

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。

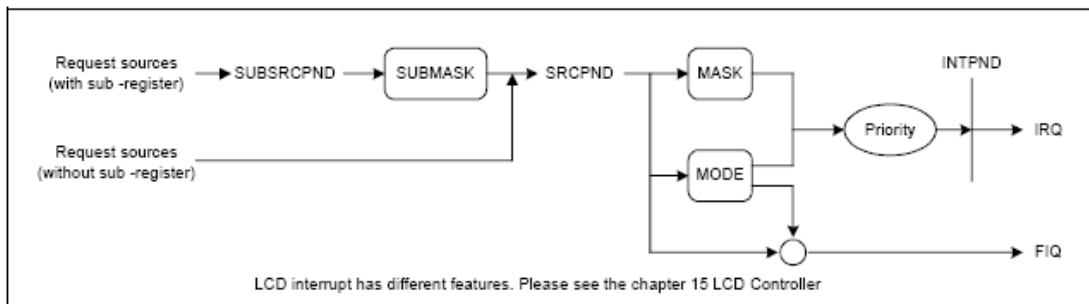


Figure 14-1. Interrupt Process Diagram

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

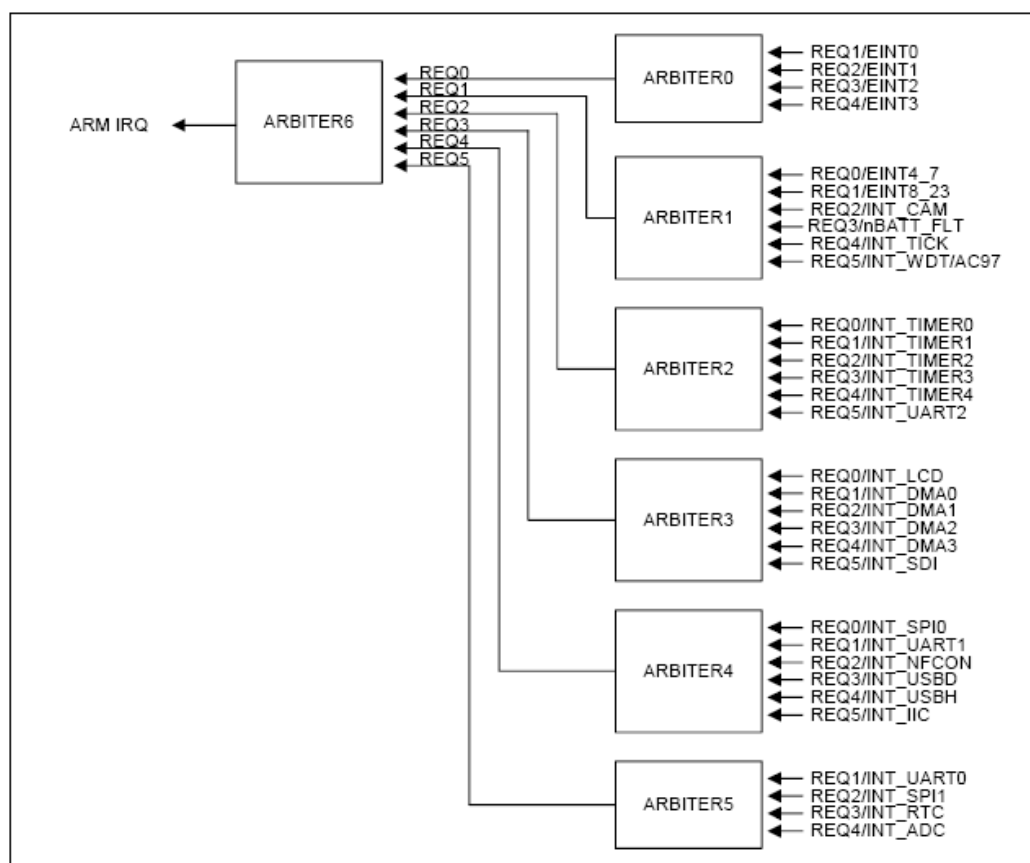


Figure 14.2. Priority Generating Block

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  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