如果 ARB_MODE 位是 1, ARB_SEL 位以翻转的方式改变。例如如果 REQ1 被服务, 则 ARB_SEL 位自动的变为 01b, 把 REQ1 放到最低的优先级。ARB_SEL 变化的详细规则如下:

如果 REQ0 或 REQ5 被服务, ARB_SEL 位完全不会变化。

如果 REQ1 被服务, ARB_SEL 位变为 01b。

如果 REQ2 被服务, ARB_SEL 位变为 10b。

如果 REQ3 被服务, ARB_SEL 位变为 11b。

如果 REQ4 被服务, ARB_SEL 位变为 00b。

14.2 中断控制器特殊寄存器

在中断控制器中有五个控制寄存器: 中断源未决寄存器, 中断模式寄存器, 屏蔽寄存器, 优先级寄存器和中断未决寄存器。

中断源的所有中断请求首先都是在中断源未决寄存器中等级。基于中断模式寄存器中断请求分为两组包括快速中断请求和中断请求。对于多 IRQ 的仲裁过程是基于优先级寄存器。

14.2.1 中断源未决寄存器

SOURCE PENDING REGISTER (SRCPND)

SRCPND 寄存器包括 32 位, 每位与一个中断源相关。如果相应的中断源产生中断请求且等待中断服务, 则每个位置 1。因此这个寄存器指出那个中断源在等待请求服务。注意 SRCPND 的每个位都由中断源自动置位, 不管 INTMASK 寄存器的屏蔽位。此外, SRCPND 寄存器不会受到中断控制器的优先级逻辑的影响。

对于一个特定中断源的中断服务程序中, SRCPND 寄存器的相应位必须被清除目的是下次能正确得到同一个中断源的中断请求。如果你从中断服务程序返回却没有清除该位, 中断控制器将操作好像又有同一个中断源的中断请求到来。换言之, 如果 SRCPND 的一个特殊位置 1, 其总是认为一个有效的中断请求等待服务。

清除相应位的时间依赖于用户的需求。如果你想收到另一个来此同一个中断源的有效请求，你应该清除相应的位，然后使能中断。

你可以通过写数据到这个寄存器来清除 SRCPND 寄存器的某个位。你可以通过对相应位置 1 来清除相应位。如果你对相应位写 0，则该位的数值保持不变。

寄存器	地址	读写	描述	复位值
SRCPND	0x4A000000	R/W	指出中断请求的状态 0 = 中断未请求 1 = 中断源已经申请中断	0x00000000

SRCPND	位	描述	初始值
INT_ADC	[31]	0 = 未请求 1 = 已请求	0
INT_RTC	[30]	0 = 未请求 1 = 已请求	0
INT_SPI1	[29]	0 = 未请求 1 = 已请求	0
INT_UART0	[28]	0 = 未请求 1 = 已请求	0
INT_IIC	[27]	0 = 未请求 1 = 已请求	0
INT_USBH	[26]	0 = 未请求 1 = 已请求	0
INT_USBD	[25]	0 = 未请求 1 = 已请求	0
INT_NFCON	[24]	0 = 未请求 1 = 已请求	0
INT_UART1	[23]	0 = 未请求 1 = 已请求	0
INT_SPI0	[22]	0 = 未请求 1 = 已请求	0
INT_SDI	[21]	0 = 未请求 1 = 已请求	0
INT_DMA3	[20]	0 = 未请求 1 = 已请求	0
INT_DMA2	[19]	0 = 未请求 1 = 已请求	0
INT_DMA1	[18]	0 = 未请求 1 = 已请求	0
INT_DMA0	[17]	0 = 未请求 1 = 已请求	0
INT_LCD	[16]	0 = 未请求 1 = 已请求	0
INT_UART2	[15]	0 = 未请求 1 = 已请求	0
INT_TIMER4	[14]	0 = 未请求 1 = 已请求	0
INT_TIMER3	[13]	0 = 未请求 1 = 已请求	0
INT_TIMER2	[12]	0 = 未请求 1 = 已请求	0
INT_TIMER1	[11]	0 = 未请求 1 = 已请求	0
INT_TIMER0	[10]	0 = 未请求 1 = 已请求	0
INT_WDT_AC97	[9]	0 = 未请求 1 = 已请求	0
INT_TICK	[8]	0 = 未请求 1 = 已请求	0
nBATT_FLT	[7]	0 = 未请求 1 = 已请求	0
INT_CAM	[6]	0 = 未请求 1 = 已请求	0
EINT8_23	[5]	0 = 未请求 1 = 已请求	0
EINT4_7	[4]	0 = 未请求 1 = 已请求	0
EINT3	[3]	0 = 未请求 1 = 已请求	0
EINT2	[2]	0 = 未请求 1 = 已请求	0
EINT1	[1]	0 = 未请求 1 = 已请求	0
EINT0	[0]	0 = 未请求 1 = 已请求	0

14.2.2 中断模式寄存器

INTERRUPT MODE REGISTER (INTMOD)

该寄存器包括 32 位，每位与一个中断源相关。如果某位置 1，相应的中断将在 FIQ 模式下处理。否则在 IRQ 模式下操作。

请注意仅有一个中断源能够在 FIR 模式下服务，也就是说，INTMOD 仅有一个位可以被置 1。

寄存器	地址	读写	描述	复位值
INTMOD	0x4A000004	R/W	中断模式寄存器 0 = IRQ 模式 1 = FIR 模式	0x00000000

INTMOD	位	描述	初始值
INT_ADC	[31]	0 = IRQ 1 = FIR	0
INT_RTC	[30]	0 = IRQ 1 = FIR	0
INT_SPI1	[29]	0 = IRQ 1 = FIR	0
INT_UART0	[28]	0 = IRQ 1 = FIR	0
INT_IIC	[27]	0 = IRQ 1 = FIR	0
INT_USBH	[26]	0 = IRQ 1 = FIR	0
INT_USBD	[25]	0 = IRQ 1 = FIR	0
INT_NFCON	[24]	0 = IRQ 1 = FIR	0
INT_UART1	[23]	0 = IRQ 1 = FIR	0
INT_SPI0	[22]	0 = IRQ 1 = FIR	0
INT_SDI	[21]	0 = IRQ 1 = FIR	0
INT_DMA3	[20]	0 = IRQ 1 = FIR	0
INT_DMA2	[19]	0 = IRQ 1 = FIR	0
INT_DMA1	[18]	0 = IRQ 1 = FIR	0
INT_DMA0	[17]	0 = IRQ 1 = FIR	0
INT_LCD	[16]	0 = IRQ 1 = FIR	0
INT_UART2	[15]	0 = IRQ 1 = FIR	0
INT_TIMER4	[14]	0 = IRQ 1 = FIR	0
INT_TIMER3	[13]	0 = IRQ 1 = FIR	0
INT_TIMER2	[12]	0 = IRQ 1 = FIR	0
INT_TIMER1	[11]	0 = IRQ 1 = FIR	0
INT_TIMER0	[10]	0 = IRQ 1 = FIR	0
INT_WDT_AC97	[9]	0 = IRQ 1 = FIR	0
INT_TICK	[8]	0 = IRQ 1 = FIR	0
nBATT_FLT	[7]	0 = IRQ 1 = FIR	0
INT_CAM	[6]	0 = IRQ 1 = FIR	0
EINT8_23	[5]	0 = IRQ 1 = FIR	0
EINT4_7	[4]	0 = IRQ 1 = FIR	0
EINT3	[3]	0 = IRQ 1 = FIR	0
EINT2	[2]	0 = IRQ 1 = FIR	0
EINT1	[1]	0 = IRQ 1 = FIR	0
EINT0	[0]	0 = IRQ 1 = FIR	0

14.2.3 中断屏蔽寄存器

INTERRUPT MASK REGISTER (INTMSK)

该寄存器包括 32 位，每个都是和一个中断源相关。如果某位置 1，则CPU不会服务相应中断源的中断请求（注意SRCPND的相应位还是会被置 1）。如果屏蔽位为 0，中断请求可以被服务。

寄存器	地址	读写	描述	复位值
INTMSK	0x4A000008	R/W	决定哪个中断源被屏蔽。 0 = 中断服务无效 1 = 中断服务有效	0xFFFFFFFF

INTMSK	位	描述	初始值
INT_ADC	[31]	0 = 服务有效 1 = 服务屏蔽	0
INT_RTC	[30]	0 = 服务有效 1 = 服务屏蔽	0
INT_SPI1	[29]	0 = 服务有效 1 = 服务屏蔽	0
INT_UART0	[28]	0 = 服务有效 1 = 服务屏蔽	0
INT_IIC	[27]	0 = 服务有效 1 = 服务屏蔽	0
INT_USBH	[26]	0 = 服务有效 1 = 服务屏蔽	0
INT_USBD	[25]	0 = 服务有效 1 = 服务屏蔽	0
INT_NFCON	[24]	0 = 服务有效 1 = 服务屏蔽	0
INT_UART1	[23]	0 = 服务有效 1 = 服务屏蔽	0
INT_SPI0	[22]	0 = 服务有效 1 = 服务屏蔽	0
INT_SDI	[21]	0 = 服务有效 1 = 服务屏蔽	0
INT_DMA3	[20]	0 = 服务有效 1 = 服务屏蔽	0
INT_DMA2	[19]	0 = 服务有效 1 = 服务屏蔽	0
INT_DMA1	[18]	0 = 服务有效 1 = 服务屏蔽	0
INT_DMA0	[17]	0 = 服务有效 1 = 服务屏蔽	0
INT_LCD	[16]	0 = 服务有效 1 = 服务屏蔽	0
INT_UART2	[15]	0 = 服务有效 1 = 服务屏蔽	0
INT_TIMER4	[14]	0 = 服务有效 1 = 服务屏蔽	0
INT_TIMER3	[13]	0 = 服务有效 1 = 服务屏蔽	0
INT_TIMER2	[12]	0 = 服务有效 1 = 服务屏蔽	0
INT_TIMER1	[11]	0 = 服务有效 1 = 服务屏蔽	0
INT_TIMER0	[10]	0 = 服务有效 1 = 服务屏蔽	0
INT_WDT_AC97	[9]	0 = 服务有效 1 = 服务屏蔽	0
INT_TICK	[8]	0 = 服务有效 1 = 服务屏蔽	0
nBATT_FLT	[7]	0 = 服务有效 1 = 服务屏蔽	0
INT_CAM	[6]	0 = 服务有效 1 = 服务屏蔽	0
EINT8_23	[5]	0 = 服务有效 1 = 服务屏蔽	0
EINT4_7	[4]	0 = 服务有效 1 = 服务屏蔽	0
EINT3	[3]	0 = 服务有效 1 = 服务屏蔽	0
EINT2	[2]	0 = 服务有效 1 = 服务屏蔽	0
EINT1	[1]	0 = 服务有效 1 = 服务屏蔽	0
EINT0	[0]	0 = 服务有效 1 = 服务屏蔽	0

14.2.4 优先级寄存器

PRIORITY REGISTER (PRIORITY)

寄存器	地址	读写	描述	复位值
PRIORITY	0x4A00000C	R/W	IRQ优先级控制器	0x7F

INTMSK	位	描述	初始值
ARB_SEL6	[20:19]	仲裁器组 6 优先级顺序集 00 = REQ 0-1-2-3-4-5 01 = REQ 0-2-3-4-1-5 10 = REQ 0-3-4-1-2-5 11 = REQ 0-4-1-2-3-5	00
ARB_SEL5	[18:17]	仲裁器组 5 优先级顺序集 00 = REQ 1-2-3-4 01 = REQ 2-3-4-1 10 = REQ 3-4-1-2 11 = REQ 4-1-2-3	00
ARB_SEL4	[16:15]	仲裁器组 4 优先级顺序集 00 = REQ 0-1-2-3-4-5 01 = REQ 0-2-3-4-1-5 10 = REQ 0-3-4-1-2-5 11 = REQ 0-4-1-2-3-5	00
ARB_SEL3	[14:13]	仲裁器组 3 优先级顺序集 00 = REQ 0-1-2-3-4-5 01 = REQ 0-2-3-4-1-5 10 = REQ 0-3-4-1-2-5 11 = REQ 0-4-1-2-3-5	00
ARB_SEL2	[12:11]	仲裁器组 2 优先级顺序集 00 = REQ 0-1-2-3-4-5 01 = REQ 0-2-3-4-1-5 10 = REQ 0-3-4-1-2-5 11 = REQ 0-4-1-2-3-5	00
ARB_SEL1	[10:9]	仲裁器组 1 优先级顺序集 00 = REQ 0-1-2-3-4-5 01 = REQ 0-2-3-4-1-5 10 = REQ 0-3-4-1-2-5 11 = REQ 0-4-1-2-3-5	00
ARB_SEL0	[8:7]	仲裁器组 0 优先级顺序集 00 = REQ 1-2-3-4 01 = REQ 2-3-4-1 10 = REQ 3-4-1-2 11 = REQ 4-1-2-3	00
ARB_MODE6	[6]	仲裁器组 6 优先级翻转使能 0 = 优先级不翻转 1 = 优先级翻转使能	1
ARB_MODE5	[5]	仲裁器组 5 优先级翻转使能 0 = 优先级不翻转 1 = 优先级翻转使能	1
ARB_MODE4	[4]	仲裁器组 4 优先级翻转使能 0 = 优先级不翻转 1 = 优先级翻转使能	1
ARB_MODE3	[3]	仲裁器组 3 优先级翻转使能 0 = 优先级不翻转 1 = 优先级翻转使能	1
ARB_MODE2	[2]	仲裁器组 2 优先级翻转使能 0 = 优先级不翻转 1 = 优先级翻转使能	1
ARB_MODE1	[1]	仲裁器组 1 优先级翻转使能 0 = 优先级不翻转 1 = 优先级翻转使能	1
ARB_MODE0	[0]	仲裁器组 0 优先级翻转使能 0 = 优先级不翻转 1 = 优先级翻转使能	1

14.2.5 中断未决寄存器

INTERRUPT PENDING REGISTER (INTPND)

中断未决寄存器的 32 位显示是否相应的中断请求有最高优先级，其中断请求未屏蔽且在等待中断服务。因为INTPND寄存器位于优先级逻辑之后，仅 1 位可以被置 1，且中断请求生成对CPU的IRQ。在对于IRQ的中断服务程序中，我们可以读取寄存器决定那个中断源被服务。

寄存器	地址	读写	描述	复位值
INTPND	0x4A000010	R/W	指出中断请求的状态 0: 中断还没有被请求 1: 中断源已经申请中断请求	0x00000000

INTPND	位	描述		初始值
INT_ADC	[31]	0 = 未请求	1 = 已请求	0
INT_RTC	[30]	0 = 未请求	1 = 已请求	0
INT_SPI1	[29]	0 = 未请求	1 = 已请求	0
INT_UART0	[28]	0 = 未请求	1 = 已请求	0
INT_IIC	[27]	0 = 未请求	1 = 已请求	0
INT_USBH	[26]	0 = 未请求	1 = 已请求	0
INT_USBD	[25]	0 = 未请求	1 = 已请求	0
INT_NFCON	[24]	0 = 未请求	1 = 已请求	0
INT_UART1	[23]	0 = 未请求	1 = 已请求	0
INT_SPI0	[22]	0 = 未请求	1 = 已请求	0
INT_SDI	[21]	0 = 未请求	1 = 已请求	0
INT_DMA3	[20]	0 = 未请求	1 = 已请求	0
INT_DMA2	[19]	0 = 未请求	1 = 已请求	0
INT_DMA1	[18]	0 = 未请求	1 = 已请求	0
INT_DMA0	[17]	0 = 未请求	1 = 已请求	0
INT_LCD	[16]	0 = 未请求	1 = 已请求	0
INT_UART2	[15]	0 = 未请求	1 = 已请求	0
INT_TIMER4	[14]	0 = 未请求	1 = 已请求	0
INT_TIMER3	[13]	0 = 未请求	1 = 已请求	0
INT_TIMER2	[12]	0 = 未请求	1 = 已请求	0
INT_TIMER1	[11]	0 = 未请求	1 = 已请求	0
INT_TIMER0	[10]	0 = 未请求	1 = 已请求	0
INT_WDT_AC97	[9]	0 = 未请求	1 = 已请求	0
INT_TICK	[8]	0 = 未请求	1 = 已请求	0
nBATT_FLT	[7]	0 = 未请求	1 = 已请求	0
INT_CAM	[6]	0 = 未请求	1 = 已请求	0
EINT8_23	[5]	0 = 未请求	1 = 已请求	0
EINT4_7	[4]	0 = 未请求	1 = 已请求	0
EINT3	[3]	0 = 未请求	1 = 已请求	0
EINT2	[2]	0 = 未请求	1 = 已请求	0
EINT1	[1]	0 = 未请求	1 = 已请求	0
EINT0	[0]	0 = 未请求	1 = 已请求	0

14.2.6 中断偏移寄存器

INTERRUPT OFFSET REGISTER (INTOFFSET)

中断偏移寄存器中的值显示了那个IRQ模式的中断请求在INTPND寄存器中，该位可以通过清除SRCPND和INTPND寄存器被自动清除。

寄存器	地址	读写	描述	复位值
INTOFFSET	0x4A000014	R	指出IRQ中断请求源	0x00000000

INTOFFSET	偏移量	INTOFFSET	偏移量
INT_ADC	31	INT_UART2	15
INT_RTC	30	INT_TIMER4	14
INT_SPI1	29	INT_TIMER3	13
INT_UART0	28	INT_TIMER2	12
INT_IIC	27	INT_TIMER1	11
INT_USBH	26	INT_TIMER0	10
INT_USBD	25	INT_WDT_AC97	9
INT_NFCON	24	INT_TICK	8
INT_UART1	23	nBATT_FLT	7
INT_SPI0	22	INT_CAM	6
INT_SDI	21	EINT8_23	5
INT_DMA3	20	EINT4_7	4
INT_DMA2	19	EINT3	3
INT_DMA1	18	EINT2	2
INT_DMA0	17	EINT1	1
INT_LCD	16	EINT0	0

14.2.7 子中断源未决寄存器

SUB SOURCE PENDING REGISTER (SUBSRCPND)

寄存器	地址	读写	描述	复位值
SUBSRCPND	0x4A000018	R/W	指出中断请求的状态 0 = 中断未请求 1 = 中断源已经申请中断	0x00000000

SUBSRCPND	位	描述		初始值
保留	[31:15]	0 = 未请求	1 = 已请求	0
INT_AC97	[14]	0 = 未请求	1 = 已请求	0
INT_WDT	[13]	0 = 未请求	1 = 已请求	0
INT_CAM_P	[12]	0 = 未请求	1 = 已请求	0
INT_CAM_C	[11]	0 = 未请求	1 = 已请求	0
INT_ADC_S	[10]	0 = 未请求	1 = 已请求	0
INT_TC	[9]	0 = 未请求	1 = 已请求	0
INT_ERR2	[8]	0 = 未请求	1 = 已请求	0
INT_TXD2	[7]	0 = 未请求	1 = 已请求	0
INT_RXD2	[6]	0 = 未请求	1 = 已请求	0
INT_ERR1	[5]	0 = 未请求	1 = 已请求	0
INT_TXD1	[4]	0 = 未请求	1 = 已请求	0
INT_RXD1	[3]	0 = 未请求	1 = 已请求	0
INT_ERR0	[2]	0 = 未请求	1 = 已请求	0
INT_TXD0	[1]	0 = 未请求	1 = 已请求	0
INT_RXD0	[0]	0 = 未请求	1 = 已请求	0

4.2.8 子中断屏蔽寄存器

INTERRUPT SUB MASK REGISTER (INTSUBMSK)

该寄存器有 15 位，每位和一个中断源相关。如果某位置 1，则 CPU 不会服务相应中断源的中断请求（注意 SRCPND 的相应位还是会被置 1）。如果屏蔽位为 0，中断请求可以被服务。

寄存器	地址	读写	描述	复位值
SUBINTMSK	0x4A00001C	R/W	决定哪个中断源被屏蔽。 0 = 中断服务无效 1 = 中断服务有效	0xFFFF

SUBINTMSK	位	描述	初始值
保留	[31:15]	-	0
INT_AC97	[14]	0 = 服务有用 1 = 服务屏蔽	1
INT_WDT	[13]	0 = 服务有用 1 = 服务屏蔽	1
INT_CAM_P	[12]	0 = 服务有用 1 = 服务屏蔽	1
INT_CAM_C	[11]	0 = 服务有用 1 = 服务屏蔽	1
INT_ADC_S	[10]	0 = 服务有用 1 = 服务屏蔽	1
INT_TC	[9]	0 = 服务有用 1 = 服务屏蔽	1
INT_ERR2	[8]	0 = 服务有用 1 = 服务屏蔽	1
INT_TXD2	[7]	0 = 服务有用 1 = 服务屏蔽	1
INT_RXD2	[6]	0 = 服务有用 1 = 服务屏蔽	1
INT_ERR1	[5]	0 = 服务有用 1 = 服务屏蔽	1
INT_TXD1	[4]	0 = 服务有用 1 = 服务屏蔽	1
INT_RXD1	[3]	0 = 服务有用 1 = 服务屏蔽	1
INT_ERR0	[2]	0 = 服务有用 1 = 服务屏蔽	1
INT_TXD0	[1]	0 = 服务有用 1 = 服务屏蔽	1
INT_RXD0	[0]	0 = 服务有用 1 = 服务屏蔽	1

第十六章 ADC 和触摸屏接口

16.1 概述

10 位 CMOS 的 ADC（模数转换器）是有 8 通道模拟输入的循环类型设备。其转换模拟输入信号到 10 位的数字编码，最大的转换率是在 2.5MHz 转换时钟下达到 500KSPS。AD 转换器支持片上采样和保持功能及掉电模式。

触摸屏接口可以控制或选择触摸屏触点用于 XY 坐标的转换。触摸屏接口包括触摸触点控制逻辑和有中断产生逻辑的 ADC 接口逻辑。

16.2 特点

- 分辨率：10 位
- 微分线性误差： $\pm 1.0\text{LSB}$
- 积分线性误差： $\pm 2.0\text{LSB}$
- 最大转换速率：500KSPS
- 低功耗
- 供电电压：3.3V
- 输入模拟电压范围：0~3.3V
- 片上采样保持功能
- 普通转换模式
- 分离的 XY 坐标转换摸
- 自动连续 XY 坐标转换模式
- 等待中断模式

16.3 ADC 及触摸屏接口操作

模块图

如图 16-1 所示 AD 转换器和触摸屏接口的功能模块图。注意 AD 转换器设备是一个循环类型。

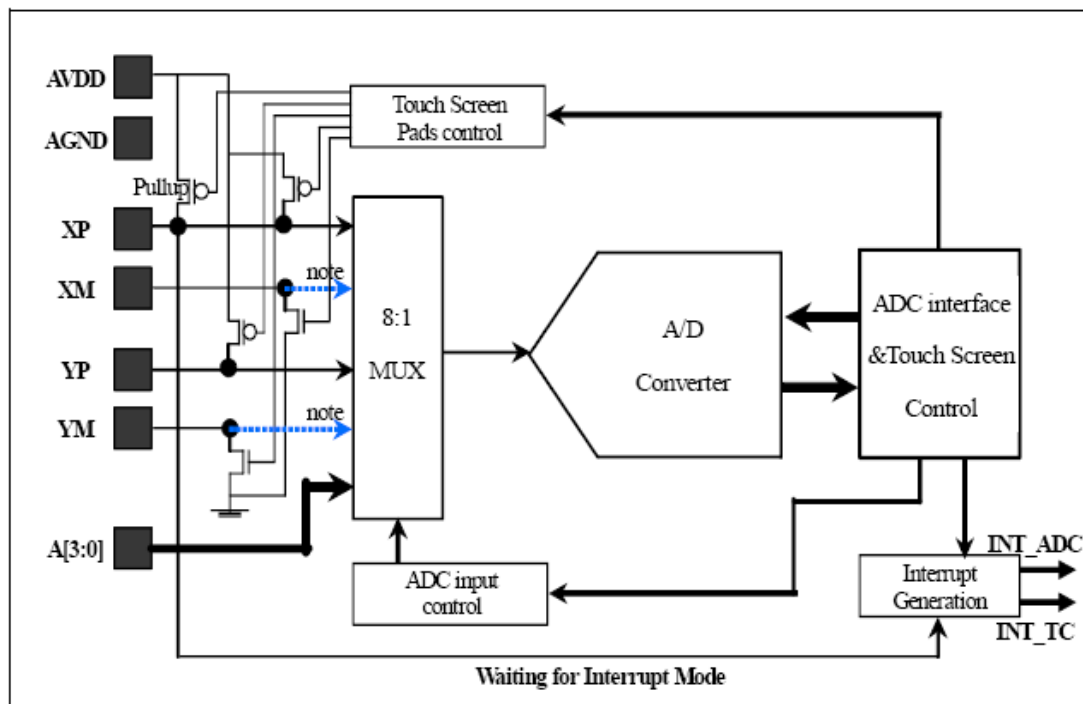


Figure 16-1. ADC and Touch Screen Interface Functional Block Diagram

注意 (图标→)

当触摸屏接口 使用时, XM或PM应该接触摸屏接口的地。

当触摸屏设备不使用时, XM或PM应该连接模拟输入信号作为普通ADC转换用。

16.4 功能描述

16.4.1 AD 转换时间

当GCLK频率为 50MHz和预分频器(预定标器)值为 49, 总共 10 位转换时间如下:

$AD\text{转换器频率} = 50\text{MHz}/(49+1) = 1\text{MHz}$

$\text{转换时间} = 1/(1\text{MHz} / 5\text{cycles}) = 1/200\text{KHz} = 5\text{us}$

注: AD转换器设计在最大 2.5MHz时钟下工作, 所以转换率最高达到 500KSPS。

16.4.2 触摸屏接口模式

(1) 正常转换模式

单个转换模式可能多数是使用在通用目的的ADC转换。该模式可以通过设置 ADCCON (ADC控制寄存器) 来初始化并且完成对ADCDAT0 的读写操作 (ADC数据寄存器 0)。

(2) 分离XY坐标转换模式

触摸屏控制器可以在两种转换模式中的一种模式下操作。分离的XY坐标转换模式由以下方法操作。X坐标模式写X坐标转换数据到ADCDAT0，触摸屏接口产生中断源到中断控制器。Y坐标模式写Y坐标转换数据到ADCDAT1，触摸屏接口产生中断源到中断控制器。

	XP	XM	YP	YM
X Position Conversion	External Voltage	GND	AIN[5]	Hi-Z
Y Position Conversion	AIN[7]	Hi-Z	External Voltage	GND

(3) 自动（连续）XY坐标转换模式

自动（连续）XY坐标转换模式是如下操作，触摸屏控制器连续的转换触摸X坐标和Y坐标。在触摸控制器写X测量数据到ADCDAT0 且写Y测量数据到ADCDAT1 后，触摸屏接口产生中断源到自动坐标转换模式下的中断控制器。

	XP	XM	YP	YM
X Position Conversion	External Voltage	GND	AIN[5]	Hi-Z
Y Position Conversion	AIN[7]	Hi-Z	External Voltage	GND

(4) 等待中断模式

当光标按下，触摸屏控制器产生中断信号（INT_TC）。触摸屏控制器的等待中断模式必须设定为触摸屏接口中触点的状态（XP、XM、YP、YM）。

在触摸屏控制器产生中断信号（INT_TC），等待中断模式必须被清除。（XY_PST设置到无操作模式）

	XP	XM	YP	YM
Waiting for Interrupt Mode	Pull up	Hi-Z	AIN[5]	GND

备用模式：

备用模式在ADCCON[2]置 1 时激活。在此模式下，AD转换操作停止，ADCDAT0 和ADCDAT1 寄存器保留先前的转换数据。

16.4.3 编程注意

- (1) AD转换的数据可以通过中断或查询的方式来访问。使用中断方式整个转换时间（从AD转换器开始到转换数据读取）可能会因为中断服务程序的返回时间和数据访问时间而延长。使用查询方式，通过查看ADCCON[15]位（转换标志结束位），ADCDAT寄存器的读取时间可以确定。
- (2) 提供另外的开启AD转换的方法。在ADCCON[1]置 1（AD转换开始读取模式），只要转换数据被读取，AD转换同时开始。

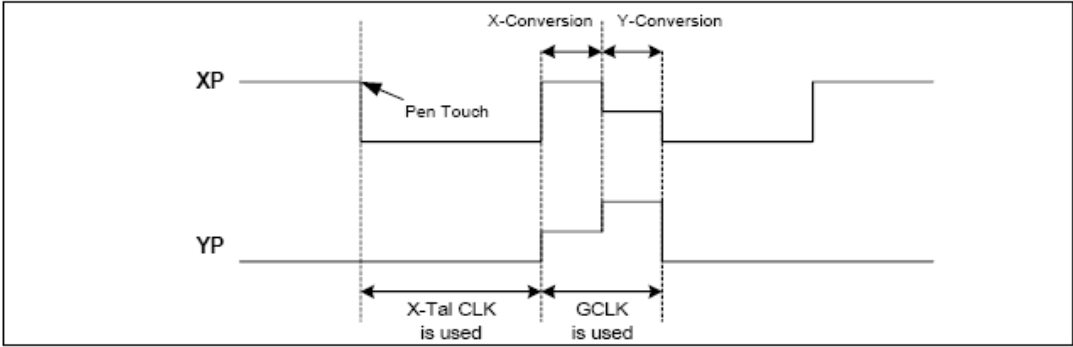


Figure 16-2 ADC and Touch Screen Operation signal

16.5 ADC 及触摸屏接口特殊寄存器

- (1) ADC控制寄存器 (ADCCON)
- (2) ADC触摸屏控制寄存器 (ADCTSC)
- (3) ADC开始延时寄存器 (ADCDLY)
- (4) ADC转换数据寄存器 0 (ADCDAT0)
- (5) ADC转换数据寄存器 1 (ADCDAT1)
- (6) ADC触摸屏指针上下中断检测寄存器 (ADCUPDN)

16.5.1 ADC 控制寄存器

ADC CONTROL REGISTER (ADCCON)

寄存器	地址	读写	描述	复位值
ADCCON	0x58000000	R/W	ADC控制寄存器	0x3FC4

ADCCON	位	描述	初始值
ECFLG	[15]	转换标志结束 (只读) 0: AD转换在过程中 1: AD转换结束	0
PRSCEN	[14]	AD转换器预分频器 (预定标器) 使能 0: 无效 1: 有效	0
PRSCVL	[13:6]	AD转换器预分频器 (预定标器) 值, 数值: 0~255。 注意: ADC频率应该设置至少小于PCLK的 1/5。	0xFF
SEL_MUX	[5:3]	模拟输入通道选择。 000: AIN0 001: AIN1 010: AIN2 011: AIN3 100: YM 101: YP 110: XM 111: XP	0
STDBM	[2]	备用操作模式选择 0: 普通操作模式 1: 备用模式	1
READ_START	[1]	AD转换通过读取开始 0: 通过读取操作开始无效 1: 通过读取操作开始有效	0
READ_START	[0]	AD转换开始有效。 如果READ_START有效, 该值无效。 0: 无操作 1: AD转换开始且该位在开始后清零	0

注: 当触摸屏触点 (YM、YP、XM、XP) 无效, 这些引脚应该用于作为ADC的模拟输入引脚 (AIN4、AIN5、AIN6、AIN7)。

16.5.2 ADC 触摸屏控制寄存器

ADC TOUCH SCREEN CONTROL REGISTER (ADCTSC)

寄存器	地址	读写	描述	复位值
ADCTSC	0x58000004	R/W	ADC触摸屏控制寄存器	0x58

ADCTSC	位	描述	初始值
UD_SEN	[8]	检测光标上下状态 0: 检测光标按下中断信号 1: 检测光标抬起中断信号	0
YM_SEN	[7]	YM开关使能 0: YM输出驱动无效(Hi-z) 1: YM输出驱动有效(GND)	0
YP_SEN	[6]	YP开关使能 0: YP输出驱动有效(Ext -vol) 1: YP输出驱动无效(AIN5)	1
XM_SEN	[5]	XM开关使能 0: XM输出驱动无效(Hi-z) 1: XM输出驱动有效(GND)	0
XP_SEN	[4]	XP开关使能 0: XP输出驱动有效(Ext -vol) 1: XP输出驱动无效(AIN7)	1
PULL_UP	[3]	上拉开关使能 0: XP上拉有效 1: XP上拉无效	1
AUTO_PST	[2]	自动连续转换X坐标和Y坐标 0: 普通ADC转换 1: 自动连续测量X坐标和Y坐标	0
XY_PST	[1:0]	手动测量X坐标和Y坐标 00: 无操作模式 01: X坐标测量 10: Y坐标测量 11: 等待中断模式	0

注:

- (1) 当等待触摸屏中断时, XP_SEN位(XP输出无效)应该置1且PULL_UP(XP上拉使能)位应该置0。
- (2) 仅在自动连续XY坐标转换中, AUTO_PST位应该置1
- (3) 当睡眠模式下为了避免泄漏电流, XP、YP应该和地断开。因为XP和YP在睡眠模式下保持高电平。

在XY坐标转换中的触摸屏引脚条件

	XP	XM	YP	YM	ADC通道选择
X坐标(0100)	Vref	GND	Hi-Z	Hi-Z	YP
Y坐标(0001)	Hi-Z	Hi-Z	Vref	GND	XP

16.5.3 ADC 开始延时寄存器

ADC START DELAY REGISTER (ADCDLY)

寄存器	地址	读写	描述	复位值
ADCDLY	0x58000008	R/W	ADC开始延时寄存器	0x58

ADCDLY	位	描述	初始值
DELAY	[15:0]	(1) 普通转换模式, XY坐标模式, 自动坐标模式。 ->AD转换开始延迟值。 (2) 等待中断模式。 当光标按下出现在睡眠模式时, 产生一个用于退出睡眠模式的唤醒信号, 有几个毫秒的时间间隔。注: 不要用 0 值	00ff

注: 在ADC转换前, 触摸屏使用晶振时钟 (3.68MHz), 在AD转换中使用GCLK (最大50MHz)。

16.5.4 ADC 转换数据寄存器 0

ADC CONVERSION DATA REGISTER (ADCDAT0)

寄存器	地址	读写	描述	复位值
ADCDAT0	0x5800000C	R/W	ADC转换数据寄存器	-

ADCDAT0	位	描述	初始值
UPDOWN	[15]	对于等待中断模式的光标按下或提起状态 0: 光标按下状态 1: 光标提起状态	-
AUTO_PST	[14]	X坐标和Y坐标的自动连续转换 0: 普通ADC转换 1: X坐标和Y坐标的连续测量	-
XY_PST	[13:12]	X坐标和Y坐标的手动测量 00: 无操作模式 01: X坐标测量 10: Y坐标测量 11: 等待中断模式	-
保留	[11:10]	保留	-
XPDATA	[9:0]	X坐标转换数据值 (包括普通ADC转换数据值) 数据值: 0~3FF	-

16.5.5 ADC 转换数据寄存器 1

ADC CONVERSION DATA REGISTER (ADCDAT1)

寄存器	地址	读写	描述	复位值
ADCDAT1	0x58000010	R/W	ADC转换数据寄存器	-

ADCDAT1	位	描述	初始值
UPDOWN	[15]	对于等待中断模式的光标按下或提起状态 0: 光标按下状态 1: 光标提起状态	-
AUTO_PST	[14]	X坐标和Y坐标的自动连续转换 0: 普通ADC转换 1: X坐标和Y坐标的连续测量	-
XY_PST	[13:12]	X坐标和Y坐标的手动测量 00: 无操作模式 01: X坐标测量 10: Y坐标测量 11: 等待中断模式	-
保留	[11:10]	保留	-
YPDATA	[9:0]	Y坐标转换数据值 (包括普通ADC转换数据值) 数据值: 0~3FF	-

16.5.6 ADC 触摸屏指针上下中断检测寄存器

ADC TOUCH SCREEN UP-DOWN INT CHECK REGISTER (ADCUPDN)

寄存器	地址	读写	描述	复位值
ADCUPDN	0x58000014	R/W	触摸屏指针上下中断检测寄存器	0x0

ADCUPDN	位	描述	初始值
TSC_UP	[1]	光标提起中断 0: 无光标提起状态 1: 出现光标提起中断	0
TSC_DN	[0]	光标按下中断 0: 无光标按下状态 1: 出现光标按下中断	0

第十七章 实时时钟

17.1 概述

实时时钟 (RTC) 单元在系统电源关闭的情况下可以在备用电池下工作。RTC 可以使用 STRB/LDRB ARM 操作传输二进制码十进制数的 8 位数据给 CPU。数据包括秒、分钟、小时、日期、天、月、年的时间信息。RTC 单元可以在 32.768KHz 的外部晶振下工作，可以执行报警功能。

17.2 特点

- BCD 数：秒、分钟、小时、日期、日、月、年
- 闰年生成器
- 报警功能：报警中断或从掉电模式中唤醒
- 已经解决 2000 年问题
- 独立电源引脚 (RTCVDD)
- 支持对于实时内核时间节拍的毫秒节拍时间中断

17.3 实时时钟操作

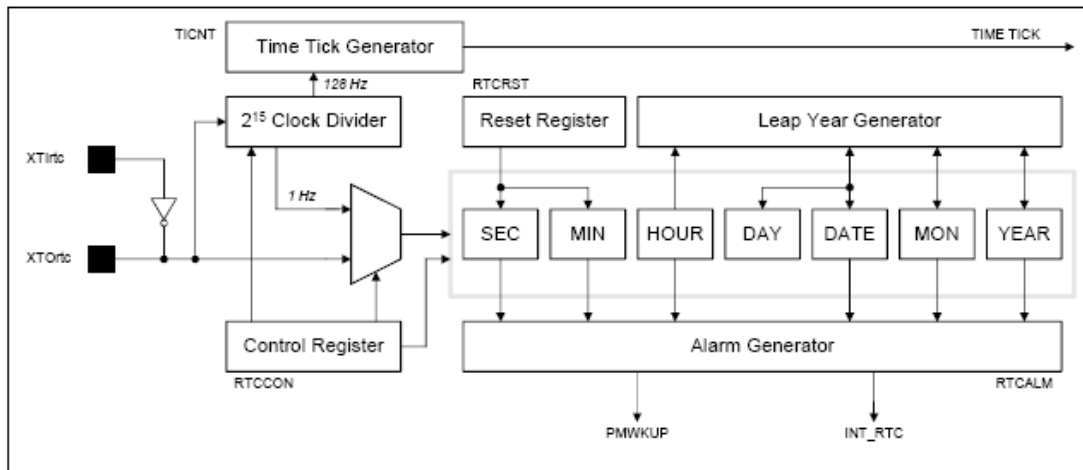


Figure 17-1. Real Time Clock Block Diagram

17.3.1 闰年发生器

闰年发生器可以基于 BCDDATE、BCDMON、BCDYEAR 的数据，从 28、29、30、31 中确定每个月的最后一天。该模块在确定某月最后一天的时候会考虑闰年的因素。一个 8 位的计数器仅能代表两个 BCD 数字，所以它不能确定是否是 00 年（该年的最后两个数字是 00）。例如，它不能区别 1900 年和 2000 年。为了解决这个问题，S3C2440A 的 RTC 模块有一个硬件逻辑来支持在 2000 年的闰年。记录 1900 年不是闰年而 2000 年是闰年。因此两个数字 00 在 S3C2440A 中记录的是 2000 年而不是 1900 年。

17.3.2 读写寄存器

为了写RTC模块中的BCD寄存器，RTCCON寄存器的位0必须置1。为了显示秒分小时星期日月年，CPU应该分别读取在RTC模块中的BCDSEC，BCDMIN，BCD HOUR，BCDDAY，BCDDATE，BCDMON，和BCDYEAR。但是，因为多寄存器读取可能存在一秒的误差。例如，当用户读BCDYEAR和BCDMON，结构假定是2059年12月31日23点59分。当用户读BCDSEC寄存器，值的范围是从1到59（秒）就没有问题，但是如果值是0，年月日就变成了2060年1月1日0时0分因为有刚才提到的1秒误差。在这种情况下如果BCDSEC为0，用户应该重读BCDYEAR到BCDSEC。

17.3.3 备用电池操作

RTC 逻辑可以由备用电池驱动，其通过 RCTVDD 引脚给 RTC 模块提供电源，即使系统电源关闭。当系统关闭时，CPU 和 RTC 模块的接口是封闭的，备用电池仅驱动振荡电路和 BCD 计数器以最小化电源消耗。

17.3.4 报警功能

RTC 在掉电模式或正常操作模式下的特定时间会发出报警信号。在正常操作模式下报警中断（INT_RTC）被激活。在掉电模式下，电源管理唤醒信号（PMWKUP）也如 INT_RTC 一样被激活。RTC 报警寄存器（RTCALM）决定了报警的使能状态和报警时间设定的条件。

17.3.5 节拍时间中断

RTC 节拍时间是用于中断请求。TICNT 寄存器有一个中断使能位和对于中断的计数器值。当节拍时间中断出现时，计数器的值为0。中断周期如下：

$$\text{Period} = (n+1) / 128 \text{ second}$$

n: 节拍计数器值 (1~127)

RTC 节拍时间可以用于实时操作系统内核时间节拍。如果时间节拍由 RTC 时间节拍生成，与实时操作系统功能相关的时间就会和实时同步。

17.3.6 32.768KHz 晶振连接举例

如图 17-2 所示的 RTC 单元在 32.768KHz 频率下振荡的电路

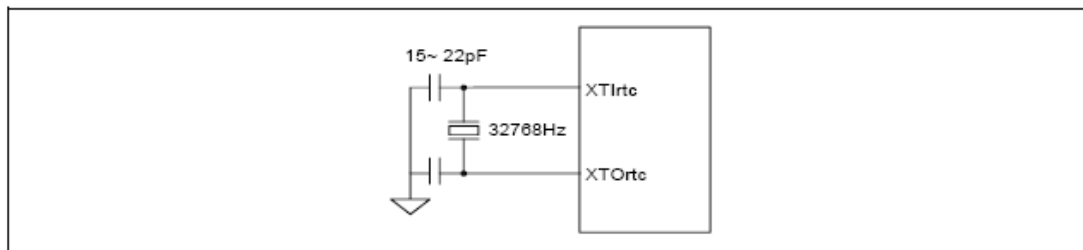


Figure 17-2. Main Oscillator Circuit Example

17.4 实时时钟特殊寄存器

- (1) 实时时钟控制寄存器 (RTCCON)
- (2) 节拍时间计数寄存器 (TICNT)
- (3) RTC 报警控制寄存器 (RTCALM)
- (4) 报警秒数据寄存器 (ALMSEC)
- (5) 报警分钟数据寄存器 (ALMMIN)
- (6) 报警小时数据寄存器 (ALMHOUR)
- (7) 报警日期数据寄存器 (ALMDATE)
- (8) 报警月数据寄存器 (ALMMON)
- (9) 报警年数据寄存器 (ALMYEAR)
- (10) BCD 秒寄存器 (BCDSEC)
- (11) BCD 分寄存器 (BCDMIN)
- (12) BCD 小时寄存器 (BCDHOURL)
- (13) BCD 日期寄存器 (BCDDATE)
- (14) BCD 日寄存器 (BCDDAY)
- (15) BCD 月寄存器 (BCDMON)
- (16) BCD 年寄存器 (BCDYEAR)

17.4.1 实时时钟控制寄存器

REAL TIME CLOCK CONTROL REGISTER (RTCCON)

RTCCON 寄存器包括 4 位, 例如 RTCEN, 控制 BCD 寄存器的读写使能, CLKSEL, CNTSEL 和 CLKRST 用于测试。

RCTEN 位可以控制 CPU 和 RTC 之间的所有接口, 所以在系统复位后在使能数据读写的 RTC 控制代码中该位应该置 1。在关闭电源前, 该位应该被清零以避免无意的写 RTC 寄存器。

寄存器	地址	读写	描述	复位值
RTCCON	0x57000040(L) 0x57000043(B)	R/W (字节)	RTC控制寄存器	0x0

RTCCON	位	描述	初始值
CLKRST	[3]	RTC时钟计数器复位 0 = No reset, 1 = Reset	0
CNTSEL		BCD计数器选择 0 = Merge BCD counters 1 = Reserved (单独BCD计数器)	0
CLKSEL	[1]	BCD时钟选择 0 = XTAL 1/2 ¹⁵ 分频时钟 1 = Reserved (XTAL clock only for test)	0
RTCEN	[0]	RTC控制使能 0 = 无效 1 = 有效 注: 仅BCD时间计数器和读操作可以执行。	0

注: (1) 所有的 RTC 寄存器都可以使用 STRB 和 LDRB 指令或字符型指针针对每个字节单元访问。

(2) (L): 小端。(B): 大端。

17.4.2 节拍时间计数寄存器

TICK TIME COUNT REGISTER (TICNT)

寄存器	地址	读写	描述	复位值
TICNT	0x57000044(L) 0x57000047(B)	R/W (字节)	节拍时间计数寄存器	0x0

TICNT	位	描述	初始值
TICK INT ENABLE	[7]	节拍时间中断使能 0 = 无效, 1 = 有效	0
TICK TIME COUNT	[6:0]	节拍时间计数值 (1~127). 该计数器的值在内部减少, 工作期间用户不能读取该计数器值。	000000

17.4.3 RTC 报警控制寄存器

RTC ALARM CONTROL REGISTER (RTCALM)

RTCALM 寄存器决定了报警使能和报警时间。在掉电模式下 RTCALM 寄存器通过 INT_RTC 和 PMWKUP 产生报警信号, 在正常操作模式下仅通过 INT_RTC 产生。

寄存器	地址	读写	描述	复位值
RTCALM	0x57000050(L) 0x57000053(B)	R/W (字节)	RTC报警控制寄存器	0x0

RTCALM	位	描述	初始值
Reserved	[7]		0
ALMEN	[6]	全局报警使能 0 = 无效, 1 = 有效	0
YEAREN	[5]	年报警使能 0 = 无效, 1 = 有效	0
MONREN	[4]	月报警使能 0 = 无效, 1 = 有效	0
DATEEN	[3]	日期报警使能 0 = 无效, 1 = 有效	0
HOUREN	[2]	小时报警使能 0 = 无效, 1 = 有效	0
MINEN	[1]	分钟报警使能 0 = 无效, 1 = 有效	0
SECEN	[0]	秒报警使能 0 = 无效, 1 = 有效	0

17.4.4 报警秒数据寄存器

ALARM SECOND DATA REGISTER (ALMSEC)

寄存器	地址	读写	描述	复位值
ALMSEC	0x57000054(L) 0x57000057(B)	R/W (字节)	报警秒数据寄存器	0x0

ALMSEC	位	描述	初始值
保留	[7]		0
SECDATA	[6:4]	对于报警秒的BCD值 0~5	000
	[3:0]	0~9	0000

17.4.5 报警分钟数据寄存器

ALARM MIN DATA REGISTER (ALMMIN)

寄存器	地址	读写	描述	复位值
ALMMIN	0x57000058(L) 0x5700005B(B)	R/W (字节)	报警分钟数据寄存器	0x0

ALMMIN	位	描述	初始值
保留	[7]		0
MINDATA	[6:4]	对于报警分的BCD值 0~5	000
	[3:0]	0~9	0000

17.4.6 报警小时数据寄存器

ALARM HOUR DATA REGISTER (ALMHOUR)

寄存器	地址	读写	描述	复位值
ALMHOUR	0x5700005C(L) 0x5700005F(B)	R/W (字节)	报警小时数据寄存器	0x0

ALMHOUR	位	描述	初始值
保留	[7:6]		00
HOURLDATA	[5:4]	对于报警分的BCD值 0~2	00
	[3:0]	0~9	0000

17.4.7 报警日期数据寄存器

ALARM DATE DATA REGISTER (ALMDATE)

寄存器	地址	读写	描述	复位值
ALMDATE	0x57000060(L) 0x57000063(B)	R/W (字节)	报警日期数据寄存器	0x01

ALMDATE	位	描述	初始值
保留	[7:6]		00
DATEDATA	[5:4]	对于报警日期的BCD值, 从 0 到 28, 29, 30, 31 0~3	00
	[3:0]	0~9	0001

17.4.8 报警月数据寄存器

ALARM MON DATA REGISTER (ALMMON)

寄存器	地址	读写	描述	复位值
ALMMON	0x57000064(L) 0x57000067(B)	R/W (字节)	报警月数据寄存器	0x01

ALMMON	位	描述	初始值
保留	[7:5]		00
MONDATA	[4]	对于报警月的BCD值, 0~1	00
	[3:0]	0~9	0001

17.4.9 报警年数据寄存器

ALARM YEAR DATA REGISTER (ALMYEAR)

寄存器	地址	读写	描述	复位值
ALMYEAR	0x57000068(L) 0x5700006B(B)	R/W (字节)	报警年数据寄存器	0x01

ALMYEAR	位	描述	初始值
YEARDATA	[7:0]	年的BCD值 00 ~ 99	0x0

17.4.10 BCD 秒寄存器

BCD SECOND REGISTER (BCDSEC)

寄存器	地址	读写	描述	复位值
BCDSEC	0x57000070(L) 0x57000073(B)	R/W (字节)	BCD秒寄存器	-

BCDSEC	位	描述	初始值
SECDATA	[6:4]	对于秒的BCD值, 0~5	-
	[3:0]	0~9	-

17.4.11 BCD 分钟寄存器

BCD MINUTE REGISTER (BCDMIN)

寄存器	地址	读写	描述	复位值
BCDMIN	0x57000074(L) 0x57000077(B)	R/W (字节)	BCD分钟寄存器	-

BCDMIN	位	描述	初始值
MINDATA	[6:4]	对于分钟的BCD值, 0~5	-
	[3:0]	0~9	-

17.4.12 BCD 小时寄存器

BCD HOUR REGISTER (BCD HOUR)

寄存器	地址	读写	描述	复位值
BCD HOUR	0x57000078(L) 0x5700007B(B)	R/W (字节)	BCD小时寄存器	-

BCD HOUR	位	描述	初始值
保留	[7:6]	-	-
HOURDATA	[5:4]	对于小时的BCD值, 0~5	-
	[3:0]	0~9	-

17.4.13 BCD 日期寄存器

BCD DATE REGISTER (BCDDATE)

寄存器	地址	读写	描述	复位值
BCDDATE	0x5700007C(L) 0x5700007F(B)	R/W (字节)	BCD日期寄存器	-

BCDDATE	位	描述	初始值
保留	[7:6]	-	-
DATEDATA	[5:4]	对于小时的BCD值, 0~3	-
	[3:0]	0~9	-

17.4.14 BCD 日寄存器

BCD DAY REGISTER (BCDDAY)

寄存器	地址	读写	描述	复位值
BCDDAY	0x57000080(L) 0x57000083(B)	R/W (字节)	BCD星期寄存器	-

BCDDAY	位	描述	初始值
保留	[7:3]	-	-
DAYDATA	[2:0]	对于星期的BCD值, 1~7	-

17.4.15 BCD 月寄存器

BCD MONTH REGISTER (BCDMON)

寄存器	地址	读写	描述	复位值
BCDMON	0x57000084(L) 0x57000087(B)	R/W (字节)	BCD月寄存器	-

BCDMON	位	描述	初始值
保留	[7:5]	-	-
MONDATA	[4]	对于月的BCD值, 0~1	-
	[3:0]	0~9	

17.4.16 BCD 年寄存器

BCD YEAR REGISTER (BCDYEAR)

寄存器	地址	读写	描述	复位值
BCDMON	0x57000088(L) 0x5700008B(B)	R/W (字节)	BCD年寄存器	-

BCDDAY	位	描述	初始值
YEARDATA	[7:0]	对于年的BCD值, 00~99	-

第十八章 看门狗定时器

18.1 概述

S3c2440A 看门狗定时器是用于当例如噪音或系统错误引起的故障时，恢复控制器操作。其作为一个普通的 16 位时间间隔定时器来请求中断服务。看门狗定时器产生一个 128PCLK 周期的复位信号。

18.2 特点

- 有中断请求的普通间隔定时器模式
- 当定时器计数器值位 0（超时）对于 128 个 PCLK 周期的内部复位信号被激活。

18.3 看门狗定时器操作

如图 18-1 所示的看门狗定时器功能模块图。看门狗定时器使用 PCLK 作为其源时钟。PCLK 频率被于预分频产生相应的看门狗定时器时钟，得到的频率再次被分频。

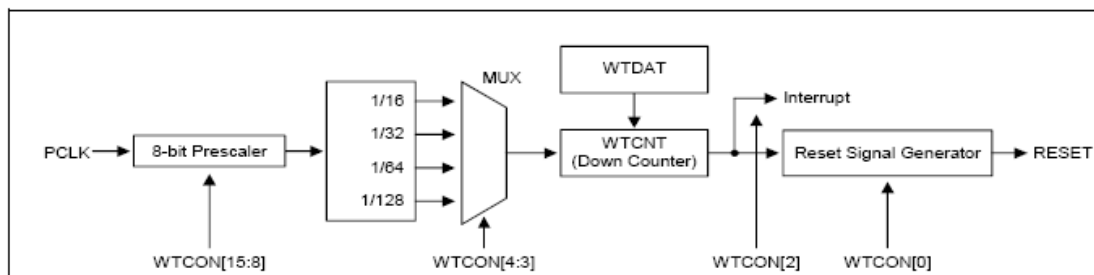


Figure 18-1. Watchdog Timer Block Diagram

预定标器的值和频率除数因子由看门狗定时器控制寄存器来定义，有效的预定标器的值范围从 0 到 $2^{18}-1$ 。频率除数因此可以选择 16、32、64 或 128。

可以使用以下公式来计算看门狗定时器的时钟频率和每个定时器时钟周期的持续时间：

$$t_{\text{watchdog}} = 1 / [\text{PCLK} / (\text{Prescaler value} + 1) / \text{Division_factor}]$$

18.3.1 WTDAT & WTCNT

一旦看门狗定时器使能，看门狗定时器数据寄存器（WTDAT）的值不能自动重载到定时器计数器。因此，在看门狗定时器开始工作前，一个初始值必须写如看门狗定时器计数寄存器（WTCNT）。

18.3.2 调试环境的考虑

当 s3c2440A 在调试模式下使用 IEC，不必操作看门狗定时器。

看门狗定时器可以从 CPU 内核信号（DBGACK 信号）来判断是否当前在调试模式。一旦 DBGACK 信号有效，当看门狗定时器溢出时，看门狗定时器的复位信号不被激活。

18.4 看门狗定时器特殊寄存器

- (1) 看门狗定时器控制寄存器 (WTCON)
- (2) 看门狗定时器数据寄存器 (WTDAT)
- (3) 看门狗定时器计数寄存器 (WTCNT)

18.4.1 看门狗定时器控制寄存器

WATCHDOG TIMER CONTROL REGISTER (WTCON)

WTCON 寄存器允许用户使能看门狗定时器, 从不同的四个源选择时钟, 使能中断, 使能看门狗定时器输出。看门狗定时器用于 S3C2440A 上电后故障复位。如果不希望控制器复位, 则使看门狗定时器无效。

如果用户想使用看门狗定时器提供的普通定时器功能, 应使能中断并使得看门狗定时器无效。

寄存器	地址	读写	描述	复位值
WTCON	0x53000000	R/W	看门狗定时器控制寄存器	0x8021

WTCON	位	描述	初始值
Prescaler value	[15:8]	预定标器值 该值有效范围是从 0 到 255(2^8-1).	0x80
保留	[7:6]	保留。在正常操作下该值必须为 00	00
Watchdog timer	[5]	看门狗定时器使能位。 0 = 无效 1 = 有效	1
Clock select	[4:3]	决定时钟除数因子 00: 16 01: 32 10: 64 11: 128	00
Interrupt generation	[2]	中断使能位 0 = 无效 1 = 有效	0
Reserved	[1]	保留。在正常操作中, 该位必须为 0。	0
Reset enable/ disable	[0]	对于复位信号看门狗定时器输出使能位 1: 看门狗定时器超时, 发出s3c2440A复位信号。 0: 看门狗定时器复位功能无效	1

18.4.2 看门狗定时器数据寄存器

WATCHDOG TIMER DATA REGISTER (WTDAT)

WTDAT 寄存器是用于确定超时期限。WTDAT 的内容在最初的看门狗定时器操作时不能自动加载到定时器计数器中。但是使用 0x8000（初始值）将驱使第一次超时。在这种情况下，WTDAT 的值将被自动重载入 WTDAT 中。

寄存器	地址	读写	描述	复位值
WTDAT	0x53000004	R/W	看门狗定时器数据寄存器	0x8000

WTDAT	位	描述	初始值
Count reload value	[15:0]	用于重载的看门狗定时器计数值	0x8000

18.4.3 看门狗定时器计数寄存器

WATCHDOG TIMER COUNT REGISTER (WTCNT)

WTCNT 寄存器包含在正常操作期间的看门狗定时器的当前计数值。注意在看门狗定时器刚有效时，WTDAT 寄存器的内容不能自动的装载到定时器计数寄存器中。因此 WTCNT 将成为必须在使能之前设定初始值。

寄存器	地址	读写	描述	复位值
WTCNT	0x53000008	R/W	看门狗定时器计数寄存器	0x8000

WTCNT	位	描述	初始值
Count value	[15:0]	看门狗定时器的当前计数值	0x8000

第十九章 MMC/SD/SDIO 控制器

19.1 特点

- 兼容 SD 存储卡规格 (1.0 版本) 或 MMC 规格 (2.11 版本)
- 兼容 SDIO 卡规格 (1.0 版本)
- 用于数据接收发送的 16 字 (64 字节) FIFO
- 40 位的命令寄存器
- 136 位的响应寄存器
- 8 位的预定标器逻辑 (频率=系统时钟/(P+1))
- 通常模式或 DMA 传输模式 (字节、半字及字传输)
- DMA burst4 访问支持 (仅字传输)
- 1 位/4 位 (宽总线) 模式及模块/流模式开关支持

19.2 模块图

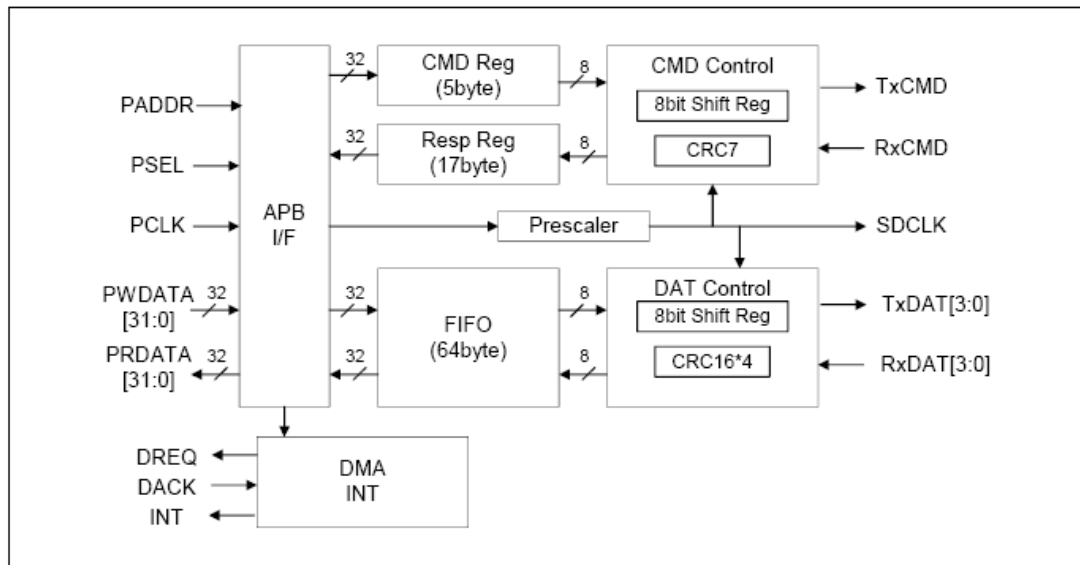


Figure 19-1. SD Interface block diagram

19.3 SD 操作

串行时钟线同步在五根数据线上的信息移位和采样。传输频率通过设定 SDIPRE 寄存器的相应位的设定来控制。你可以修改频率来调节波特率数据寄存器值。

19.3.1 编程过程 (普通)

对 SDI 模块编程, 按以下基本步骤:

- (1) 设置 SDICON 寄存器来配置适当的时钟及中断使能
- (2) 设置 SDIPRE 寄存器配置适当的值。
- (3) 等待 74 个 SDCLK 时钟以初始化卡。

19.3.2 CMD 路径编程

- (1) 写命令参数 32 位到 SDICmdArg
- (2) 决定命令类型并通过设置 SDICmdCon 开始命令传输
- (3) 当 SDICmdSta 的特殊标志被置位, 确认 SDICMD 路径操作的结束。
- (4) 如果命令类型是不相应, 标志是 CmdSent。
- (5) 如果命令类型是相应, 标志是 RspFin。
- (6) 通过对相应位写 1, 清除 SDICmdStaD 的标志。

19.3.3 数据路径编程

- (1) 写数据超时期间到 SDIDTimer
- (2) 写模块大小 (模块长度) 到 SDIBSize (通常是 0x80 字)
- (3) 确定模块模式, 宽总线, DMA 等且通过 SDIDatCon 来开始数据传输
- (4) 发送数据->写数据到数据寄存器 (SDIDAT), 当发送 FIFO 有效 (TFDET 置位), 或一半 (TFHalf 置位), 或空 (TFEmpty 置位)。
- (5) 接收数据->从数据寄存器 (SDIDAT) 读数据, 当接收 FIFO 有效 (RFDET 置位), 或满 (RFFull 置位)。或一半 (RFHalf 置位), 或准备最后数据 (RFLast 置位)。
- (6) 当 SDIDatSta 寄存器的 DatFin 标志置位, 确认 SDIDAT 路径操作结束。
- (7) 通过对相应位写 1, 清除 SDIDatSta 的标志。

19.4 SDIO 操作

有 SDIO 操作的两个功能: SDIO 中断接收和读等待请求生成。当寄存器 SDICON 的 RcvOInt 位和 RwaitEn 位分别被激活, 这两个功能可以操作。两个功能的步骤与条件如下:

19.4.1 SDIO 中断

在 SD1 位模式中, 通过 RxDAT[1] 引脚的所有范围, 中断被接收。

在 SD4 位模式中, RxDAT[1] 引脚由数据接收和中断接收共享。

当中断侦测范围 (中断期间) 是:

- (1) 单模块: A 与 B 之间的时间
 - A: 在一个数据包完成后的两个时钟
 - B: 完成发送下一个 withdata 命令结束位
- (2) 多模块, PrdType=0: A 与 B 之间的时间, C 复位
 - A: 在完成一个数据包后两个时钟
 - B: A 后的两个时钟
 - C: 在退出命令结束位相应后两个时钟
- (3) 多模块, PrdType=1: A 与 B 之间的时间, A 复位
 - A: 在完成一个数据包后两个时钟
 - B: A 后两个时钟
 - 在最后一个时钟, 中断期间开始于 A, 但是不在 B 结束 (CMD53)

19.4.2 读等待请求

不管 1 位还是 4 位模式, 在以下条件下, 读等待请求信号发送到 TxDAT[2] 引脚

- 在读多操作中, 在数据模块结束后两个时钟, 请求信号发送开始
- 当用户设置 SDIDatSta 寄存器的 RwaitReq 位, 发送结束

19.5 SDI 特殊寄存器

- (1) SDI 控制寄存器 (SDICON)
- (2) SDI 波特率预定标器寄存器 (SDIPRE)
- (3) SDI 命令参数寄存器 (SDICmdArg)
- (4) SDI 命令控制寄存器 (SDICmdCon)
- (5) SDI 响应寄存器 0 (SDIRSP0)
- (6) SDI 响应寄存器 1 (SDIRSP1)
- (7) SDI 响应寄存器 2 (SDIRSP2)
- (8) SDI 响应寄存器 3 (SDIRSP3)
- (9) SDI 数据/忙定时器寄存器
- (10) SDI 模块大小寄存器
- (11) SDI 数据控制寄存器
- (12) SDI 数据保留计数器寄存器
- (13) SDI数据状态寄存器
- (14) SDI FIFO状态寄存器
- (15) SDI中断屏蔽寄存器
- (16) SDI数据寄存器

19.5.1 SDI 控制寄存器

SDI Control Register (SDICON)

寄存器	地址	读写	描述	复位值
SDICON	0x5A000000	R/W	SDI控制寄存器	0x0

SDICON	位	描述	初始值
保留	[31:9]		
SDMMC Reset (SDreset)	[8]	复位整个sdmmc时钟。该位自动清零。 0: 正常模式 1: SDMMC复位	0
保留	[7:6]		0
Clock Type (STPY)	[5]	确定哪个时钟类型用作SDCLK 0: SD类型 1: MMC类型	0
Byte Order Type(ByteOrder)	[4]	当你使用字边界读(写)数据从(到)主FIFO, 决定字节顺序类型。 0: A型 1: B型	0
Receive SDIO Interrupt from card (RcvIOInt)	[3]	觉得是否SD主设备从卡接收SDIO中断 0: 忽略 1: 接收SDIO中断	0
Read Wait Enable(RWaitEn)	[2]	决定读等待请求信号生成, 当SD主设备在多模块读模式下等待下一个时钟。该位需要延时到下一个时钟由卡发出。 0: 无效(不生成) 1: 读等待使能(使用SDIO)	0
保留	[1]		
Clock Out Enable (ENCLK)	[0]	确定是否SDCLK输出使能 0: 无效(预定标器关闭) 1: 时钟使能	0

* Byte Order Type

- Type A: (Access byWord) D[7:0] -> D[15:8] -> D[23:16] -> D[31:24]

- (Access by Halfword) D[7:0] → D[15:8]
- Type B: (Access by Word) D[31:24] → D[23:16] → D[15:8] → D[7:0]
(Access by Halfword) D[15:8] → D[7:0]

19.5.2 SDI 波特率预定标器寄存器

SDI Baud Rate Prescaler Register (SDIPRE)

寄存器	地址	读写	描述	复位值
SDIPRE	0x5A000004	R/W	SDI波特率预定标器寄存器	0x01

SDIPRE	位	描述	初始值
Prescaler Value	[7:0]	按以上等式确定SDI时钟率 (SDCLK)	0x01

19.5.3 SDI 命令参数寄存器

SDI Command Argument Register (SDICmdArg)

寄存器	地址	读写	描述	复位值
SDICmdArg	0x5A00000C	R/W	SDI命令控制寄存器	0x0

SDICmdCon	位	描述	初始值
保留	[31:13]		
Abort Command (AbortCmd)	[12]	决定是否命令类型用于退出(for SDIO). 0 = 正常命令, 1 = 退出命令(CMD12, CMD52)	0
Command with Data (WithData)	[11]	决定是否命令类型有数据(for SDIO). 0 = 无数据, 1 = 有数据	0
LongRsp	[10]	决定是否主设备接收一个 136 位长的响应 0 = 短响应, 1 = 长响应	0
WaitRsp	[9]	决定是否主设备等待响应 0 = 不响应, 1 = 等待响应	0
Command Start(CMST)	[8]	决定命令操作是否开始。该位自动清零。 0 = 命令准备好, 1 = 命令开始	0
CmdIndex	[7:0]	有开始两位的命令索引(8bit)	0x00

19.5.4 SDI 命令状态寄存器

SDI Command Status Register (SDICmdSta)

寄存器	地址	读写	描述	复位值
SDICmdSta	0x5A000010	R/W	SDI命令状态寄存器	0x0

SDICmdSta	位	描述	初始值
保留	[31:13]		
Response CRC Fail (RspCrc)	[12]	当收到命令响应, CRC校验失败。通过对该位置 1, 该标志被清除。 0: 不检测 1: CRC失败	0
Command Sent (CmdSent)	[11]	命令发送 (不包括响应)。通过对该位置 1, 该标志被清除。 0: 不检测 1: 命令结束	0
Command Time Out (CmdTout)	[10]	命令响应超时 (64clk)。通过对该位置 1, 该标志被清除。 0: 不检测 1: 超时	0
Response Receive End (RspFin)	[9]	收到命令响应。通过对该位置 1, 该标志被清除。 0: 不检测 1: 响应结束	0
CMD line progress On (CmdOn)	[8]	命令传输在过程中。 0: 不检测 1: 命令在过程中	0
RspIndex	[7:0]	有开始两位 (8 位) 的响应索引 6 位	0x00

19.5.5 SDI 响应寄存器 0

SDI Response Register 0(SDIRSP0)

寄存器	地址	读写	描述	复位值
SDIRSP0	0x5A000014	R	SDI响应寄存器 0	0x0

SDIRSP0	位	描述	初始值
Resonse0	[31:0]	Card status[31:0](short), card status[127:96](long)	0x00000000

19.5.6 SDI 响应寄存器 1

SDI Response Register1 (SDIRSP1)

寄存器	地址	读写	描述	复位值
SDIRSP1	0x5A000018	R	SDI响应寄存器 1	0x0

SDIRSP1	位	描述	初始值
RCRC7	[31:24]	CRC7 (有结束位, 短), card status[95:88](long)	0x00
Resonse1	[23:0]	不使用 (短), card status[87:64](long)	0x000000

19.5.7 SDI 响应寄存器 2

SDI Response Register 2 (SDIRSP2)

寄存器	地址	读写	描述	复位值
SDIRSP2	0x5A00001C	R	SDI响应寄存器 2	0x0

SDIRSP2	位	描述	初始值
Resonse2	[31:0]	unused(short), card status[63:32](long)	0x00000000

19.5.8 SDI 响应寄存器 3

SDI Response Register 3 (SDIRSP3)

寄存器	地址	读写	描述	复位值
SDIRSP3	0x5A00001C	R	SDI响应寄存器 3	0x0

SDIRSP3	位	描述	初始值
Resonse3	[31:0]	unused(short), card status[31:0](long)	0x00000000

19.5.9 SDI 数据/忙定时器寄存器

SDI Data / Busy Timer Register (SDIDTimer)

寄存器	地址	读写	描述	复位值
SDIDTimer	0x5A000024	R/W	SDI数据/忙定时器寄存器	0x0

SDIDTimer	位	描述	初始值
保留	[31:23]		
DataTimer	[22:0]	Data / Busy timeout period	0x10000

19.5.10 SDI 模块大小寄存器

SDI Block Size Register (SDIBSize)

寄存器	地址	读写	描述	复位值
SDIBSize	0x5A000028	R/W	SDI模块大小寄存器	0x0

SDIBSize	位	描述	初始值
保留	[31:12]		
BlkSize	[11:0]	Block Size value(0~4095 byte), 流模式下不考虑	0x10000

在多模块情况下, BlkSize 必须分配字大小(4 字节)。(BlkSize[1:0]=00)

19.5.11 SDI 数据控制寄存器

SDI Data Control Register (SDIDatCon)

寄存器	地址	读写	描述	复位值
SDIDatCon	0x5A00002C	R/W	SDI数据控制寄存器	0x0

SDIBSize	位	描述	初始值
保留	[31:25]		
Burst4 enable (Burst4)	[24]	在DMA模式下使能Burst4。仅当数据大小是字时该位被置位。0: 无效 1: Burst4 使能	0
Data Size (DataSize)	[23:22]	指出用FIFO传输的大小, 哪个类型, 半字或字。 00 = 字节传输, 01 = 半字传输 10 = 字传输, 11 = 保留	0
SDIO Interrupt Period Type (PrdType)	[21]	决定SDIO的中断周期是 2 个周期还是外部更多周期, 当数据块最后被发送 (对SDIO)。 0=正好两个周期 1=更多周期 (像单周期)	0
Transmit After Response (TARSP)	[20]	决定数据传输在响应收到后开始或不开始。 0= 在DatMode设置后直接 1= 在响应收到后 (假定设置DatMode设为 2b11)	0
Receive After Command (RACMD)	[19]	决定数据传输在命令发出后开始或不开始 0= 在DatMode设置后直接 1= 在命令发出后 (假定设置DatMode设为 2b10)	0
Busy After Command (BACMD)	[18]	决定忙接收在命令发出后开始或不开始 0= 在DatMode设置后直接 1= 在命令发出后 (假定设置DatMode设为 2b01)	0
Block mode (BlkMode)	[17]	数据传输模式 0=流数据传输 1=模块数据传输	0
Wide bus enable (WideBus)	[16]	决定使能宽总线模式 0: 标准总线模式 (仅使用SDIDAT[0]) 1: 宽总线模式 (使用SDIDAT[3])	0
DMA Enable (EnDMA)	[15]	使能DMA (当DMA操作完成时, 该位应该无效) 0: 无效 (查询) 1: DMA使能	0
Data Transfer Start(DTST)	[14]	决定数据传输是否开始, 该位自动清除。 0: 数据准备好, 1: 数据开始	0
Data Transfer Mode (DatMode)	[13:12]	决定数据传输的方向 00 = 无操作, 01 = 仅忙检测模式 10 = 数据接收模式, 11 = 数据发送模式	00
BlkNum	[11:0]	模块数(0~4095), 当流模式时不考虑	0x000

如果你想对TARSP, RACMD, BACMD位(SDIDatCon[20:18])中的一个位置 1, 你需要在写SDICmdCon寄存器之前写SDIDatCon寄存器。(对SDIO总是需要)

19.5.12 SDI 数据保留计数器寄存器

SDI Data Remain Counter Register (SDIDatCnt)

寄存器	地址	读写	描述	复位值
SDIDatCnt	0x5A000030	R	SDI数据保留计数器寄存器	0x0

SDIDatCnt	位	描述	初始值
保留	[31:24]		
BlkNumCnt	[23:12]	保留模块数	0x000
BlkCnt	[11:0]	1 个模块的保留数据字节	0x000

19.5.13 SDI数据状态寄存器

SDI Data Status Register (SDIDatSta)

寄存器	地址	读写	描述	复位值
SDIDatSta	0x5A000034	R	SDI数据状态寄存器	0x0

SDIDatSta	位	描述	初始值
保留	[31:12]		
No Busy (NoBusy)	[11]	仅在忙检测模式下cmd包发送后, 在 16 个周期期间忙不激活。通过对该位置 1 清除标志。 0: 不侦测 1: 无忙信号	0
Read Wait Request Occur (RWaitReq)	[10]	读等待请求信号发送到sd卡。请求信号停止且通过对该位置 1 清除标志。 0: 不出现 1: 读等待请求出现	0
SDIO Interrupt Detect (IOIntDet)	[9]	SDIO中断侦测。通过对该位置 1 清除标志。 0: 不侦测 1: SDIO中断侦测	0
保留	[8]		
CRC Status Fail (CrcSta)	[7]	当数据块发送后 (CRC校验失败) CRC状态错误。通过对该位置 1 清除标志。 0: 不侦测 1: CRC状态失败	0
Data Receive CRC Fail (DatCrc)	[6]	数据模块接收到错误 (CRC校验失败)。通过对该位置 1 清除标志。 0: 不侦测 1: 接收CRC失败	0
Data Time Out (DatTout)	[5]	数据/忙接收超时。通过对该位置 1 清除标志。 0: 不侦测 1: 超时	0
Data Transfer Finish (DatFin)	[4]	数据传输结束 (数据计数器为 0)。通过对该位置 1 清除标志。 0: 不侦测 1: 数据完成侦测	0
Busy Finish (BusyFin)	[3]	仅忙检查完成。通过对该位置 1 清除标志。 0: 不检测 1: 侦测忙完成	0
保留	[2]		0
Tx Data progress On (TxDatOn)	[1]	数据发送在过程中 0: 不激活 1: 数据发送在过程中	0
Rx Data Progress On (RxDatOn)	[0]	数据接收在过程中 0: 不激活 1: 数据接收在过程中	0

19.5.14 SDI FIFO 状态寄存器

SDI FIFO Status Register (SDIFSTA)

寄存器	地址	读写	描述	复位值
SDIFSTA	0x5A000038	R	SDI FIFO状态寄存器	0x0

SDIFSTA	位	描述	初始值
保留	[31:16]		
FIFO Reset(FRST)	[16]	FIFO值复位。该位会自动清楚 0: 正常模式 1: FIFO复位	0
FIFO Fail error (FFfail)	[15:14]	当FIFO出现溢出或低估数据保存是, FIFO失败错误。该位通过置位清除。 00: 不侦测 01: FIFO失败 10 在最后传输中FIFO失败 (仅FIFO复位需要) 11: 保留	00
FIFO available Detect for Tx (TFDET)	[13]	该位指出FIFO数据对发送有效, 当DatMode是数据发送模式。如果DMA模式有效, SD主设备请求DMA操作。 0: 不侦测 (FIFO满) 1: 侦测 (1<FIFO<63)	0
FIFO available Detect for Rx (RFDET)	[12]	该位指出FIFO数据对接收有效, 当DatMode是数据接收模式, 如果DMA模式有效, SD主设备请求DMA操作。 0: 不侦测 (FIFO空) 1: 侦测 (1<FIFO<64)	0
Tx FIFO Half Full (TFHalf)	[11]	只要发送FIFO小于 33 字节, 该位置 1。 0: 33<Tx FIFO<64 1: 1<Tx FIFO<32	0
Tx FIFO Empty (TFEmpty)	[10]	只要发送FIFO为空, 该位置 1。 0: 1<FIFO<64 1: 空 (0 字节)	0
Rx FIFO Last Data Ready (RFLast)	[9]	当接收FIFO出现操作所有模块的最后数据, 该位置 1。通过置 1 清除该标志 0: 还没收到 1: 接收FIFO得到最后数据	0
Rx FIFO Full (RFFull)	[8]	只要接收FIFO满, 该位置 1。 0: 1<Rx FIFO<63 1: 满 (64 字节)	0
Rx FIFO Half Full (RFHalf)	[7]	只要接收FIFO大于 31 个字节, 该位置 1。 0: 1<FIFO<31 1: 32<FIFO<64	0
FIFO Count (FFCNT)	[6:0]	FIFO中的数据个数	0000000

19.5.15 SDI 中断屏蔽寄存器

SDI Interrupt Mask Register (SDIIntMsk)

寄存器	地址	读写	描述	复位值
SDIIntMsk	0x5A00003C	R/W	SDI中断屏蔽寄存器	0x0

SDIFSTA	位	描述	初始值
保留	[31:19]		
NoBusy Interrupt Enable (NoBusyInt)	[18]	如果忙信号不激活, 决定SDI产生中断。 0 =无效, 1 = 中断使能	0
RspCrc Interrupt Enable (RspCrcInt)	[17]	如果响应CRC校验失败, 决定SDI产生中断 0 =无效, 1 = 中断使能	0
CmdSent Interrupt Enable (CmdSentInt)	[16]	如果命令发出, 决定SDI产生中断 0 =无效, 1 = 中断使能	0
CmdTout Interrupt Enable (CmdToutInt)	[15]	如果命令响应超时, 决定SDI产生中断 0 =无效, 1 = 中断使能	0
RspEnd Interrupt Enable (RspEndInt)	[14]	如果收到命令响应, 决定SDI产生中断 0 =无效, 1 = 中断使能	0
RWaitReq Interrupt Enable (RWaitReqInt)	[13]	如果出现读等待请求, 决定SDI产生中断 0 =无效, 1 = 中断使能	0
IOIntDet Interrupt Enable (IOIntDetInt)	[12]	如果sd主设备从卡收到SDIO中断, 决定SDI产生中断。0 =无效, 1 = 中断使能	0
FFfail Interrupt Enable (FFfailInt)	[11]	如果出现FIFO失败错误, 决定SDI产生中断。0 =无效, 1 = 中断使能	0
CrcSta Interrupt Enable (CrcStaInt)	[10]	如果出现CRC状态错误, 决定SDI产生中断。0 =无效, 1 = 中断使能	0
DatCrc Interrupt Enable (DatCrcInt)	[9]	如果数据接收CRC失败, 决定SDI产生中断。0 =无效, 1 = 中断使能	0
DatTout Interrupt Enable (DatToutInt)	[8]	如果数据接收超时出现, 决定SDI产生中断。0 =无效, 1 = 中断使能	0
DatFin Interrupt Enable (DatFinInt)	[7]	如果数据计数器为 0, 决定SDI产生中断。0 =无效, 1 = 中断使能	0
BusyFin Interrupt Enable (BusyFinInt)	[6]	如果仅忙检测完成, 决定SDI产生中断。0 =无效, 1 = 中断使能	0
保留	[5]		0
TFHalf Interrupt Enable (TFHalfInt)	[4]	如果发送FIFO半填满, 决定SDI产生中断。0 =无效, 1 = 中断使能	0
TFFEmpty Interrupt Enable (TFFEmptyInt)	[3]	如果发送FIFO空, 决定SDI产生中断。0 =无效, 1 = 中断使能	0
RFLast Interrupt Enable (RFLastInt)	[2]	如果接收FIFO有最后数据, 决定SDI产生中断。0 =无效, 1 = 中断使能	0
RFFull Interrupt Enable (RFFullInt)	[1]	如果接收FIFO填满, 决定SDI产生中断。0 =无效, 1 = 中断使能	0
RFHalf Interrupt Enable (RFHalfInt)	[0]	如果接收FIFO半满, 决定SDI产生中断。0 =无效, 1 = 中断使能	0

19.5.16 SDI 数据寄存器

SDI Data Register (SDIDAT)

寄存器	地址	读写	描述	复位值
SDIDAT	0x5A000040, 44, 48, 4C(Li/W, Li/HW, Li/B, Bi/W) 0x5A000041(Bi/HW), 0x5A000043(Bi/B)	R/W	SDI数据寄存器	0x0

SDIDAT	位	描述	初始值
Data Register	[31:19]	该区域包括通过SDI通道发送或接收的数据	0x00000000

* (Li/W, Li/HW, Li/B): 小端模式下通过字、半字、字节访问。

* (Bi/W): 大端模式下通过字单元访问。

* (Bi/HW): 大端模式下通过半字单元访问。

* (Bi/B): 大端模式下通过字节单元访问。

第二十章 IIC 总线接口

20.1 概述

S3C2440A RISC 微处理器可以支持多主设备 IIC 总线串行接口。专用串行数据线（SDA）和串行时钟线（SCL）承载总线主设备和连接 IIC 总线的外围设备之间的信息。SDA 和 SCL 线都是双向的。

在多主设备 IIC 总线模式下，多个 S3C2440A 微处理器可以从从属设备接收或发送串行数据。主设备 S3C2440A 可以初始化和终止一个基于 IIC 总线的数据传输。在 S3C2440A 中的 IIC 总线使用标准总线仲裁步骤。

为了控制多主设备 IIC 总线操作，这些值必须写入以下寄存器：

- 多主设备 IIC 总线控制寄存器（IICCON）
- 多主设备 IIC 总线控制状态寄存器（IICSTAT）
- 多主设备 IIC 总线接收发送数据移位寄存器（IICDS）
- 多主设备 IIC 总线地址寄存器（IICADD）

当 IIC 总线是空闲的，SDA 和 SCL 线应该都是高电平。SDA 从高到低的变化能够初始化一个开始条件。当 SCL 保持稳定在高电平下，SDA 从低到高的变化可以初始化一个停止条件。

开始和停止条件都是由主设备生成。在第一个字节中的一个 7 位的地址值可以决定一个由主设备选择的从设备，其地址值在开始条件初始化后被放到总线上。第 8 位决定的是传输方向（读或写）。

放到 SDA 线上的每个数据字节总共应该是 8 位。在总线传输期间，该字节可以被无限制的发送或接收。数据发送总是先对 MSB，每个字节应该紧跟一个应答位（ACK）。

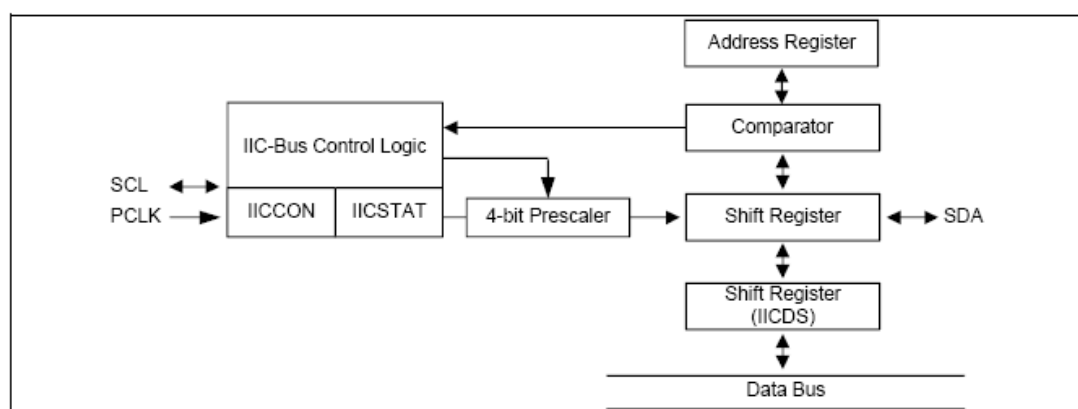


Figure 20-1. IIC-Bus Block Diagram

20.1.1 IIC 总线接口

S3C2440A 的 IIC 总线接口有四个操作模式

- 主设备发送模式
- 主设备接收模式
- 从设备发送模式
- 从设备接收模式

这些操作模式间功能关系将在下面描述。

20.1.2 开始和停止条件

当 IIC 总线接口是不活动，其通常是在从设备模式。换言之，接口在检测到 SDA 线上的开始条件之间一直处于从设备模式（开始条件可以被初始化，当 SCL 时钟信号是高电平是 SDA 线下跳变）。当接口状态变为主设备模式，在 SDA 线上的数据传输被初始化且 SCL 信号生成。

开始条件可以通过 SDA 线传输一个字节串行数据，一个停止条件可以终止一个数据传输。停止条件是当 SCL 是高电平时，SDA 线从低电平到高电平的跳变。开始和停止条件都是由主设备生成。当开始条件生成，IIC 总线忙。停止条件将使 IIC 总线空闲。

当一个主设备初始化开始条件，它应该发送一个从地址来停止从设备。一个字节的地址域包含 7 位地址和一位传输方向指示（说明读写）。如果位 8 是 0，说明是写操作。如果位 8 是 1 说明是数据读请求。

主设备通过发送停止条件来完成一个传输操作。如果主设备想继续到总线的数据传输，他应该生成另一个开始条件和从地址。用这个方法，读写操作可以在不同形式下执行。

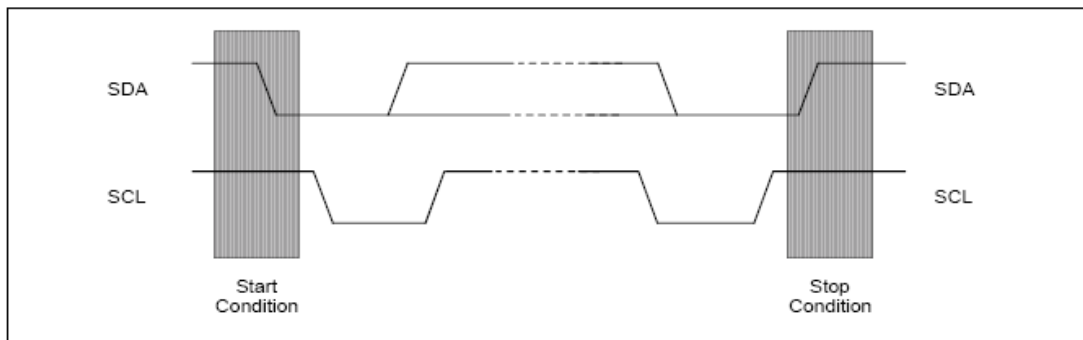


Figure 20-2. Start and Stop Condition

20.1.3 数据传输格式

每个放到 SDA 线上的字节应该是 8 位长度。该字节应该被无限制的传输。紧接着开始条件的第一个字节应该有一个地址区域。当 IIC 在主设备模式下操作，地址区域应该由主设备发送。每个字节后面应该紧跟一个应答位（ACK）。串行数据的 MSB 位和地址总是先发送。

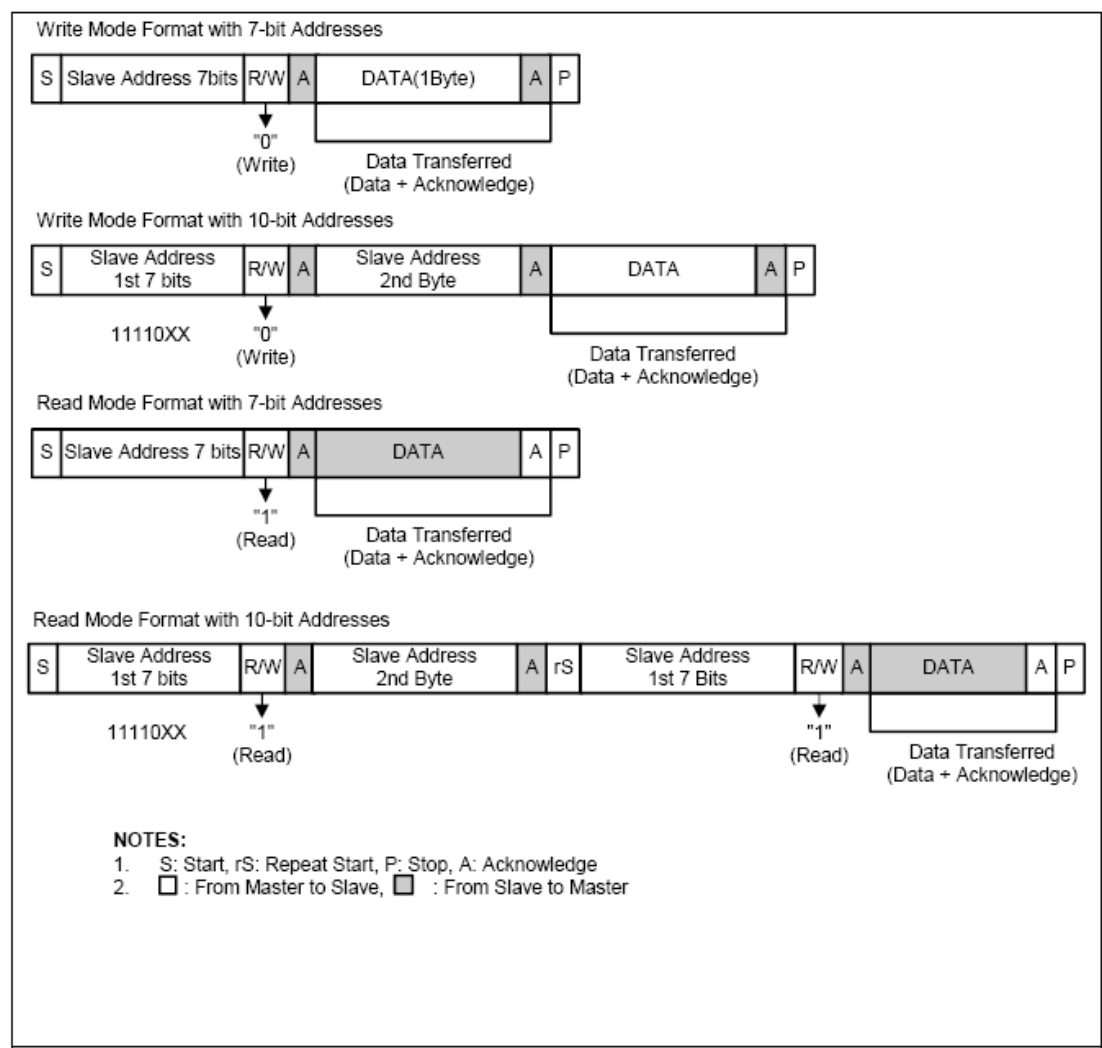


Figure 20-3. IIC-Bus Interface Data Format

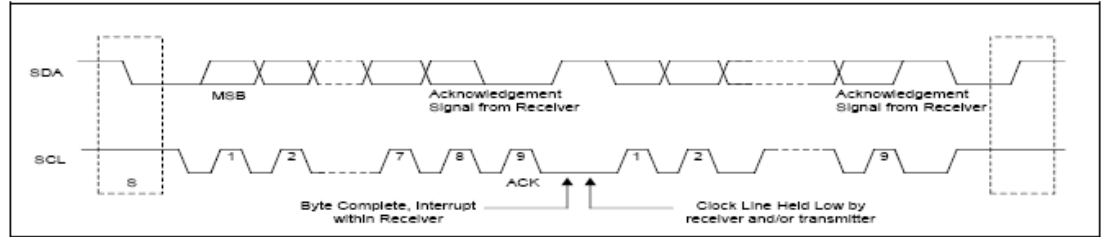


Figure 20-4. Data Transfer on the IIC-Bus

20.1.4 ACK 信号传输

为了完成一个字节的传输操作，接收器应该发送一个 ACK 位给发送器。ACK 脉冲应该出现在 SCL 线的第九个时钟。一个字节数据传输需要 8 个时钟。主设备生成传输 ACK 位所需的时钟脉冲。

发送器应该在接收到 ACK 时钟脉冲以后通过拉高 SDA 线释放 SDA 线。接收器也应该在 ACK 时钟脉冲期间拉低 SDA 线，以至于在第九个 SCL 脉冲为高电平期间 SDA 保持低电平。

ACK 位发送功能可以通过软件（IICSTAT）使能或使无效。但是在 SCL 第九个时钟的 ACK 脉冲是用于完成一个字节数据传输。

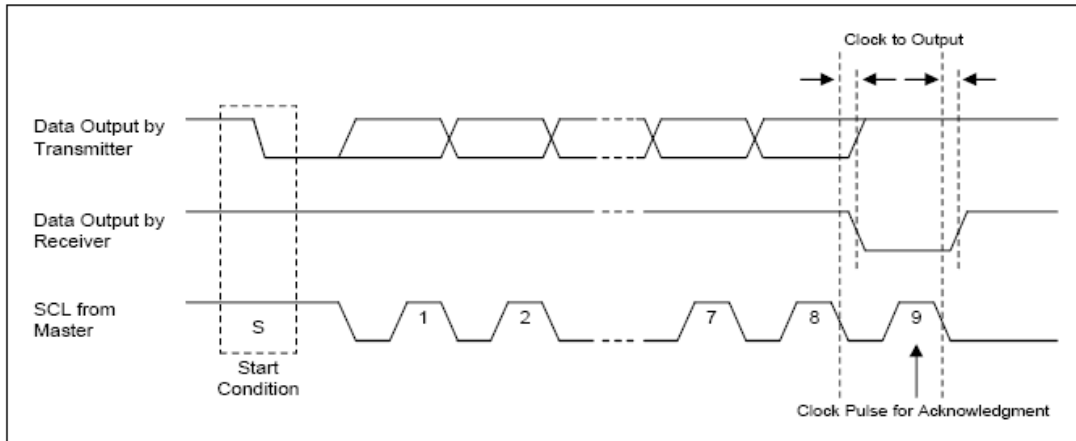


Figure 20-5. Acknowledge on the IIC-Bus

20.1.5 读写操作

在发送模式下，如果数据传输后，IIC 总线接口会等待直到 IIC 总线数据移位寄存器（IICDS）收到一个新数据。在新数据写入寄存器之前，SCL 线将保持低电平，然后在数据写入后释放。S3C2440A 应该保持中断来识别当前数据发送是否完成。在 CPU 收到中断请求以后，它应该再写一个新数据到 IICDS 寄存器。

在接收模式下，当数据收到后，IIC 总线接口应该等待直到 IICDS 寄存器被读取。在一个新数据被读出之前，SCL 线应该保持低电平，在数据被读出后再释放。S3C2440A 应该保持中断来识别当前数据接收是否完成。在 CPU 收到中断请求以后，它应该从 IICDS 寄存器读取数据。

20.1.6 总线仲裁步骤

仲裁是为了避免两个主设备在总线上对 SDA 线的争夺。如果一个设置 SDA 线位高电平的主设备发现了另一个设置 SDA 线为低电平的主设备，它将无法初始化一个数据传输，因为总线当前电平不满足它自己设定的电平。仲裁过程将展开知道 SDA 线位变为高电平。

但是，当主设备同时拉低 SDA 线，每个主设备应该估计是否控制器是分配给自己。对于估计的目的是每个主设备应该检测地址位。当每个主设备生成从属设备地址时，它也应该侦测 SDA 线上的地址位，因为 SDA 线更可能拉低而不是变高。假定一个主设备生成一个低电平信号作为第一个地址位，而另一个主设备保持高电平。在这种情况下，两个主设备都会侦测到总线上是低电平，因为在低电平状态优先于高电平状态。如果这种情况发生，产生低电平的主设备应该获得控制权而产生高电平的主设备应该放弃总线控制权。如果两个设备都产生一个低电平作为地址的第一个位，则应该再仲裁第二个地址位。仲裁会一直继续直到最后一个地址位。

20.1.7 中止条件

如果一个从接收器不能应答从属设备地址的确认，它应该保持 SDA 线的电平为高。在这种情况下，主设备应该生成一个停止条件并中止传输。

如果主设备的接收器也参与了被中止的传输，它应该通过在从从设备收到最后数据字节后取消 ACK 信号的生成，给从设备传输操作的最后发信号。从设备发送器应该释放 SDA 线以允许主设备产生一个停止条件。

20.1.8 配置 IIC 总线

为了控制串行时钟（SCL）的频率，4 位的预定标器值可以通过 IICCON 寄存器来被编程。IIC 总线接口地址是存储在 IIC 总线地址寄存器（IICADD）中。（默认情况下，IIC 总线接口地址有一个未知值）

20.1.9 每个模式下的操作流程图

以下步骤必须在任何 IIC 接收发送操作之前被执行。

- (1) 如果需要，写自己从属地址到 IICADD 寄存器。
- (2) 设置 IICCON 寄存器
 - (a) 使能中断
 - (b) 定义 SCL 期间
- (3) 设置 IICSTAT 寄存器来使能串行输出。

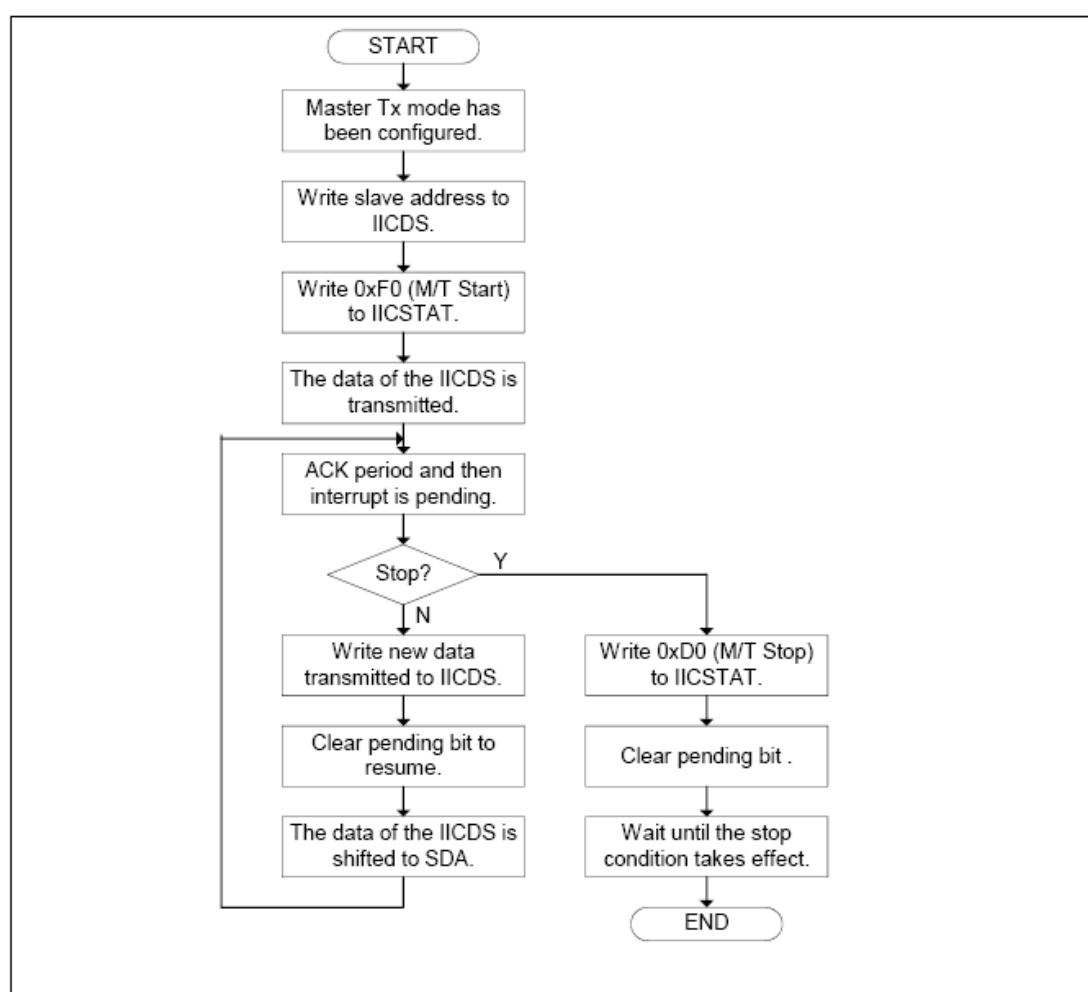


Figure 20-6 Operations for Master/Transmitter Mode

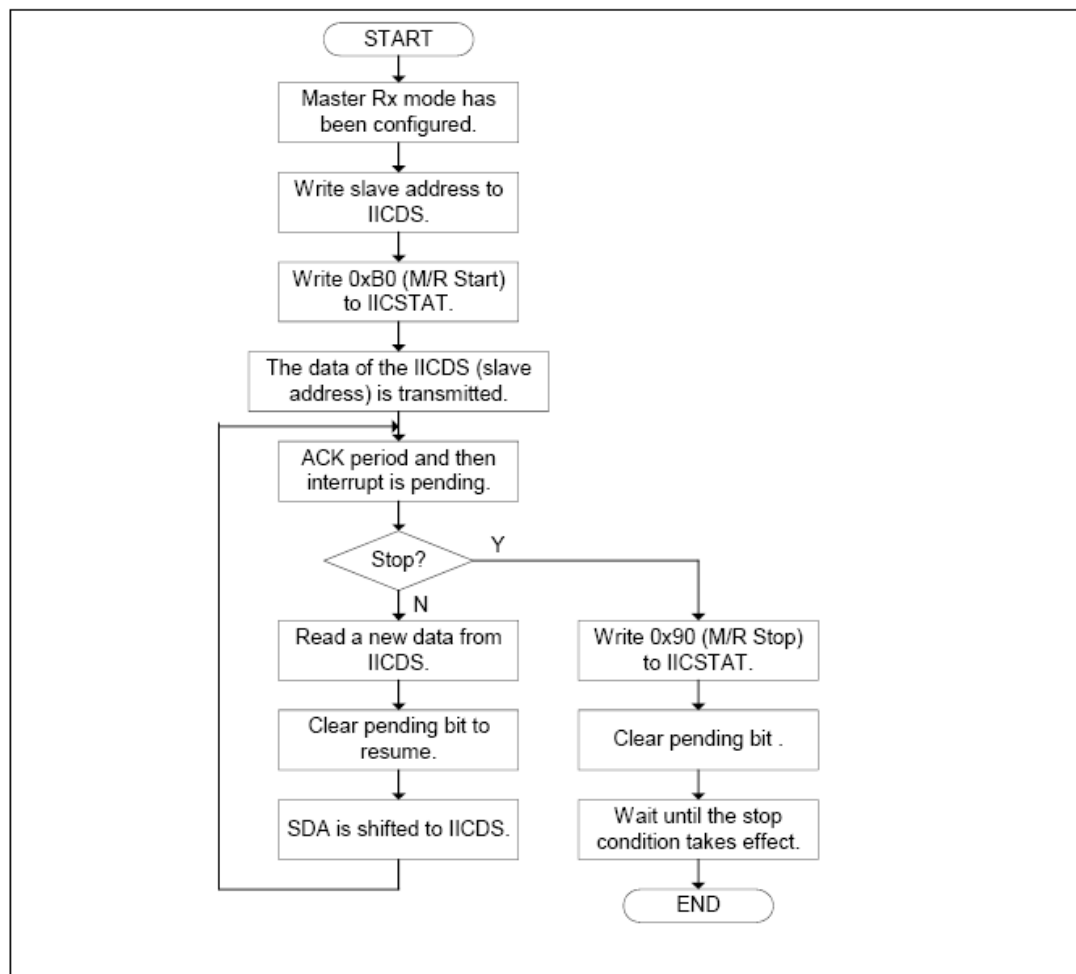


Figure 20-7 Operations for Master/Receiver Mode

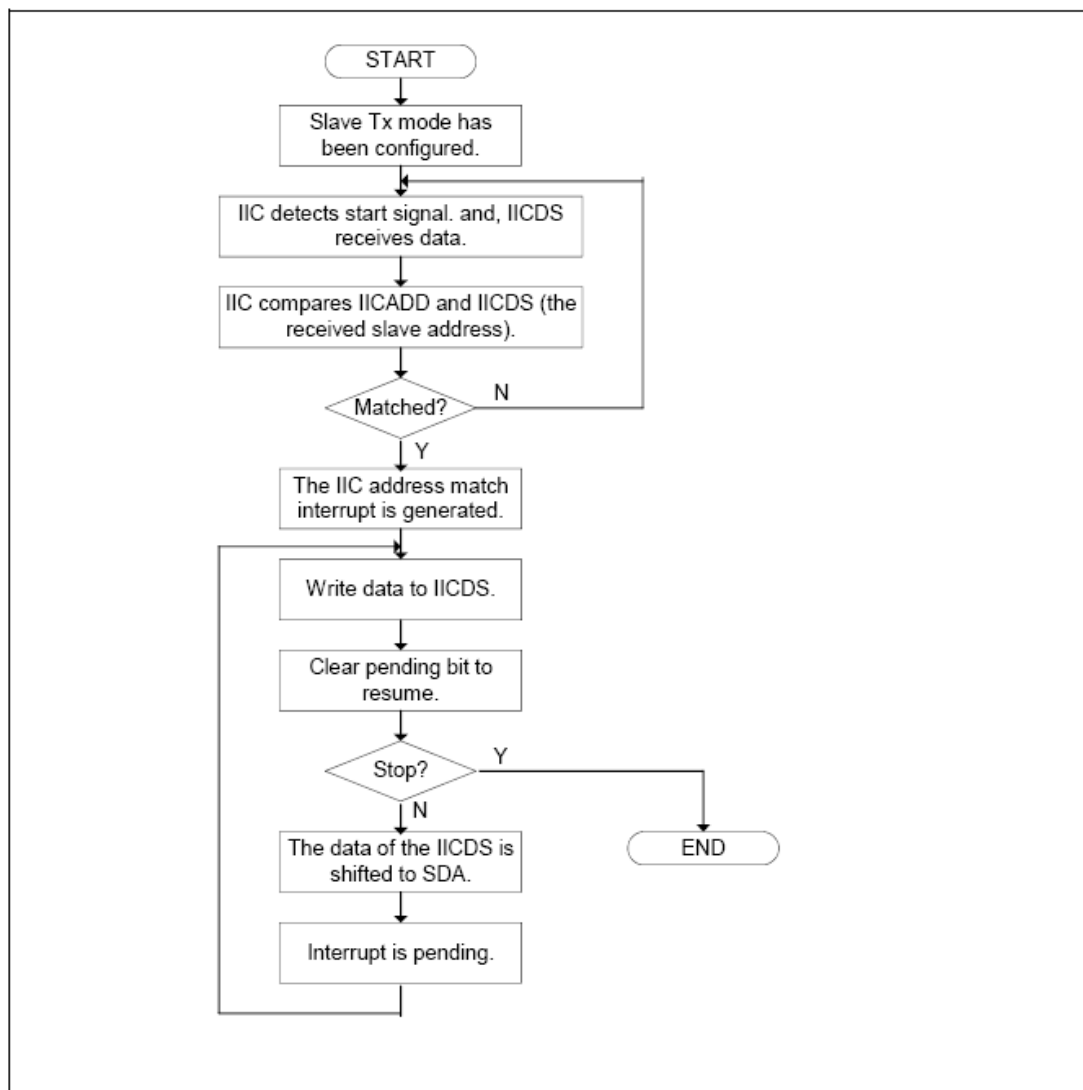


Figure 20-8 Operations for Slave/Transmitter Mode

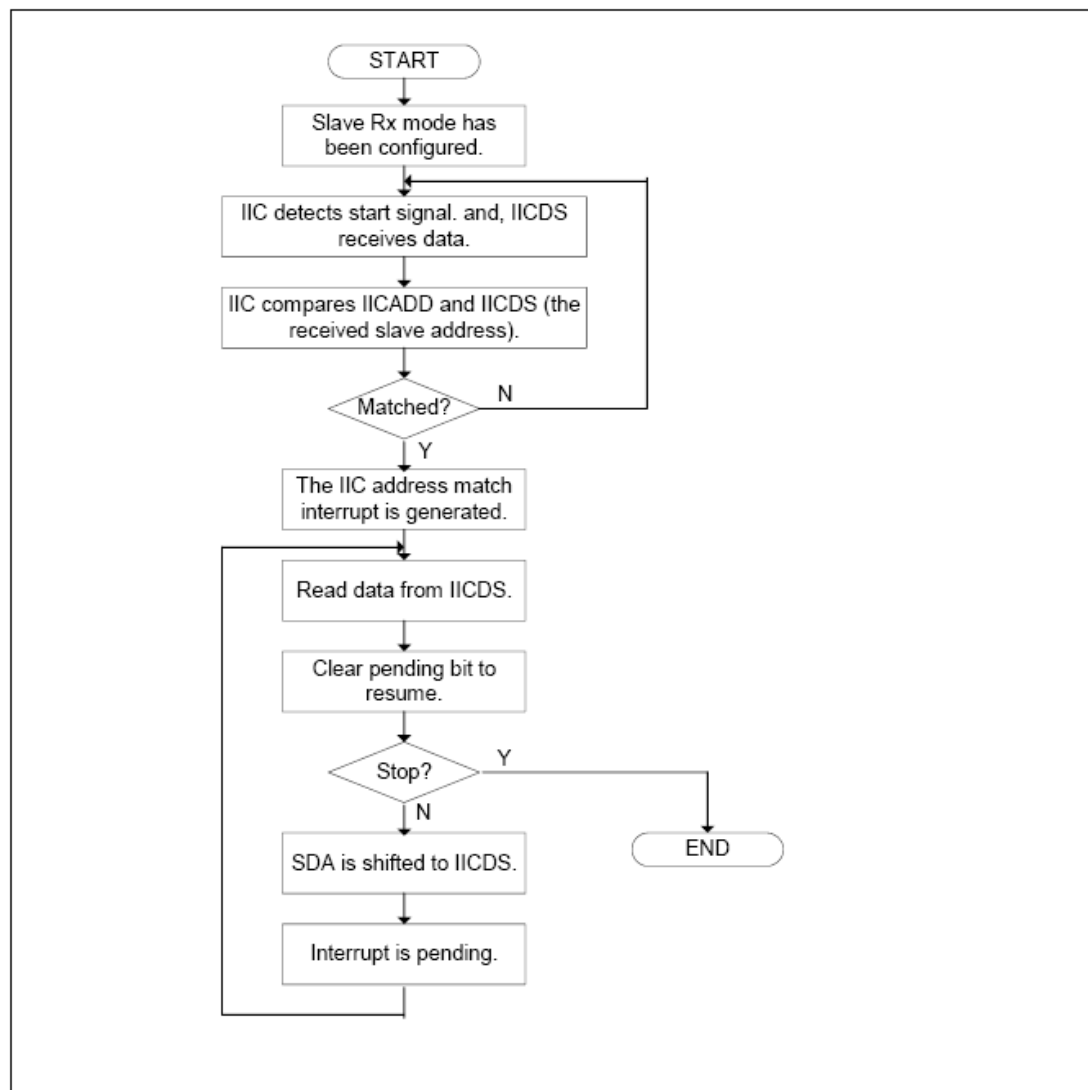


Figure 20-9 Operations for Slave/Receiver Mode

20.2 IIC 总线接口特殊寄存器

- (1) 多主设备 IIC 总线控制寄存器 (IICCON)
- (2) 多主设备 IIC 总线控制状态寄存器 (IICSTAT)
- (3) 多主设备 IIC 总线地址寄存器 (IICADD)
- (4) 多主设备 IIC 总线接收发送数据移位寄存器 (IICDS)
- (5) 多主设备 IIC 总线线路控制寄存器 (IICLC)

20.2.1 多主设备 IIC 总线控制寄存器

MULTI-MASTER IIC-BUS CONTROL REGISTER (IICCON)

寄存器	地址	读写	描述	复位值
IICCON	0x54000000	R/W	IIC总线控制寄存器	0xX

IICCON	位	描述	初始值
Acknowledge generation ⁽¹⁾	[7]	IIC总线应答使能位 0: 无效 1: 有效 在发送模式下, IICSDA在应答时间内是空闲 在接收模式下, IICSDA在应答时间内是L	0
Tx clock source selection	[6]	IIC总线传输时钟预定标器源时钟选择位 0: IICCLK = f _{PCLK} /16 1: IICCLK = f _{PCLK} /512	0
Tx/ Rx Interrupt ⁽⁵⁾	[5]	IIC总线接收发送中断使能位 0: 无效 1: 有效	0
Interrupt pending flag ^{(2) (3)}	[4]	IIC总线接收发送中断挂起标志。该位不能被写 1。当该位写 1, IICSDA为L且IIC停止。为了恢复操作, 清 0 该位。 0: (1) 无中断挂起 (读) (2) 清除挂起条件&恢复操作 (写) 1: (1) 中断挂起 (读) (2) N/A (写)	0
Transmit clock value ⁽⁴⁾	[3:0]	IIC总线发送时钟预定标器。 IIC总线发送时钟频率由该 4 位预定标器的值决定, 根据以下公式: Tx clock = IICCLK/(IICCON[3:0]+1)	UDF

注:

- (1) 接口 EEPROM, 在读取最后数据之前应答生成设置为无效, 目的是在接收模式下生成停止条件。
- (2) IIC 总线中断出现
 - 1) 当一个字节发送或接收操作完成
 - 2) 当一个 general call 或从设备地址匹配出现
 - 3) 如果总线仲裁失败
- (3) 为了在 IICSDA 上升沿之前调整 IICSDA 的建立时间, IICDS 必须在清除 IIC 中断挂起位之前写。
- (4) IICCLK 由 IICCON[6]决定。发送时钟在 SCL 跳变时间内改变。当 IICCON[6]=0, IICCON[3:0]=0x0 或 0x1 是没用的。
- (5) 如果 IICCON[5]=0, IICCON[4]没有正确操作。因此推荐设置 IICCON[5]=1, 景观你

没有使用 IIC 中断。

20.2.2 多主设备 IIC 总线控制状态寄存器

MULTI-MASTER IIC-BUS CONTROL/STATUS REGISTER (IICSTAT)

寄存器	地址	读写	描述	复位值
IICSTAT	0x54000004	R/W	IIC总线控制状态寄存器	0x0

IICSTAT	位	描述	初始值
Mode selection	[7:6]	IIC总线主从接收发送模式选择位 00: 从接收模式 01: 从发送模式 10: 主接收模式 11: 主发送模式	00
Busy signal status / START STOP condition	[5]	IIC总线忙状态位 0: (读) 不忙 1: (读) 忙 (写) 停止信号生成 (写) 开始信号输出 在开始信号后, IICDS中的数据自动被传输。	0
Serial output	[4]	IIC总线数据输出使能位。 0: 无效Rx/Tx 1: 有效Rx/Tx	0
Arbitration status flag	[3]	IIC总线仲裁过程状态标志位 0: 总线仲裁成功 1: 在串行IO中总线仲裁失败	0
Address-as-slave status flag	[2]	IIC总线address-as-slave状态标志位 0: 当检测到开始或停止条件, 该位被清除 1: 收到的从设备地址和IICADD中的地址匹配	0
Address zero status flag	[1]	IIC总线地址 0 状态标志位 0: 当检测到开始或停止条件, 该位被清除 1: 收到的从地址是 00000000b	0
Last-received bit status flag	[0]	IIC总线最后收到位状态标志位 0: 最后收到位是 0 (收到ACK) 1: 最后收到位是 1 (未收到ACK)	0

20.2.3 多主设备 IIC 总线地址寄存器

MULTI-MASTER IIC-BUS ADDRESS REGISTER (IICADD)

寄存器	地址	读写	描述	复位值
IICADD	0x54000008	R/W	IIC总线地址寄存器	0xXX

IICADD	位	描述	初始值
Slave address	[7:0]	从IIC总线锁存的 7 位从设备地址。 当串行输出使能=0, IICADD是写使能。不管当前串行输出使能位(IICSTAT中)设置, IICADD的值可以在任何时间读取。 Slave address : [7:1] Not mapped : [0]	XXXXXXXX

20.2.4 多主设备 IIC 总线接收发送数据移位寄存器

MULTI-MASTER IIC-BUS TRANSMIT/RECEIVE DATA SHIFT REGISTER (IICDS)

寄存器	地址	读写	描述	复位值
IICDS	0x5400000C	R/W	IIC总线发送接收数据移位寄存器	0xXX

IICDS	位	描述	初始值
Data shift	[7:0]	对于IIC总线发送接收操作的 8 位数据移位寄存器。 当在IICSTAT寄存器中的串行输出使能=1，IICDS是写使能。不管当前串行输出使能位（IICSTAT中）设置，IICDS的值应该可以在任何时间读取	XXXXXXXX

20.2.5 多主设备 IIC 总线线路控制寄存器

MULTI-MASTER IIC-BUS LINE CONTROL REGISTER (IICLC)

寄存器	地址	读写	描述	复位值
IICLC	0x54000010	R/W	IIC总线多主设备线路控制寄存器	0x00

IICLC	位	描述	初始值
Filter Enable	[2]	IIC总线滤波器使能位。 当SDA接口作为输入操作，该位应该是高电平。过滤器可以避免在连个PCLK期间干扰出现错误 0：滤波器无效 1：滤波器有效	0
SDA output delay	[1:0]	IIC总线SDA线路延时长度选择位 SDA线按以下时钟时间（PCLK）的延时 00： 0 clocks 01： 5 clocks 10： 10 clocks 11： 15 clocks	00

第二十一章 IIS 总线接口

21.1 概述

当前很多音频系统以CD的形式，数字音频带，数字音频处理器和数字TV音响，在市场上吸引消费者。S3C2440A的Inter-IC Sound (IIS)总线接口作为一个编解码接口连接外部 8/16 位立体声音频解码IC用于迷你碟机和可携式应用。IIS总线接口支持IIS总线数据格式和 MSB-justified数据格式。该接口对FIFO的访问采用了DMA模式取代了中断。它可以在同一时间接收和发送数据。

模块图

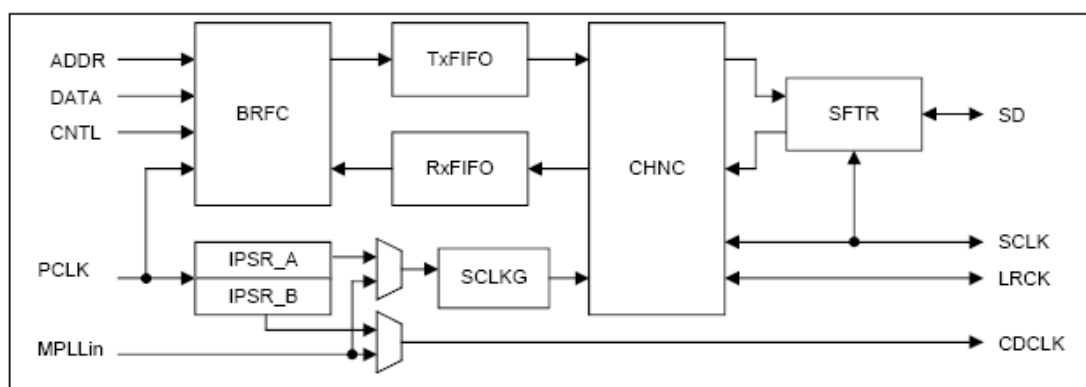


Figure 21-1. IIS-Bus Block Diagram

21.2 功能描述

总线接口，寄存器组 and 状态机 (BRFC)：总线接口逻辑和 FIFO 访问由状态机控制。

5 位双预定标器 (IPSR)：一个预定标器用于 IIS 总线接口的主时钟发生器，另外一个用作外部编解码时钟发生器。

64 位 FIFO (TxFIFO 和 RxFIFO)：在发送数据传输时，数据写到 TxFIFO；在接收数据传输时，从 RxFIFO 读取数据。

主 IISCLK 发生器 (SCLKG)：在主设备模式，串行位时钟是从主时钟生成。

通道发生器和状态机 (CHNC)：IISCLK 和 iislrck 是由通道状态机生成并控制。

15 位移位寄存器 (SFTR)：在发送模式下并行数据移位成串行数据输出，在接收模式下串行数据输入移位成并行数据。

21.3 发送接收模式

通常传输

IIS 控制寄存器对于发送接收 FIFO 有一个 FIFO 准备标志位。当 FIFO 准备发送数据时，如果 FIFO 非空，FIFO 准备标志位置 1。如果 FIFO 为空，FIFO 准备标志位置 0。当接收 FIFO 非满，对于接收 FIFO 的 FIFO 准备标志位置 1。其指出 FIFO 准备好接收数据。如果接收 FIFO 为满，FIFO 准备标志置 0。这些标志用于决定 CPU 读写 FIFO 的时间。用这种方法当 CPU 在访问发送接收 FIFO 时，串行数据能被发送和接收。

DMA 传输

在此模式下，发送或接收 FIFO 对 DMA 控制器是可访问的。在发送或接收模式下的 DMA 服务请求是由 FIFO 准备标志自动执行。

发送和接收模式

在此模式下 IIS 总线接口可以同时接收和发送数据。

21.4 音频串行接口格式

21.4.1 IIS 总线格式

IIS 总线有四线包括串行数据输入（IISDI），串行数据输出（IISDO），左右通道选择（IISLRCK）和串行位时钟（IISCLK）。生成 IISLRCK 和 IISCLK 的设备是主设备。

串行数据以 2 的补码发送，MSB(Most Significant Bit 最高位)先发。因为发送器和接收器可能有不同的字长，MSB（最高位）先发。发送器不必知道接收器可以处理多少位，接收器也不必知道会收到多少位。

当系统字长大于发生器的字长，字为了数据发送而被截断（最低位被置 0）。如果接收器接收大于其字长的位，在 LSB（最低位）后的位被忽略。另外，如果接收器收到的位数小于其字长，缺少的位被置 0。因此 MSB 有一个固定的位置，而 LSB 的位置取决于字长。只要 IISLRCK 发送改变，发送器在一个时钟周期内发送下一个字的 MSB。

由发送器发送的串行数据可以和时钟信号的下降沿和上升沿同步。但是，串行数据必须在串行时钟信号的上升沿锁存到接收器。因此当同步上升沿的数据发送时有一些限制。

左右通道选择线指出了正在发送的通道。IISLRCK 可以在串行时钟的下降沿或上升沿被改变，当时其不需要对称。在从设备，信号在串行时钟的下降沿或上升沿被锁存。在 MSB 被发送，IISLRCK 线改变一个时钟周期。此允许发送器导出用于建立发送的串行数据同步时序。此外，其使能接收存储前一个字且为下一个字的接收清除输入。

21.4.2 MSB JUSTIFIED 格式

MSB JUSTIFIED 总线格式在结构上和 IIS 总线格式。唯一和 IIS 总线格式的区别，MSB JUSTIFIED 格式实现了只要 IISLRCK 改变，发送器总是发送下一个字的 MSB。

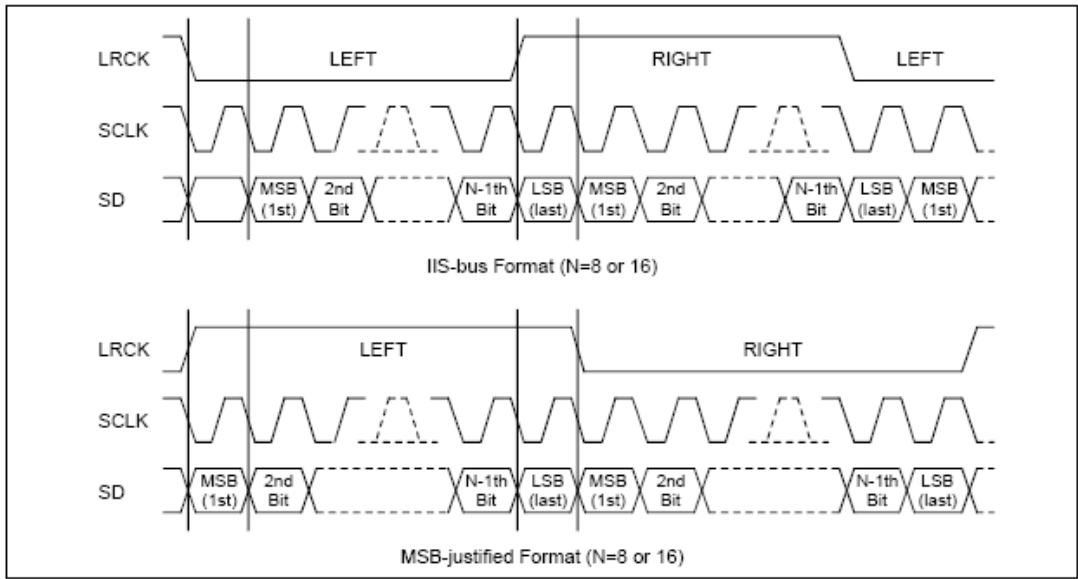


Figure 21-2. IIS-Bus and MSB (Left)-justified Data Interface Formats

21.4.3 采样频率和主设备时钟

主设备时钟频率（PCLK 或 MPLLIn）可以在如表 21-1 所示的采样频率中选择。因为主设备时钟由 IIS 预分频器（预定标器）产生，预分频器（预定标器）的值和主设备时钟类型（256 或 384fs）应该合适确定。

串行位时钟频率类型可以由每个通道的串行位和如表 21-2 所示的主设备时钟中来选择。

表 21-1CODEC 时钟（CODECLK=256 或 384fs）

IISLRCK (fs)	8.000 KHz	11.025 KHz	16.000 KHz	22.050 KHz	32.000 KHz	44.100 KHz	48.000 KHz	64.000 KHz	88.200 KHz	96.000 KHz
CODECLK (MHz)	256fs									
	2.0480	2.8224	4.0960	5.6448	8.1920	11.2896	12.2880	16.3840	22.5792	24.5760
	384fs									
	3.0720	4.2336	6.1440	8.4672	12.2880	16.9344	18.4320	24.5760	33.8688	36.8640

表 21-2 可用串行位时钟频率（IISCLK=16 或 32 或 48fs）

Serial bit per channel	8-bit	16-bit
Serial clock frequency (IISCLK)		
@CODECLK = 256fs	16fs, 32fs	32fs
@CODECLK = 384fs	16fs, 32fs, 48fs	32fs, 48fs

21.5 IIS 总线接口特殊寄存器

- (1) IIS 控制寄存器 (IISCON)
- (2) IIS 模式寄存器 (IISMOD)
- (3) IIS 预定标器寄存器 (IISPSR)
- (4) IIS FIFO 控制寄存器 (IISFCON)
- (5) IIS FIFO 寄存器 (IISFIFO)

21.5.1 IIS 控制寄存器

IIS CONTROL REGISTER (IISCON)

寄存器	地址	读写	描述	复位值
IISCON	0x55000000 (Li/HW, Li/W, Bi/W) 0x55000002 (Bi/HW)	R/W	IIS控制寄存器	0x100

IISCON	位	描述	初始值
Left/Right channel index (Read only)	[8]	0 = Left 1 = Right	1
Transmit FIFO ready flag (Read only)	[7]	0 = Empty 1 = Not empty	0
Receive FIFO ready flag (Read only)	[6]	0 = Full 1 = Not full	0
Transmit DMA service request	[5]	0 = Disable 1 = Enable	0
Receive DMA service request	[4]	0 = Disable 1 = Enable	0
Transmit channel idle command	[3]	在空闲模式下IISLRCK是非激活 (Pause Tx)。 0 = Not idle 1 = Idle	0
Receive channel idle command	[2]	在空闲模式下IISLRCK是非激活 (Pause Rx)。 0 = Not idle 1 = Idle	0
IIS prescaler	[1]	0 = Disable 1 = Enable	0
IIS interface	[0]	0 = Disable (stop) 1 = Enable (start)	0

注:

- (1) IISCON 寄存器对每个字节都是可访问的, 字单元在大小端模式下使用 STRB/STRH/STR 和 LDRB/LDRH/LDR 或 char/short int/int 类型指针。
- (2) (Li/HW/W): Little/HalfWord/Word
(Bi/HW/W): Big/HalfWord/Word

21.5.2 IIS 模式寄存器

IIS MODE REGISTER (IISMOD)

寄存器	地址	读写	描述	复位值
IISMOD	0x55000004 (Li/HW, Li/W, Bi/W) 0x55000006 (Bi/HW)	R/W	IIS模式寄存器	0x0

IISMOD	位	描述	初始值
Master Clock Select	[9]	主设备时钟选择 0 = PCLK 1 = MPLLIn	0
Master/slave mode select	[8]	0 = 主设备模式 (IISLRCK和IISCLK是输出模式)。 1 = 从设备模式 (IISLRCK和IISCLK是输入模式)。	0
Transmit/receive mode select	[7:6]	00 = No transfer 01 = Receive mode 10 = Transmit mode 11 = Transmit and receive mode	00
Active level of left/right channel	[5]	0 = Low for left channel (High for right channel) 1 = High for left channel (Low for right channel)	0
Serial interface format	[4]	0 = IIS compatible format 1 = MSB (Left)-justified format	0
Serial data bit per channel	[3]	0 = 8-bit 1 = 16-bit	0
Master clock frequency select	[2]	0 = 256fs 1 = 384fs (fs: sampling frequency)	0
Serial bit clock frequency select	[1:0]	00 = 16fs 01 = 32fs 10 = 48fs 11 = N/A	0

21.5.3 IIS 预定标器寄存器

IIS PRESCALER REGISTER (IISPSR)

寄存器	地址	读写	描述	复位值
IISPSR	0x55000008 (Li/HW, Li/W, Bi/W) 0x5500000A (Bi/HW)	R/W	IIS预定标器寄存器	0x0

IISPSR	位	描述	初始值
Prescaler control A	[9:5]	数据值: 0 ~ 31 注: 预定标器A生成用于内部模块的主设备时钟且除数因子是N+1。	00000
Prescaler control B	[4:0]	数据值: 0 ~ 31 注: 预定标器B生成用于外部模块的主设备时钟且除数因子是N+1。	00000

21.5.4 IIS FIFO 控制寄存器

IIS FIFO CONTROL REGISTER (IISFCON)

寄存器	地址	读写	描述	复位值
IISFCON	0x5500000C (Li/HW, Li/W, Bi/W) 0x5500000E (Bi/HW)	R/W	IIS FIFO控制寄存器	0x0

IISFCON	位	描述	初始值
Transmit FIFO access mode select	[15]	0 = Normal 1 = DMA	0
Receive FIFO access mode select	[14]	0 = Normal 1 = DMA	0
Transmit FIFO	[13]	0 = Disable 1 = Enable 0	0
Receive FIFO	[12]	0 = Disable 1 = Enable 0	0
Transmit FIFO data count (Read only)	[11:6]	Data count value = 0 ~ 32	000000
Receive FIFO data count (Read only)	[5:0]	Data count value = 0 ~ 32	000000

21.5.5 IIS FIFO 寄存器

IIS FIFO REGISTER (IISFIFO)

IIS 总线接口包含两个 64 位的 FIFO 用于发送和接收模式。每个 FIFO 有 16 宽 32 长的表，其允许 FIFO 不管有效数据大小对每个半字单元操作数据。接收和发送 FIFO 访问通过 FIFO 入口进行，进入地址是 0x55000010。

寄存器	地址	读写	描述	复位值
IISFIFO	0x55000010 (Li/HW, Li/W, Bi/W) 0x55000012 (Bi/HW)	R/W	IIS FIFO寄存器	0x0

IISFIFO	位	描述	初始值
FENTRY	[15:0]	对于IIS的发送/接收数据	0x0

第二十二章 SPI（串行外围设备接口）

22.1 概述

S3C2440A 的 SPI 接口可以接口串行数据传输。S3C2440A 包括两个 SPI 接口，每个接口分别有两个 8 位的数据移位器用于发送和接收。在 SPI 发送期间，数据同时发送（串行移出）和接收（串行移入）。在某个频率下的 8 位串行数据由相应的控制寄存器设置决定。如果你仅想发送，接收数据可以保持缄默。另外如果你只想接收，你应该发送缄默数据 1。

22.2 特点

- 支持两个通道的 SPI
- 兼容 SPI 协议（2.11 版本）
- 8 位发送移位寄存器
- 8 位接收移位寄存器
- 8 位预定标器
- 查询，中断和 DMA 传输模式
- 容忍 5V 输入，除 nSS

22.3 模块图

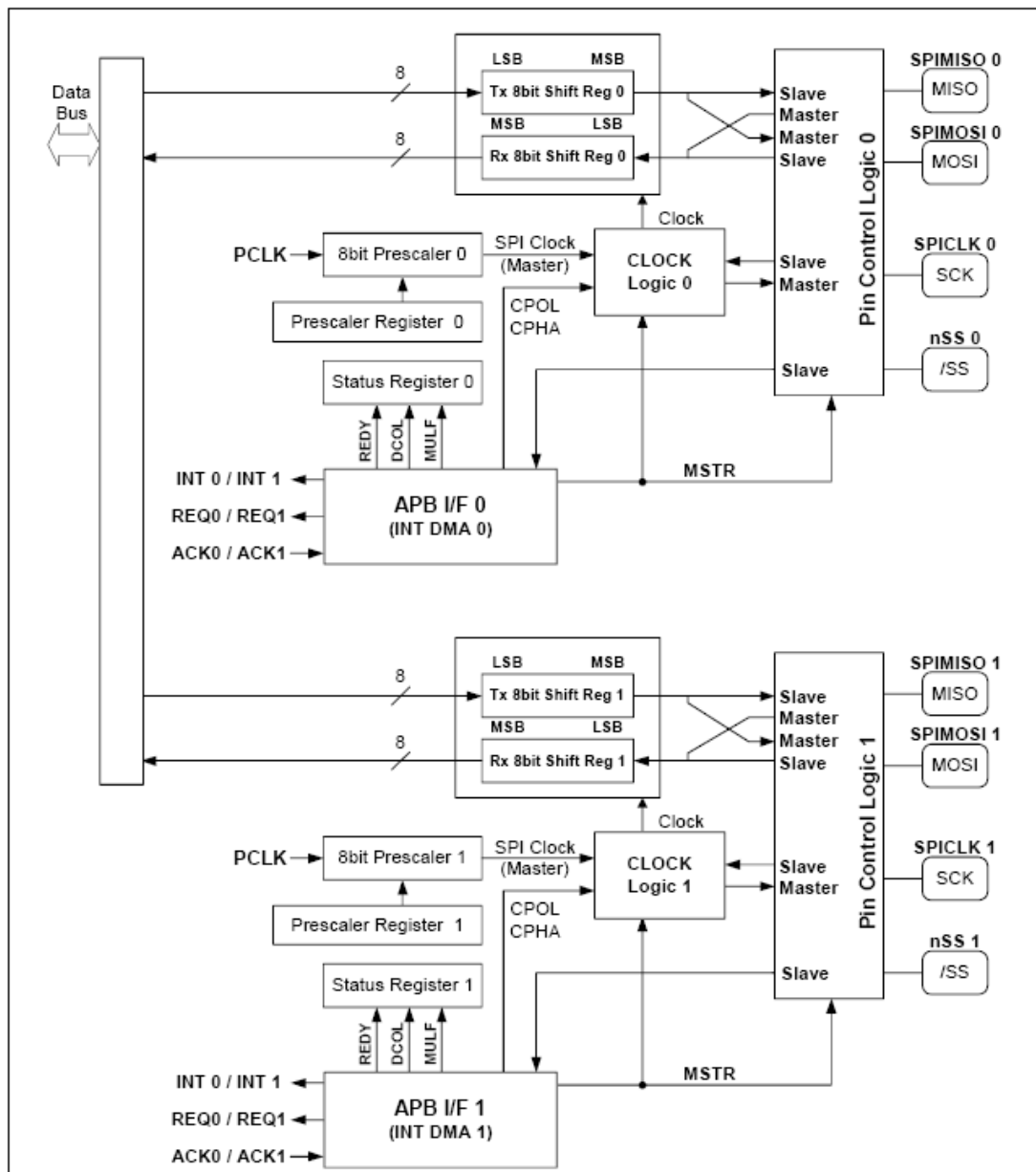


Figure 22-1 SPI Block Diagram

22.4 SPI 操作

使用 SPI 接口, S3C2440A 可以和外部设备接收发送 8 位数据。一个串行时钟线来同步两个用于信息移位和采用的数据线。当 SPI 是主机时, 传输频率通过设定 $SPPREN$ 寄存器的相应位来控制。你可以修改其频率来调节波特率数据寄存器的值。如果 SPI 是从属, 其他的主机提供时钟。当程序员写字节数据到 $SPTDATn$ 寄存器时, SPI 发送接收操作会同时开始。在这种情况下, 在写字节数据到 $SPTDATn$ 之前, nSS 应该被激活。

22.4.1 编程步骤

当一个字节的数据写入 SPTDATn 寄存器, 如果 ENSCK、SPCONn 寄存器的 MSTR 被置位, SPI 开始发送。你可以使用一个典型的编程步骤来操作 SPI 卡。

对 SPI 模块编程, 按照一下基本步骤

- (1) 时钟波特率预定标器寄存器 (SPPREN)
- (2) 设置 SPCONn 来合理配置 SPI 模块
- (3) 写数据 0xFF 到 SPTDATn 10 次, 目的是初始化 MMC 或 SD 卡。
- (4) 设置一个 GPIO 引脚, 其作为 nSS, 低电平是激活 MMC 或 SD 卡。
- (5) 发送数据->检查传输准备标志 (REDY=1) 的状态, 然后写数据到 SPTDATn
- (6) 接收数据 (1): SPCONn 的 TAGD 位是无效=normal mode
->写 0xFF 到 SPTDATn, 然后确认 REDY 置 1, 然后从读缓存读取数据
- (7) 接收数据 (2): SPCONn 的 TAGD 位是有效=Tx Auto Garbage Data mode
->确认 REDY 置 1, 然后从读缓存读取数据 (然后自动开始传输)
- (8) 设置一个 GPIO 引脚, 其作为 nSS, 高电平是解除激活 MMC 或 SD 卡。

22.4.2 SPI 传输格式

S3C2440A 支持 4 种不同格式来传输数据。如图 22-2 所示对于 SPICLK 的四种波形。

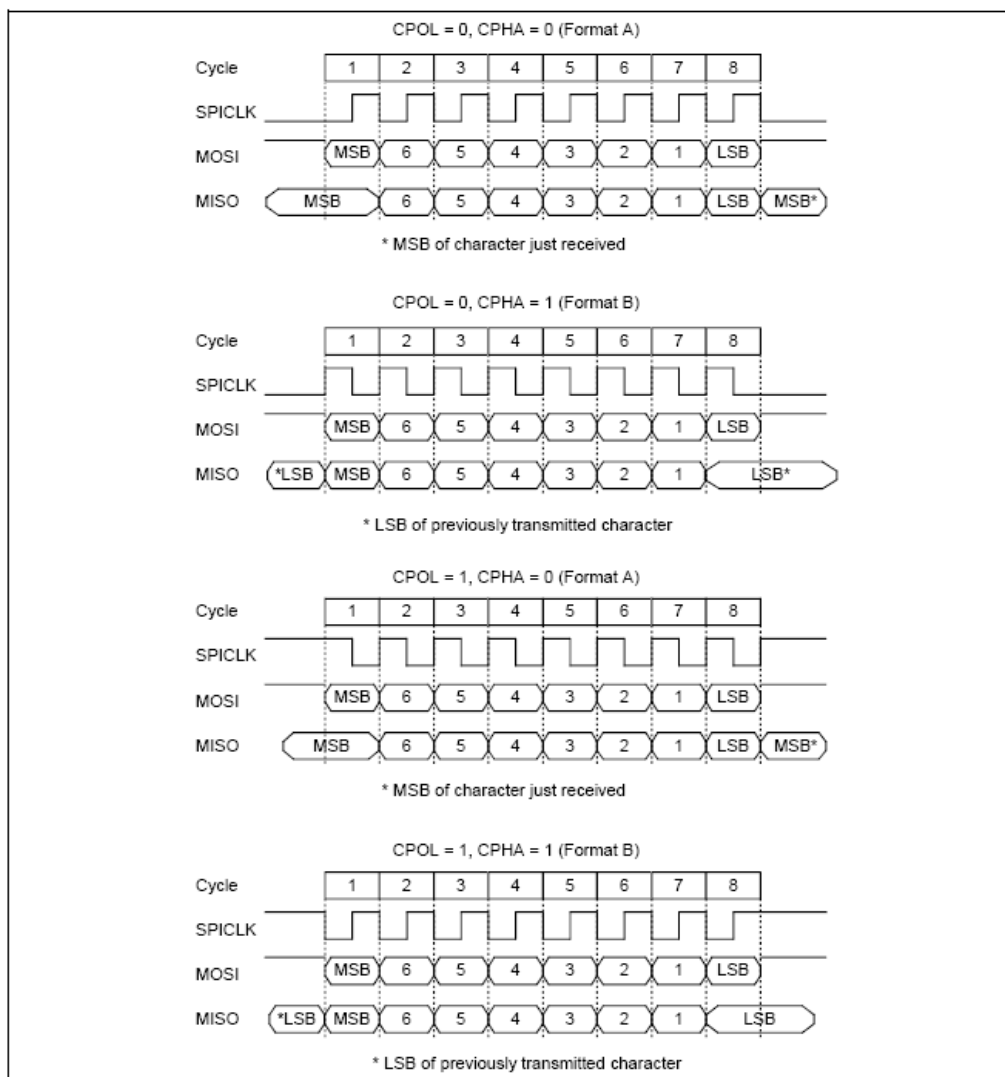


Figure 22-2 SPI Transfer Format

22.4.3 对于 DMA 的发送步骤

- (1) SPI 配置为 DMA 模式
- (2) DMA 作相应配置
- (3) SPI 请求 DMA 服务
- (4) DMA 发送 1 个字节数据到 SPI
- (5) SPI 发送数据到卡
- (6) 返回步骤 3 直到 DMA 计数器为 0
- (7) SPI 配置 SMOD 位为中断或查询模式

22.4.4 对于 DMA 的接收步骤

- (1) SPI 配置为有 SMOD 位的 DMA 开始和 TAGD 位置 1
- (2) DMA 做相应的配置
- (3) SPI 从卡接收 1 个字节的数据
- (4) SPI 请求 DMA 服务
- (5) DMA 从 SPI 接收数据
- (6) 知道写数据 0xFF 到 SPTDATn
- (7) 返回步骤 4 直到 DMA 计数为 0
- (8) SPI 配置为有 SMOD 位的查询模式和清除 TAGD 位
- (9) 如果 SPSTAn 的 READY 标志置位, 则读最后一个字节数据

注: 总的接收数据=DMA TC 值+查询模式下的最后数据 (步骤 9)
第一个 DMA 接收数据是无效的, 用户可以忽略掉。

22.5 SPI 特殊寄存器

- (1) SPI 控制寄存器 (SPCONn)
- (2) SPI 状态寄存器 (SPSTAn)
- (3) SPI 引脚控制寄存器 (SPPINn)
- (4) SPI 波特率预定标器寄存器 (SPPREn)
- (5) SPI 发送数据寄存器 (SPTDATn)
- (6) SPI 接收数据寄存器 (SPRDATn)

22.5.1 SPI 控制寄存器

SPI CONTROL REGISTER (SPCONn)

寄存器	地址	读写	描述	复位值
SPCON0	0x59000000	R/W	SPI通道 0 控制寄存器	0x00
SPCON1	0x59000020	R/W	SPI通道 1 控制寄存器	0x00

SPCONn	位	描述	初始值
SPI Mode Select (SMOD)	[6:5]	决定SPTDAT如何读写 00 = 查询模式 01 = 中断模式 10 = DMA模式 11 = reserved	00
SCK Enable (ENSCK)	[4]	决定SCK是否使能 (仅对主机) 0 = 无效 1 = 有效	0
Master/Slave Select (MSTR)	[3]	决定主从模式 0 = 从 1 = 主 Note: 在从模式下, 应该留有时间给主机初始化发送接收	0
Clock Polarity Select (CPOL)	[2]	决定高态有效或低态有效时钟 0 = 高态有效 1 = 低态有效	0
Clock Phase Select (CPHA)	[1]	从两个基本不同的传输格式中选一 0 = 格式 A 1 = 格式 B	0
Tx Auto Garbage Data mode enable (TAGD)	[0]	决定是否需要正在接收的数据 0 = normal mode 1 = Tx auto garbage data mode 注: 在正常模式下, 如果你仅想接收数据, 你可以改传输无效数据 0xFF	0

22.5.2 SPI 状态寄存器

SPI STATUS REGISTER (SPSTAn)

寄存器	地址	读写	描述	复位值
SPSTA0	0x59000004	R/W	SPI通道 0 状态寄存器	0x01
SPSTA1	0x59000024	R/W	SPI通道 1 状态寄存器	0x01

SPCONn	位	描述	初始值
保留	[7:3]		
Data Collision Error Flag (DCOL)	[2]	如果当传输在过程中且通过读SPSTAn清除, 写SPTDAT或读SPRDATn, 则该标志置位。 0: 不检测 1: 冲突错误检测	0
Multi Master Error Flag (MULF)	[1]	如果当SPI配置为主机时nSS信号为低态有效, 该标志置位。 0: 不检测 1: 多主机错误检测	0
Transfer Ready Flag (REDY)	[0]	该位是指SPTDATn或SPRDATn准备发送或接收。写数据到SPTDATn该位自动清除。 0: 不准备 1: 数据接收发送准备	1

22.5.3 SPI 引脚控制寄存器

SPI PIN CONTROL REGISTER (SPPINn)

当 SPI 系统使能, 除 nSS 外引脚的方向由 SPCONn 寄存器的 MSTR 位控制。nSS 的方向始终为输入。

当 SPI 为主机时, nSS 引脚用于检测多主机错误, 提供 SPPIN 的 ENMUL 位是激活的, 其他 GPIO 应该用于选择从设备。

如果 SPI 配置为从设备, nSS 引脚用来选择 SPI 为另一个主设备的从设备。

寄存器	地址	读写	描述	复位值
SPPIN0	0x59000008	R/W	SPI通道 0 引脚控制寄存器	0x00
SPPIN1	0x59000028	R/W	SPI通道 1 引脚控制寄存器	0x00

SPPINn	位	描述	初始值
保留	[7:3]		
Multi Master error detect Enable (ENMUL)	[2]	当SPI系统是主设备, nSS引脚用作输入来侦测多主机错误 0: 无效 (通用目的) 1: 多主机错误侦测使能	0
保留	[1]	保留	0
Master Out Keep (KEEP)	[0]	决定MOSI驱动或当 1 个字节发送完成时释放 (仅对主机) 0: 释放 1: 驱动先前电平	1

SPIMISO (MISO) 和 SPIMOSI (MOSI) 数据引脚是用来接收和发送串行数据。当 SPI 配置为主设备, SPIMISO (MISO) 是主数据输入线, SPIMOSI (MOSI) 是主数据输出线, SPICLK (SCK) 是时钟输出线。当 SPI 为从设备, 这些引脚翻转角色。在一个多主机系统中, SPICLK (SCK)、SPIMISO (MISO) 和 SPIMOSI (MOSI) 引脚分别捆绑配置成组。当另一个 SPI 设备工作在主设备状态并选择 S3C2440A SPI 为从设备, 一个主 SPI 将经历多主机错误。当错误被检测时, 接下来的措施被立即执行。但是如果你想检测这个错误, 你必须预先设置 SPPINn 的 ENMUL 位。

(1) 在从模式下 SPCONn 的 MSTR 位被强制设置为 0 来操作

(2) SPSTAn 的 MULF 标志置位, 且产生 SPI 中断。

22.5.4 SPI 波特率预定标器寄存器

SPI BAUD RATE PRESCALER REGISTER (SPPREn)

寄存器	地址	读写	描述	复位值
SPPRE0	0x5900000C	R/W	SPI通道 0 波特率预定标器寄存器	0x00
SPPRE1	0x5900002C	R/W	SPI通道 1 波特率预定标器寄存器	0x00

SPPREn	位	描述	初始值
Prescaler Value	[7:0]	决定SPI时钟率 波特率 = PCLK / 2 / (Prescaler的值+1)	0x00

注: 波特率应该小于 25MHz。

22.5.5 SPI 发送数据寄存器

SPI TX DATA REGISTER (SPTDATn)

寄存器	地址	读写	描述	复位值
SPTDAT0	0x59000010	R/W	SPI通道 0 发送数据寄存器	0x00
SPTDAT1	0x59000030	R/W	SPI通道 1 发送数据寄存器	0x00

SPTDATn	位	描述	初始值
Tx Data Register	[7:0]	该区域包含通过SPI通道发送的数据	0x00

22.5.6 SPI 接收数据寄存器

SPI RX DATA REGISTER (SPRDATn)

寄存器	地址	读写	描述	复位值
SPRDAT0	0x59000014	R/W	SPI通道 0 接收数据寄存器	0xFF
SPRDAT1	0x59000034	R/W	SPI通道 1 接收数据寄存器	0xFF

SPRDATn	位	描述	初始值
Rx Data Register	[7:0]	该区域包含通过SPI通道接收到的数据	0xFF

第二十四章 AC97 控制器

24.1 概述

S3C2440A 的 AC97 控制器单元支持 AC97 的 2.0 版本特点。AC97 控制器使用一个音频控制器连接（AC-link）来和 AC97 编解码器通讯。控制器发送立体声 PCM 数据给编解码器。编解码器中的外部数模转换器转换音频采样到模拟音频波形。控制器也从编解码器接收立体声 PCM 数据或单声道的 MIC 数据，然后将数据存储在内存中。本章描述的是 AC97 控制器单元的编程模式。该章的信息需要了解 AC97 的 2.0 版本特性。

注：AC97 控制器和 IIS 控制器不能同时使用。

24.2 特点

- 对于立体声 PCM 输入，立体声 PCM 输出和单声道 MIC 输入都有独立的通道。
- 基于 DMA 操作和基于中断的操作
- 所有通道都仅支持 16 位采样
- 不同采样率的 AC97 编解码器接口（48KHz 及以下）
- 16 位，每个通道 16 个入口 FIFO
- 仅支持主编解码器

24.3 AC97 控制器操作

24.3.1 模块图

如图 24-1 所示 S3C2440A 的 AC97 控制器功能模块图。来自 AC-link 的 AC97 信号，其连接是支持全双向数据传输的点对点的同步串行互联。所有数字音频流和命令状态信息通过 AC-link 通讯。

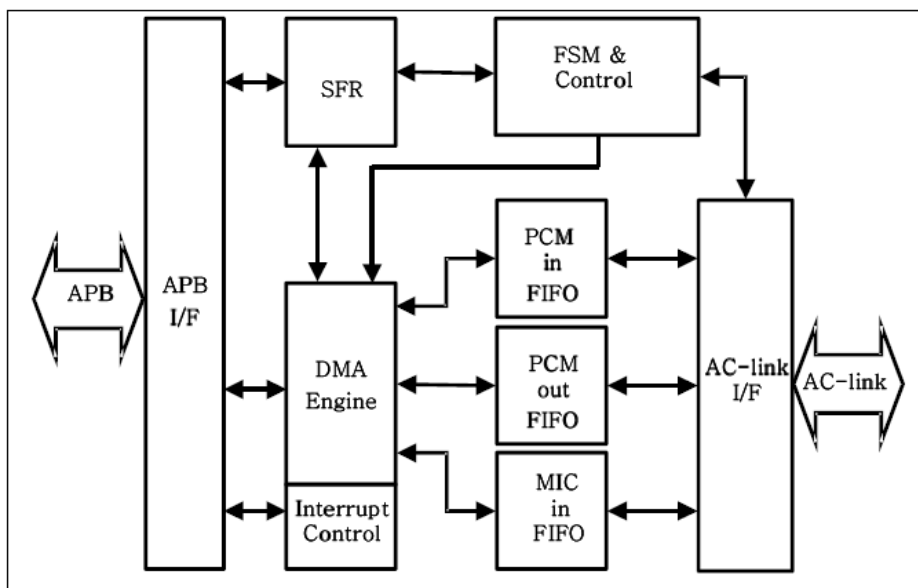


Figure 24-1 AC97 Block Diagram

24.3.2 内部数据通路

如图 24-2 所示 S3C2440A 的 AC97 控制器的内部数据通路。其有立体声脉冲编码调制 (PCM) 输入, 立体声 PCM 输出和单声道 MIC 输入缓存, 其包含 16 位, 16 个入口缓存。其有一个通过 AC-link 的 20 位 IO 移位寄存器。

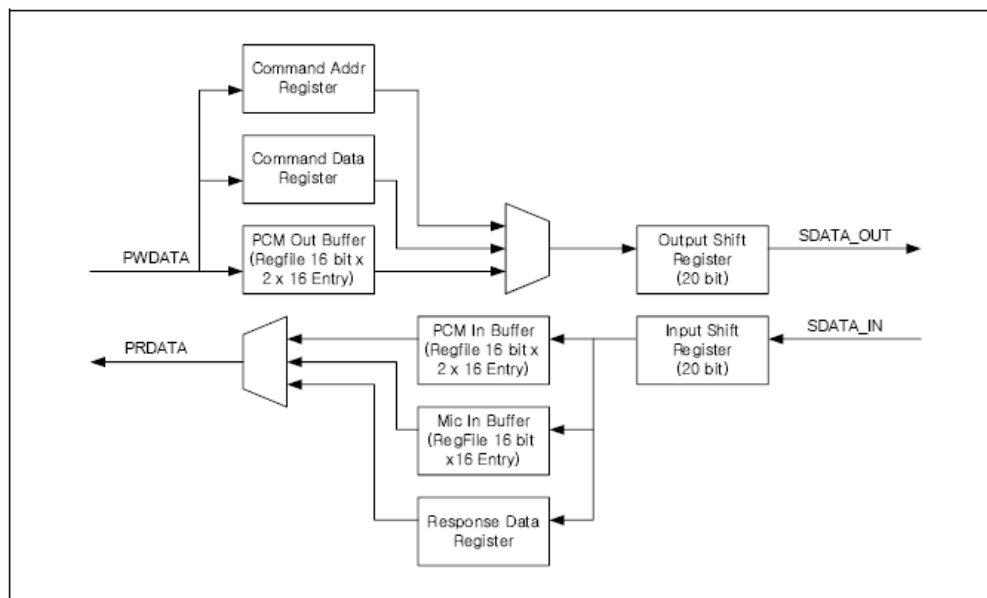


Figure 24-2 Internal Data Path

24.3.3 流程图操作

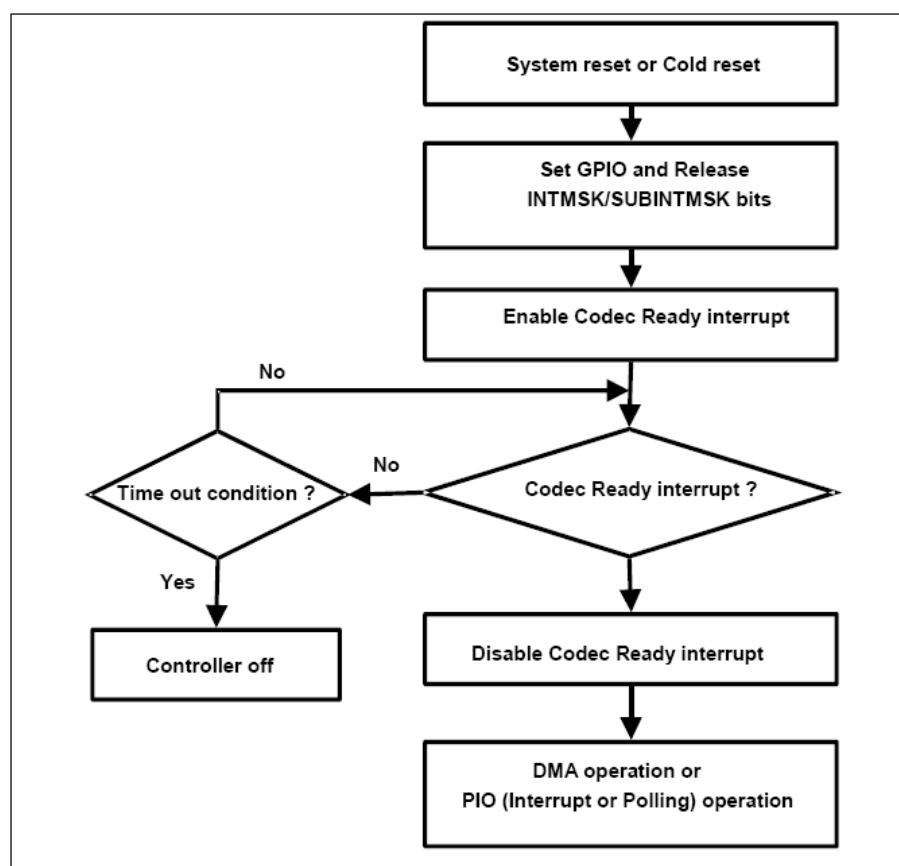


Figure 24-3 AC97 Operation Flow Chart

24.4 AC-LINK 数字接口协议

每个 AC97 编解码包含一个连接到 AC97 控制器的五个引脚的数字串行接口。AC-link 是一个全双工，固定时钟，PCM 数字流。其有一个时分多路器配置来操作控制寄存器访问和多路输入输出音频流。AC-link 架构将每个音频帧分成 12 个输出和 12 个输入数据流。每个流有一个 20 位的采样分辨率和需要最小分辨率 16 位的一个 DAC 和一个 ADC。

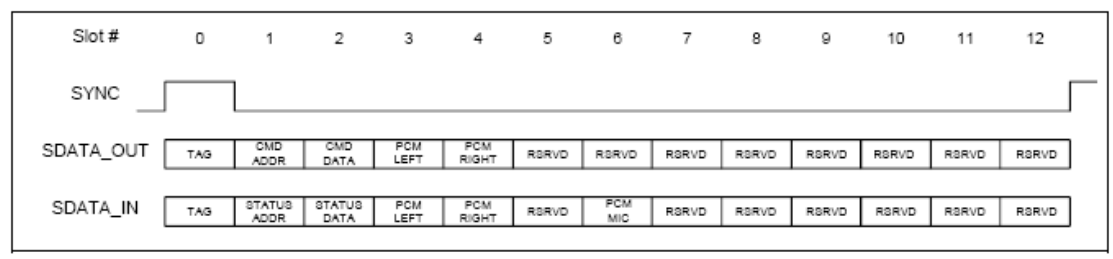


Figure 24-4 Bi-directional AC-link Frame with Slot Assignments

如图 24-2 所示 AC97 控制器所支持的时间槽定义。AC97 控制提供了对在 AC-link 上的所有数据处理的同步。

一个数据处理由 256 位的信息组成，其信息分解成 13 组时间槽并称为帧。时间槽 0 叫标签段（Tag Phase）且有 16 位长。剩下的 12 个时间槽叫做数据段。标签段包含 1 位用于识别有效帧，另外 12 位用于识别数据段中的时间槽是否包含有效数据。数据段中的每个时间槽是 20 位长。一个帧开始由 SYNC 信号变高电平。SYNC 高电平时间就是相应的标签段所占时间。

AC97 帧以固定 48KHz 的时间间隔出现且同步于 12.288MHz 比特率时钟 BITCLK。控制器和编解码器使用 SYNC 和 BITCLK 来决定何时发送数据，何时采样和接收数据。发送器在每个 BITCLK 的上升沿发送串行数据流，接收器在每个 BITCLK 的下降沿采样串行数据流。发送器必须对串行数据流中的有效槽做标记。有效槽被标记在时间槽 0 中。AC-link 的数据是从 MSB 到 LSB。标签段的第一位是位 15，每个数据段的第一位是位 19。每个槽的最后一位是位 0。

AC-link 输出帧（SDATA_OUT）

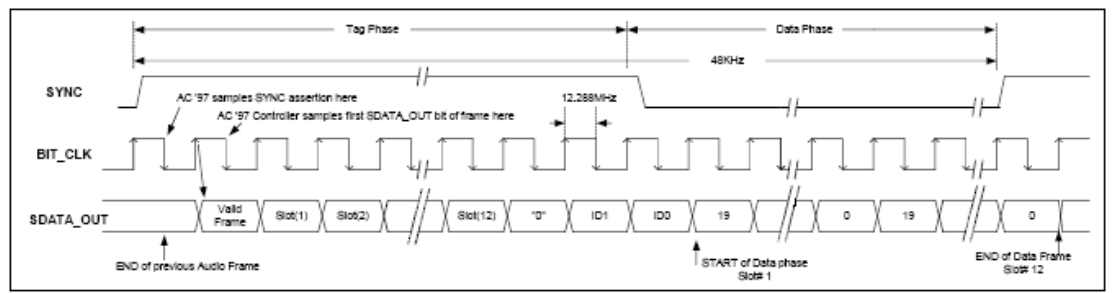


Figure 24-5 AC-link Output Frame

AC-link 输入帧 (SDATA_IN)

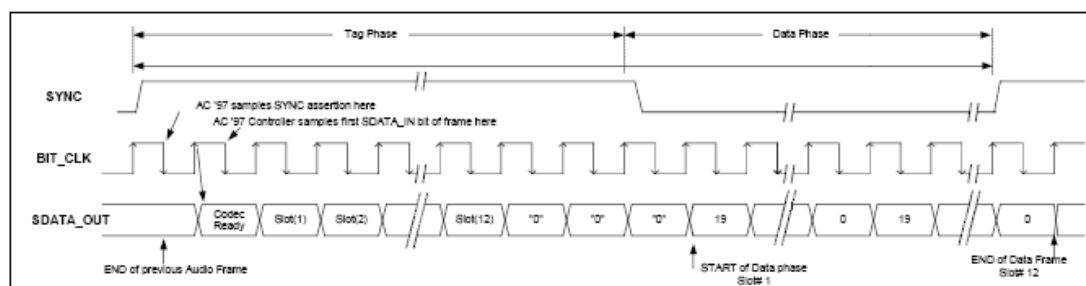


Figure 24-6 AC-link Input Frame

24.5 AC97 掉电

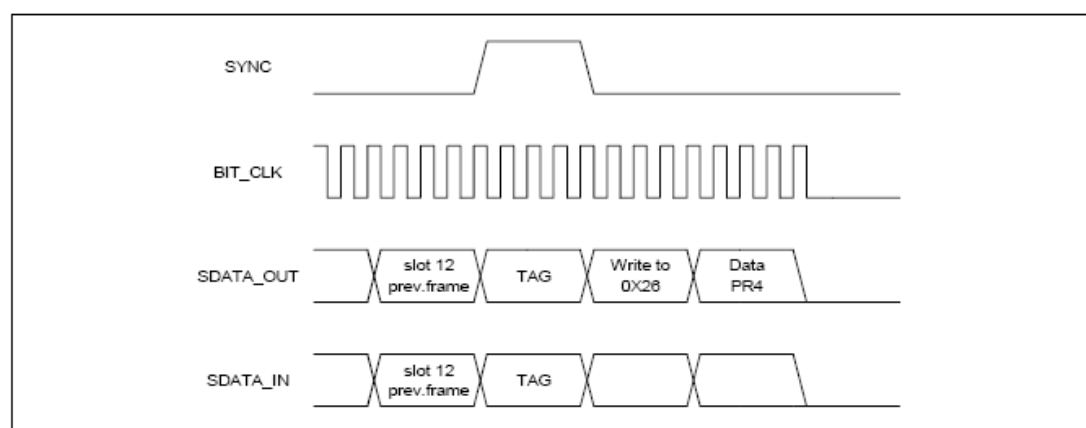


Figure 24-7 AC97 Powerdown Timing Diagram

24.5.1 掉电 AC-link

当 AC97 编解码器的掉电寄存器 (0x26) 的 PR4 位置 1 时, AC-link 信号进入低电源模式。然后主编解码器驱使 BITCLK 和 SDATA_IN 为逻辑低电压水平。顺序如时序图 24-7 所示。

AC97 控制器通过 AC-link 发送写掉电寄存器 (0x26)。建立 AC97 控制器以至于当其写掉电寄存器的 PR4 位 (数据 0x1000), 就不发送数据到槽 3-12, 当它收到掉电请求后, 就不需要编解码器去处理其他数据。当编解码器处理请求时, 它同时会拉低 BITCLK 和 SDATA_IN。

在对 AC_GLBCTRL 寄存器编程后, AC97 控制器也驱使 SYNC 和 SDATA_OUT 为低电平。

24.5.2 唤醒 AC-link- 由 AC97 控制器触发的唤醒

AC-link 协议提供了一个 AC97 冷重启和一个 AC97 热重启。当前掉电状态最后指出了会使用哪个 AC97 重启。在所有的掉电模式期间所有的寄存器都应该停留在同一状态, 除非执行一个 AC97 冷重启。在 AC97 冷重启中, AC97 寄存器被初始化到默认值。在掉电后, 在其通过重新使 SYNC 有效而重新激活之前, 掉电出现的帧之后, AC-link 必须等待最少 4 个音频帧时间。当 AC-link 上电, 其通过编解码器准备位 (输入槽 0, 位 15) 指示就绪。

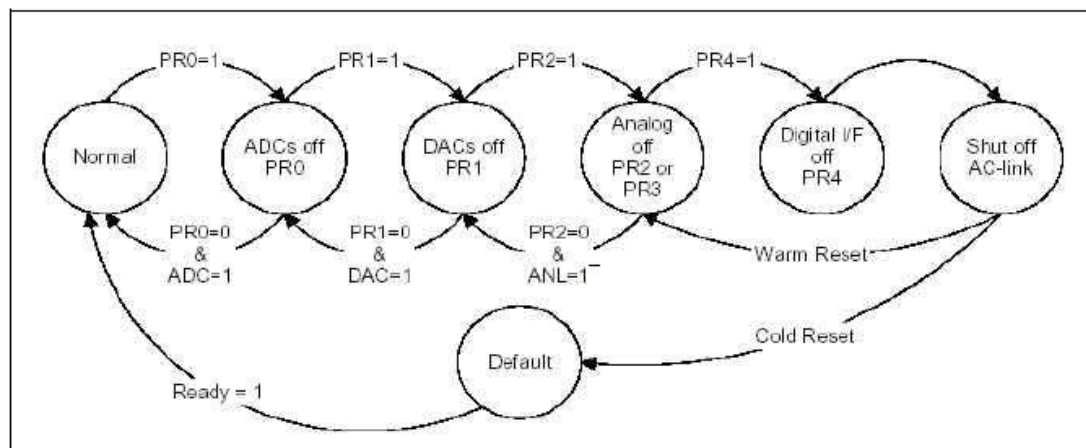


Figure 24-8 AC97 Power down/Power up Flow

AC97 冷重启

当通过 AC_GLBCTRL 寄存器使得 nRESET 引脚有效，冷启动产生。激活和使无效 nRESET 将激活 BITCLK 和 SDATA_OUT。所有的 AC97 控制寄存器都被初始化到默认上电复位值。nRESET 是一个同步 AC97 输入。

AC97 热重启

不改变当前的 AC97 寄存器值，AC97 热重启重新激活 AC-link。当没有 BITCLK 信号且 SYNC 拉高时，热启动产生。在通常的音频帧中，SYNC 是一个同步 AC97 输入。当缺少 BITCLK 时，SYNC 是作为一个用于产生 AC97 热重启的异步输入。AC97 控制器必须不激活 BITCLK 直到其采样到 SYNC 再次为低电平。这样避免了误认为一个新音频帧。

24.6 AC97 控制器特殊寄存器

- (1) AC97 全局控制寄存器 (AC_GLBCTRL)
- (2) AC97 全局状态寄存器 (AC_GLBSTAT)
- (3) AC97 CODEC 命令寄存器 (AC_CODEC_CMD)
- (4) AC97 CODEC 状态寄存器 (AC_CODEC_STAT)
- (5) AC97 PCM 输入输出通道 FIFO 地址寄存器 (AC_PCMADDR)
- (6) AC97 MIC 输入通道 FIFO 地址寄存器 (AC_MICADDR)
- (7) AC97 PCM 输入输出通道 FIFO 数据寄存器 (AC_PCMDATA)
- (8) AC97 MIC 输入通道 FIFO 数据寄存器 (AC_MICDATA)

24.6.1 AC97 全局控制寄存器

AC97 GLOBAL CONTROL REGISTER (AC_GLBCTRL)

寄存器	地址	读写	描述	复位值
AC_GLBCTRL	0x5B000000	R/W	AC97 全局控制寄存器	0x00000000

AC_GLBCTRL	位	描述	初始值
保留	[31:23]	-	0x00
Codec Ready Interrupt Enable	[22]	0: 无效 1: 有效	0
PCM Out Channel Underrun Interrupt Enable	[21]	0: 无效 1: 有效 (FIFO空)	0
PCM In Channel Overrun Interrupt Enable	[20]	0: 无效 1: 有效 (FIFO满)	0
MIC In Channel Overrun Interrupt Enable	[19]	0: 无效 1: 有效 (FIFO满)	0
PCM Out Channel Threshold Interrupt Enable	[18]	0: 无效 1: 有效 (FIFO半空)	0
PCM In Channel Threshold Interrupt Enable	[17]	0: 无效 1: 有效 (FIFO半满)	0
MIC In Channel Threshold Interrupt Enable	[16]	0: 无效 1: 有效 (FIFO半满)	0
保留	[15:14]	-	00
PCM Out Channel Transfer Mode	[13:12]	00 : Off 01 : PIO 10 : DMA 11 : Reserved	00
PCM In Channel Transfer Mode	[11:10]	00 : Off 01 : PIO 10 : DMA 11 : Reserved	00
MIC In Channel Transfer Mode	[9:8]	00 : Off 01 : PIO 10 : DMA 11 : Reserved	00
保留	[7:4]	-	0000
Transfer Data Enable Using AC-Link	[3]	0: 无效 1: 有效	0
AC-Link On	[2]	0: 关闭 1: 传输SYNC到CODEC	0
Warm Reset	[1]	0: 通常 1: 从掉电唤醒CODEC	0
Cold Reset	[0]	0: 通常 1: 重启CODEC和控制逻辑	0

24.6.2 AC97 全局状态寄存器

AC97 GLOBAL STATUS REGISTER (AC_GLBSTAT)

寄存器	地址	读写	描述	复位值
AC_GLBSTAT	0x5B000004	R/W	AC97 全局状态寄存器	0x00000000

AC_GLBSTAT	位	描述	初始值
保留	[31:23]	-	0x00
Codec Ready Interrupt	[22]	0: 不请求 1: 请求	0
PCM Out Channel Underrun Interrupt	[21]	0: 不请求 1: 请求	0
PCM In Channel Overrun Interrupt	[20]	0: 不请求 1: 请求	0
MIC In Channel Overrun Interrupt	[19]	0: 不请求 1: 请求	0
PCM Out Channel Threshold Interrupt	[18]	0: 不请求 1: 请求	0
PCM In Channel Threshold Interrupt	[17]	0: 不请求 1: 请求	0
MIC In Channel Threshold Interrupt	[16]	0: 不请求 1: 请求	0
保留	[15:3]	-	0x000
Controller Main State	[2:0]	000 : Idle 001 : Init 010 : Ready 011 : Active 100 : LP 101 : Warm	000

24.6.3 AC97 CODEC 命令寄存器

AC97 CODEC COMMAND REGISTER (AC_CODEC_CMD)

寄存器	地址	读写	描述	复位值
AC_CODEC_CMD	0x5B000008	R/W	AC97CODEC命令寄存器	0x00000000

AC_CODEC_CMD	位	描述	初始值
保留	[31:24]	-	0x00
Read Enable	[23]	0: 命令写 (注) 1: 状态读	0
Address	[22:16]	CODEC命令地址	0x00
Data	[15:0]	CODEC命令数据	0x0000

注：当命令写入 AC_CODEC_CMD 寄存器中，推荐连个命令之间的延时大于 1/48Hz。

24.6.4 AC97 CODEC 状态寄存器

AC97 CODEC STATUS REGISTER (AC_CODEC_STAT)

寄存器	地址	读写	描述	复位值
AC_CODEC_STAT	0x5B00000C	R/W	AC97CODEC状态寄存器	0x00000000

AC_CODEC_STAT	位	描述	初始值
保留	[31:23]	-	0x00
Address	[22:16]	CODEC状态地址	0x00
Data	[15:0]	CODEC状态数据	0x0000

注：如果你想通过 AC_CODEC_STAT 寄存器从 AC97CODEC 读数据，你应该依照以下步骤。

- (1) 写命令地址和数据到 AC_CODEC_CMD 寄存器，其 23 位置 1。
- (2) 延时。
- (3) 从 AC_CODEC_STAT 寄存器读命令地址和数据。

24.6.5 AC97 PCM 输入输出通道 FIFO 地址寄存器

AC97 PCM OUT/IN CHANNEL FIFO ADDRESS REGISTER (AC_PCMADDR)

寄存器	地址	读写	描述	复位值
AC_PCMADDR	0x5B000014	R/W	AC97 PCM输入输出通道FIFO地址寄存器	0x00000000

AC_PCMADDR	位	描述	初始值
保留	[31:28]	-	0000
Out Read Address	[27:24]	PCM输出通道FIFO读地址	0000
保留	[23:20]	-	0000
In Read Address	[19:16]	PCM输入通道FIFO读地址	0000
保留	[15:12]	-	0000
Out Write Address	[11:8]	MIC输出通道FIFO读地址	0000
保留	[7:4]	-	0000
In Write Address	[3:0]	MIC输入通道FIFO写地址	0000

24.6.6 AC97 MIC 输入通道 FIFO 地址寄存器

AC97 MIC IN CHANNEL FIFO ADDRESS REGISTER (AC_MICADDR)

寄存器	地址	读写	描述	复位值
AC_MICADDR	0x5B000014	R/W	AC97 MIC输入通道FIFO地址寄存器	0x00000000

AC_MICADDR	位	描述	初始值
保留	[31:20]	-	
Read Address	[19:16]	MIC输入通道FIFO读地址	0x0000
保留	[15:4]	-	
Write Address	[3:0]	MIC输入通道FIFO写地址	0x0000

24.6.7 AC97 PCM 输入输出通道 FIFO 数据寄存器

AC97 PCM OUT/IN CHANNEL FIFO DATA REGISTER (AC_PCMDATA)

寄存器	地址	读写	描述	复位值
AC_PCMDATA	0x5B000018	R/W	AC97 PCM输入输出通道FIFO数据寄存器	0x00000000

AC_PCMDATA	位	描述	初始值
Left Data	[31:16]	左声道PCM输出输入FIFO数据 读: 左声道中的PCM 写: 左声道中的PCM	0x0000
Right Data	[15:0]	右声道PCM输出输入FIFO数据 读: 右声道中的PCM 写: 右声道中的PCM	0x0000

24.6.8 AC97 MIC 输入通道 FIFO 数据寄存器

AC97 MIC IN CHANNEL FIFO DATA REGISTER (AC_MICDATA)

寄存器	地址	读写	描述	复位值
AC_MICDATA	0x5B00001C	R/W	AC97 MIC输入通道FIFO数据寄存器	0x00000000

AC_MICDATA	位	描述	初始值
保留	[31:16]	-	0x0000
Mono Data	[15:0]	MIC在单声道FIFO的数据	0x0000