嵌入式系统安全与设计

庄连生

Email: { *lszhuang@ustc.edu.cn* }

***Fall 2015*，USTC**

University of Science and Technology of China

第7讲 中断控制

内容提要：

- 中断体系结构
- 中断控制器
- 中断分发处理

第7讲 中断控制

内容提要：

- 中断体系结构
- 中断控制器
- 中断分发处理

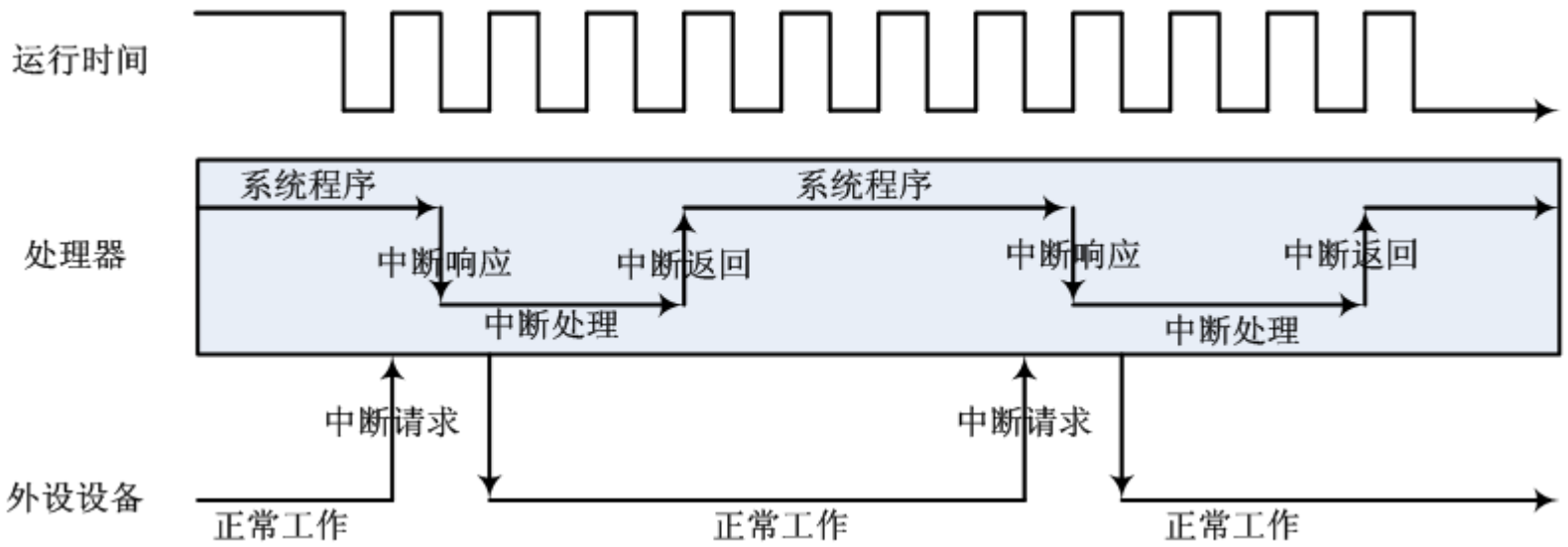
中断概述

- **中断定义：**CPU在正常执行程序的过程中，突然发生了一些需要紧急处理的事件，这些事件通过某种方式触发引起CPU暂停当前正在执行的程序，转去处理突发事件，待突发事件处理完毕后，CPU再返回继续执行刚刚被暂停的程序的过程就称之为中断；
- **中断的主要作用包括：**
  - ✓ CPU与外部设备并行工作
  - ✓ 能够处理例外事件
  - ✓ 实现实时处理
  - ✓ 实现用户程序与操作系统的联系
  - ✓ 实现多任务并行执行

  ……

中断作用

- 如：CPU与一台外部设备并行工作的时间关系图：



中断概述

- **CPU处理异常事件的方式：**
  - ✓ 查询方式：程序循环查询各个设备的状态并作出相应的反应；
  - ✓ 中断方式：当某个事件发生时，硬件会自动设置某个寄存器；当CPU执行完一个指令时，如果所关注的事件发生了，硬件会使CPU中止当前程序流程，跳转到某个固定物理地址处理此事件，处理完毕后，返回到被中止的程序，使之继续执行。

中断概述

- 引起中断的原因或者说发出中断请求的来源叫做**中断源**。根据中断源的不同，可以把中断分为**硬件中断**和**软件中断**两大类。
- **硬件中断**是由处理器自身的硬件系统或着外部硬件部件触发引起的中断类型，硬件中断又可以分为外部中断和内部中断两类：
  - ✓ **外部中断**一般是指由外设发出的中断请求，如：按键中断、串口中断、定时器中断等。外部中断是可以屏蔽的中断，也就是说，利用中断控制器可以屏蔽这些外部设备的中断请求，S3C2440共有24个外部中断EINT0-EINT24。
  - ✓ **内部中断**是指因硬件出错（如突然掉电、奇偶校验错等）或运算出错（除数为零、运算溢出、单步中断等）所引起的中断，内部中断是不可屏蔽的中断，S3C2440共有35个内部中断。

中断概述

- **软件中断**与硬件中断不一样，它不是由硬件触发的，而是由软件调用系统提供的一个指令(SWI)来触发的中断行为，是程序本身有意识的引发的中断行为，由于在用户模式下，程序对处理器资源的访问会受到一些限制，因此，需要采用软中断的方式将处理器工作模式切到异常模式下访问受限资源，因此软中断一般都被用于实现特殊系统调用功能提供给软件系统使用。

中断概述

- 中断的处理过程：
  - ✓ **中断控制器**汇集各类中断源发出的中断信号，然后通知CPU。
  - ✓ CPU保存当前程序的运行环境(各个寄存器、局部变量等)，自动跳转到固定物理地址，调用中断服务程序来处理中断。
  - ✓ 在中断服务程序中通过读取中断控制器、中断源相关的寄存器来辨别是哪个中断，并调用相应的中断服务函数进行处理。
  - ✓ 清除中断：通过读写中断控制器和中断源相关寄存器来实现。
  - ✓ 恢复被中断程序的运行环境(即第二步保存的寄存器和局部变量等)，继续执行。

中断控制器

- 当中断发生时，中断请求并不是直接发送给CPU，而是发送给中断控制器（中断控制器需要用户进行初始化），中断控制器进行裁决后，选择出当前最需要处理的中断请求发送给CPU，这样可以降低CPU的负担。
- **中断类型**：S3C2440中共包含59个中断源，分为32个一级中断源和35个二级中断源。其中：
  - ✓ 在32个一级中断源中，有24个一级中断源可以直接触发中断，剩下8个是带有二级中断源的，本身不能触发，只计其二级中断源；
  - ✓ 35个二级中断源触发时需要通过一级中断源进行请求，在处理中断时再通过寄存器判断是哪个二级中断源触发。
  - ✓ 8个带有二级中断源需要做二次判断，效率相对较低。

中断控制器

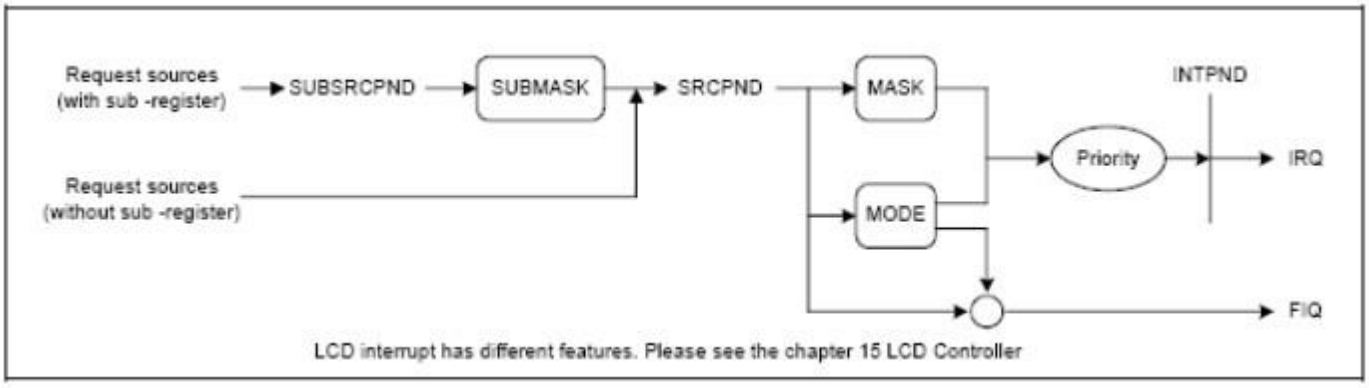
中断汇总表			
中断类型	一级中断/ SRCPND	二级中断/ SUBSRCPND 或 EINTPND	功能说明
内部中断	INT_ADC/[31]	INT_ADC_S/SUBSRCPND[10]	ADC 转换完成中断
		INT_TC/SUBSRCPND[9]	触摸屏中断(按下/提起)
	INT_RTC/[30]		RTC 闹钟中断
	INT_SPI1/[29]		SPI1 中断
	INT_UART0/[28]	INT_ERR0/SUBSRCPND[2]	错误中断
		INT_TXD0/SUBSRCPND[1]	数据发送中断
		INT_RXD0/SUBSRCPND[0]	数据接收中断
	INT_IIC/[27]		IIC 中断
	INT_USBH/[26]		USB 主机中断
	INT_USBD/[25]		USB 设备中断
	INT_NFCON/[24]		Nand Flash 控制中断
	INT_UART1/[23]	INT_ERR1/SUBSRCPND[5]	错误中断
		INT_TXD1/SUBSRCPND[4]	数据发送中断
		INT_RXD1/SUBSRCPND[3]	数据接收中断

内部中断	INT_SPI0/[22]		SPI0 中断
	INT_SDI/[21]		SDI 中断
	INT_DMA3/[20]		DMA 通道 3 中断
	INT_DMA2/[19]		DMA 通道 2 中断
	INT_DMA1/[18]		DMA 通道 1 中断
	INT_DMA0/[17]		DMA 通道 0 中断
	INT_LCD/[16]		LCD 中断
	INT_UART2/[15]	INT_ERR2/SUBSRCPND[8]	UART2 错误中断
		INT_TXD2/SUBSRCPND[7]	UART2 数据发送中断
		INT_RXD2/SUBSRCPND[6]	UART2 数据接收中断
	INT_TIMER4/[14]		定时器 4 中断
	INT_TIMER3/[13]		定时器 3 中断
	INT_TIMER2/[12]		定时器 2 中断
	INT_TIMER1/[11]		定时器 1 中断
	INT_TIMER0/[10]		定时器 0 中断
	INT_WDT_AC97/[9]	INT_AC97/SUBSRCPND[14]	AC97 中断
		INT_WDT/SUBSRCPND[13]	看门狗中断
	INT_TICK/[8]		RTC 时钟滴答中断
	nBATT_FLT/[7]		电池故障中断
	INT_CAM/[6]	INT_CAM_P/SUBSRCPND[12]	摄像头 P 端口捕捉中断
		INT_CAM_C/SUBSRCPND[11]	摄像头 C 端口捕捉中断

外部中断	EINT8_23/[5]	EINT23/EINTPND[23]	外部中断 23
		EINT22/EINTPND[22]	外部中断 22
		EINT21/EINTPND[21]	外部中断 21
		EINT20/EINTPND[20]	外部中断 20
		EINT19/EINTPND[19]	外部中断 19
		EINT18/EINTPND[18]	外部中断 18
		EINT17/EINTPND[17]	外部中断 17
		EINT16/EINTPND[16]	外部中断 16
		EINT15/EINTPND[15]	外部中断 15
		EINT14/EINTPND[14]	外部中断 14
		EINT13/EINTPND[13]	外部中断 13
		EINT12/EINTPND[12]	外部中断 12
		EINT11/EINTPND[11]	外部中断 11
		EINT10/EINTPND[10]	外部中断 10
		EINT9/EINTPND[9]	外部中断 9
		EINT8/EINTPND[8]	外部中断 8
	EINT4_7/[4]	EINT7/EINTPND[7]	外部中断 7
		EINT6/EINTPND[6]	外部中断 6
		EINT5/EINTPND[5]	外部中断 5
		EINT4/EINTPND[4]	外部中断 4
	EINT3/[3]		外部中断 3
	EINT2/[2]		外部中断 2
	EINT1/[1]		外部中断 1
	EINT0/[0]		外部中断 0

中断控制器

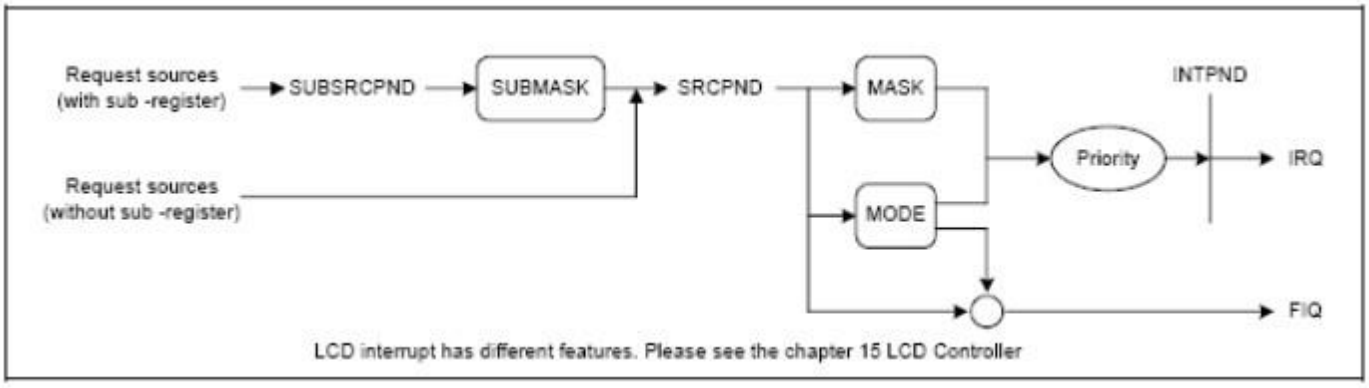
- **S3C2440中断控制器结构：**



① request sources（without sub-register）中的中断源被触发后，SRCPND寄存器中相应位被置1，如果此中断没有被INTMSK寄存器屏蔽或者快速中断的话，它将被进一步处理；

中断控制器

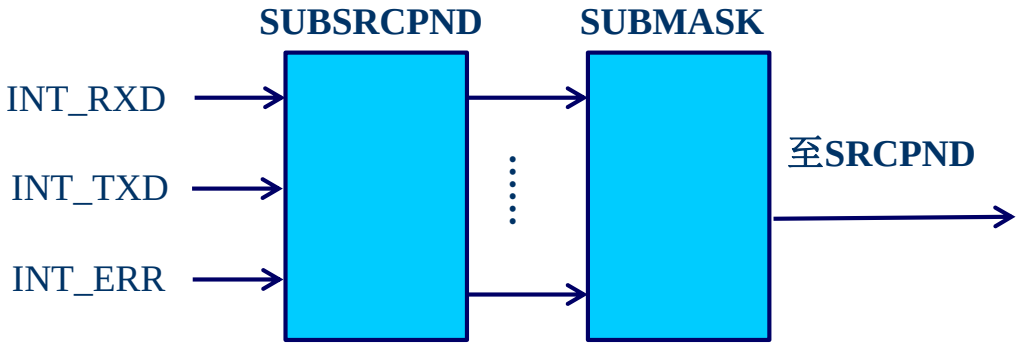
- **S3C2440中断控制器结构：**



② 对于request sources（with sub-register）中的中断源被触发后，SUBSRCPEND寄存器中的相应位被置1，如果此中断没有被INTSUBMSK寄存器屏蔽的话，它在 SRCPND寄存器中的相应位也被置1，以后的处理过程就和①步骤类似；

中断控制器

- **例子：**串口UART，UART中断包括发送中断、接收中断和因错误而产生的中断，但是这三个中断都属于一个大类，即都属于UART中断。下图显示了这类中断处理流程：

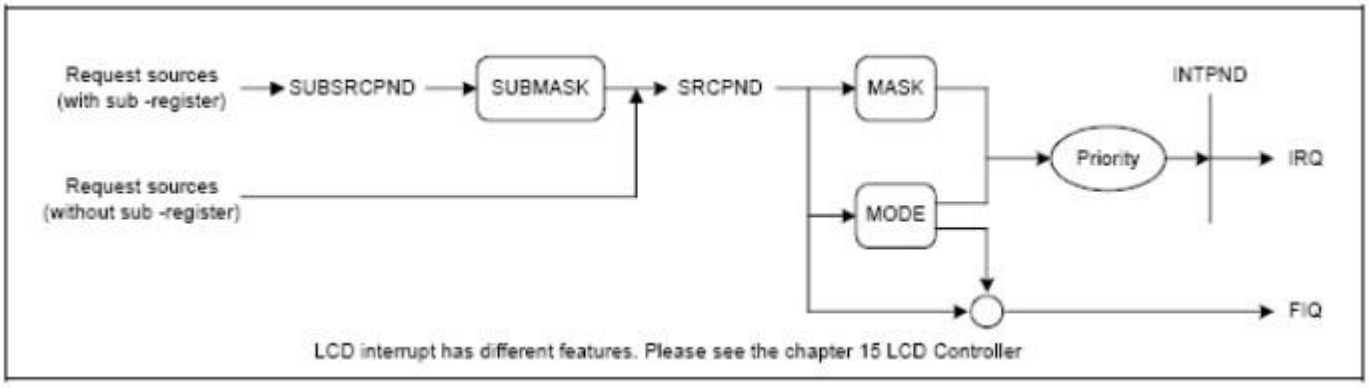


子中断的处理流程

SRCPND存储的是所有大类的中断，具体这一大类中断中哪种类型的子中断发生了，需要再查询SUBSRCPND寄存器才能得到。

中断控制器

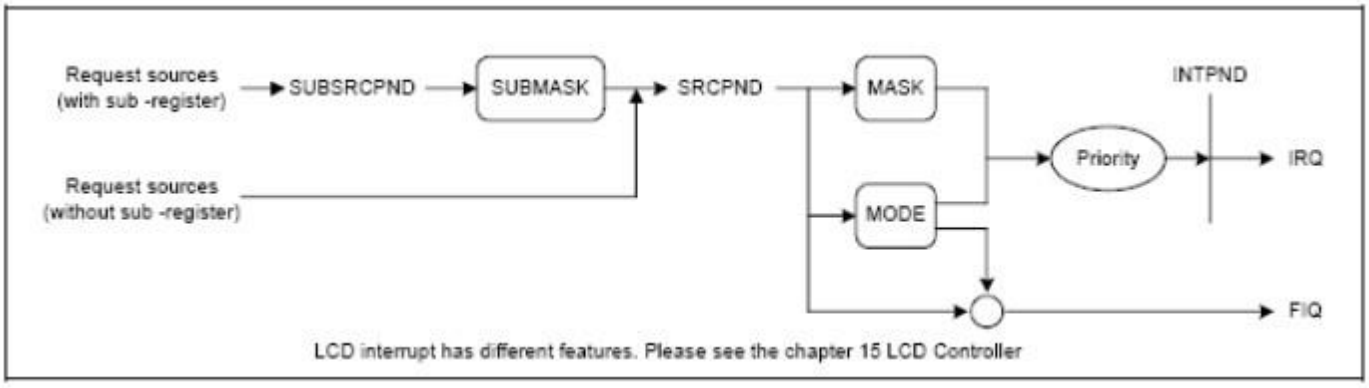
- **S3C2440中断控制器结构：**



③ 如果被触发的中断中有快速中断，INTMOD寄存器中为1的位对应的中断是FIQ，则CPU进入快速中断模式进行处理；

中断控制器

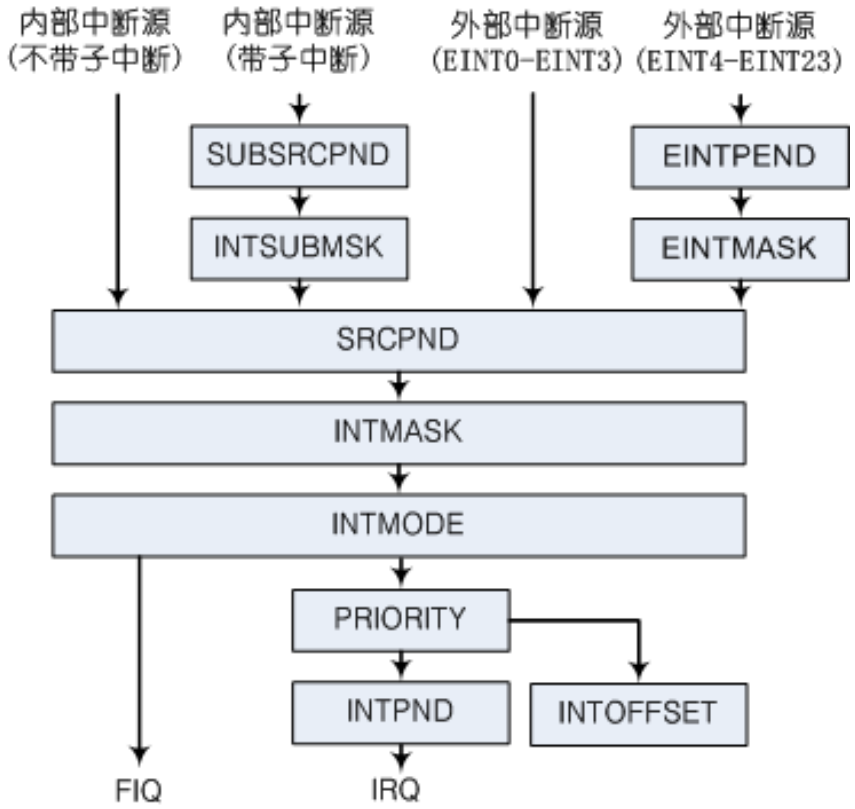
- **S3C2440中断控制器结构：**



④ 对于一般的中断IRQ，可能同时有几个中断被触发，未被INTMSK寄存器屏蔽的中断经过比较后，选出优先级最高的中断，此中断在INTPND寄存器中的 相应位被置1，然后CPU进入中断模式进行处理。中断服务程序通过读取INTPND或者INTOFFSET来确定中断源；

中断的产生

- 中断的产生过程在硬件的反映上就是几个寄存状态变化。
- 中断产生流程图示：

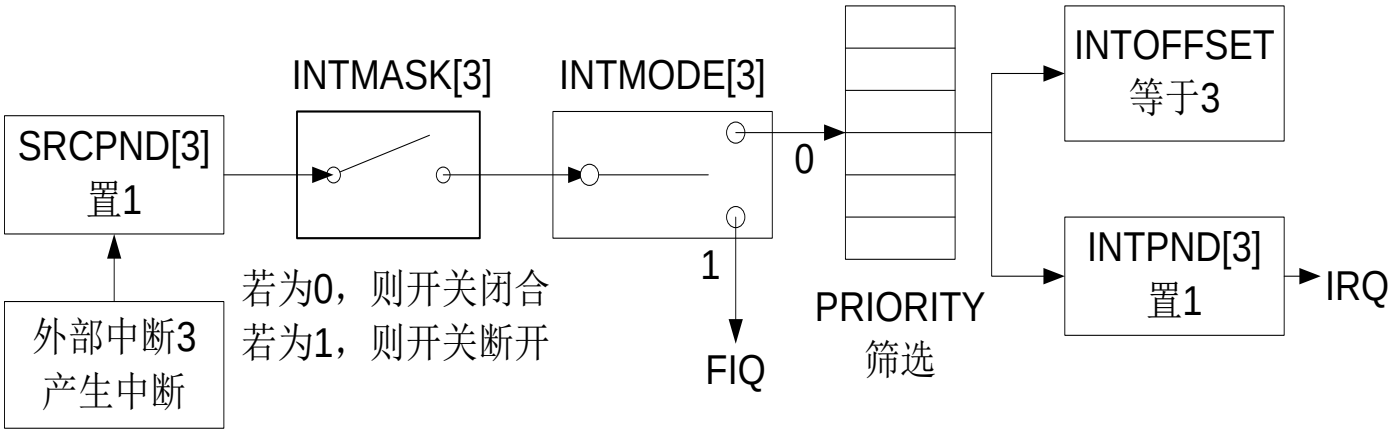


中断的产生

- 一级中断产生流程：
  - ① 设置源中断未决寄存器的标志位(SRCPND)
  - ② 判断中断位是否处于屏蔽状态(INTMASK)
    - 如果中断位已经被屏蔽，则等待中断蔽位打开，中断暂停。
  - ③ 判断中断模式是属于IRQ或着FIQ
    - 如果是FIQ中断，则判断程序状态寄存中F位是否处理屏蔽状态
    - 如果程序状态寄存器的F位没有被屏蔽，则CPU暂停当前工作，转而处理FIQ中断。
    - 如果程序状态寄存器的F位已经被屏蔽，中断暂停。
  - ④ 如果是IRQ中断，则判断程序状态寄存中I位是否处理屏蔽状态
    - 如果中断位已经被屏蔽，则等待中断屏蔽位打开，中断暂停。
  - ⑤ 从源中断未决寄存器中选择一个优先级最高的中断
  - ⑥ 设置中断未决寄存器(INTPND)
  - ⑦ 设置中断偏移寄存器(INTOFFSET)
  - ⑧ CPU暂停当前工作，转而处理IRQ中断。

中断的产生

- **例如：**外部中断3发生中断（对应的I/O口已初始化为外部中断）

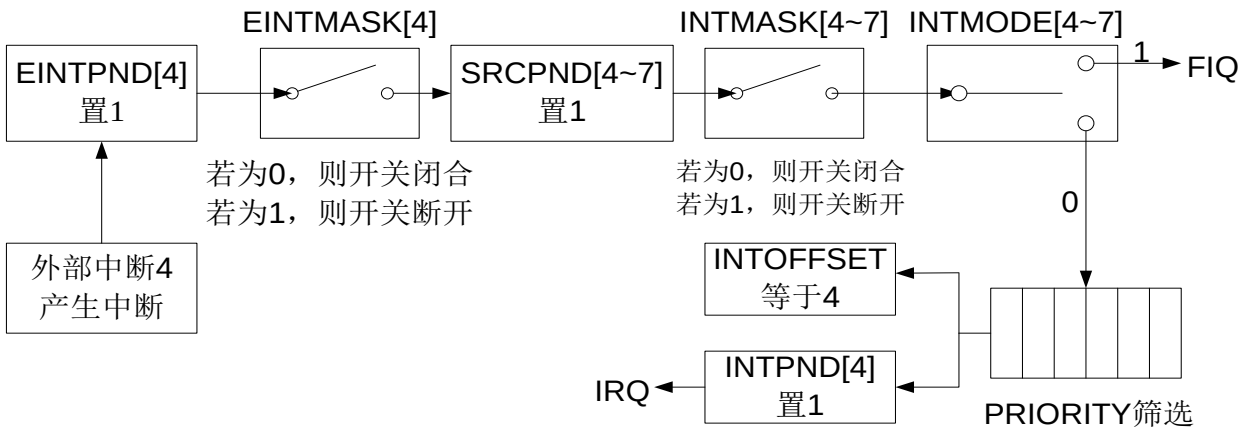


中断的产生

- 二级中断产生流程：
  1. 硬件触发产生中断
  2. 设置源中断未决寄存器的标志位(SUBSRCPND,EINTPEND)
  3. 判断中断位是否处于屏蔽状态(INTSUBMSK,EINTMASK)
     - 如果中断位已经被屏蔽，则等待中断屏蔽位打开，中断暂停。
  4. 设置相应的一级源中断未决寄存器的标志位
  5. 执行一级中断产生流程

中断的产生

- **例如：**外部中断4发生中断（对应的I/O口已初始化为外部中断）

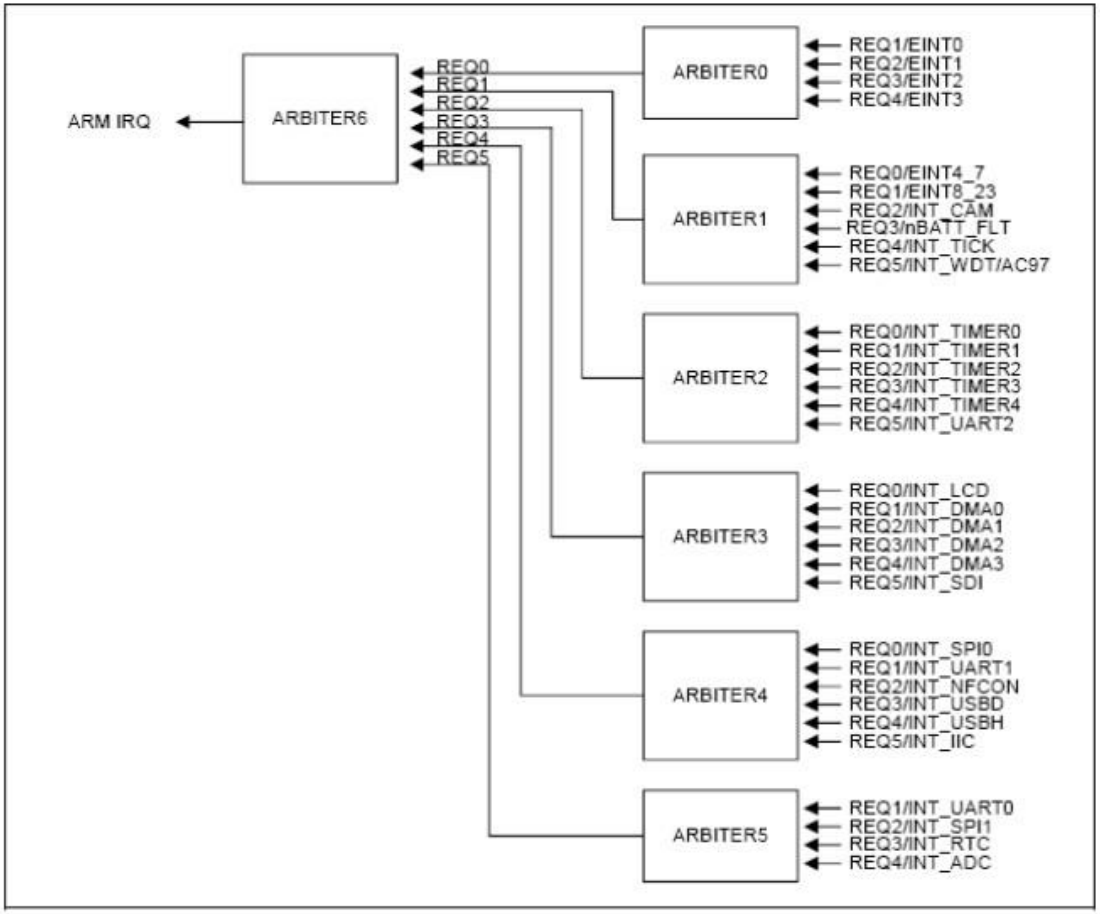


中断优先级

- 32 个一级中断请求优先级逻辑通过7个仲裁器的选择，最终生成一个优先级最高的中断源，这7个仲裁器包括：6 个一级仲裁器和1个二级仲裁器。
- **优先级选择过程：**先由一级仲裁器从各个分组中各选择一个优先级最高的中断，形成6个中断源，再通过二级仲裁器从这6个中断源中选择一个优先级最高的中断源进行触发。

中断优先级

- 仲裁过程如下图所示：



中断优先级

- **优先仲裁器：**由寄存器PRIORITY进行配置，其功能如下：
  - ✓ 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。

中断优先级

- 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。

中断寄存器

- 中断控制由以下几个寄存器来实现：
  - ✓ 外部中断未决寄存器（EINTPEND）
  - ✓ 外部中断掩码寄存器（EINTMASK）
  - ✓ 内部子中断未决寄存器（SUBSRCPND）
  - ✓ 内部子中断掩码寄存器（SUBINTMSK）
  - ✓ 中断源未决寄存器（SRCPND）
  - ✓ 中断掩码寄存器（INTMASK）
  - ✓ 中断模式寄存器（INTMOD）
  - ✓ 中断优先级寄存器（PRIORITY）
  - ✓ 中断未决寄存器（INTPND）
  - ✓ 中断偏移寄存器（INTOFFSET）
  - ✓ 外部中断的触发方式寄存器（EXTINT0-EXTINT2）
  - ✓ 外部中断控制滤波时钟和滤波宽度寄存器（EINTFLT0-EINTFLT3）

中断寄存器

SRCPND 源中断未决寄存器

- 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。
- **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。
- 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。

寄存器	地址	读/写	描述	复位值
SRCPND	0x4A000000	读写	指出中断请求的状态 0=中断未请求;1=中断已请求	0x00000000

中断寄存器

SRCPND 源中断未决寄存器

- ✓ 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。

SRCPND	功能位	描述	复位值
INT_ADC	[31]	0=中断未请求;1=中断已请求	0
INT_RTC	[30]	0=中断未请求;1=中断已请求	0
INT_SPI1	[29]	0=中断未请求;1=中断已请求	0
INT_UART0	[28]	0=中断未请求;1=中断已请求	0
INT_IIC	[27]	0=中断未请求;1=中断已请求	0
INT_USBH	[26]	0=中断未请求;1=中断已请求	0
INT_USBD	[25]	0=中断未请求;1=中断已请求	0
INT_NFCON	[24]	0=中断未请求;1=中断已请求	0
INT_UART1	[23]	0=中断未请求;1=中断已请求	0
INT_SPI0	[22]	0=中断未请求;1=中断已请求	0
INT_SDI	[21]	0=中断未请求;1=中断已请求	0
INT_DMA3	[20]	0=中断未请求;1=中断已请求	0

中断寄存器

INT_DMA2	[19]	0=中断未请求;1=中断已请求	0
INT_DMA1	[18]	0=中断未请求;1=中断已请求	0
INT_DMA0	[17]	0=中断未请求;1=中断已请求	0
INT_LCD	[16]	0=中断未请求;1=中断已请求	0
INT_UART2	[15]	0=中断未请求;1=中断已请求	0
INT_TIMER4	[14]	0=中断未请求;1=中断已请求	0
INT_TIMER3	[13]	0=中断未请求;1=中断已请求	0
INT_TIMER2	[12]	0=中断未请求;1=中断已请求	0
INT_TIMER1	[11]	0=中断未请求;1=中断已请求	0
INT_TIMER0	[10]	0=中断未请求;1=中断已请求	0
INT_WDT_AC97	[9]	0=中断未请求;1=中断已请求	0
INT_TICK	[8]	0=中断未请求;1=中断已请求	0
nBATT_FLT	[7]	0=中断未请求;1=中断已请求	0
INT_CAM	[6]	0=中断未请求;1=中断已请求	0
EINT8_23	[5]	0=中断未请求;1=中断已请求	0
EINT4_7	[4]	0=中断未请求;1=中断已请求	0
EINT3	[3]	0=中断未请求;1=中断已请求	0
EINT2	[2]	0=中断未请求;1=中断已请求	0
EINT1	[1]	0=中断未请求;1=中断已请求	0
EINT0	[0]	0=中断未请求;1=中断已请求	0

INTMASK中断掩码寄存器

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
INTMASK	0x4A000008	读写	决定哪个中断源被屏蔽。 0=允许中断;1=屏蔽中断	0xFFFFFFFF

INTMASK	功能位	描述	复位值
INT_ADC	[31]	0=允许中断;1=屏蔽中断	1
INT_RTC	[30]	0=允许中断;1=屏蔽中断	1
INT_SPI1	[29]	0=允许中断;1=屏蔽中断	1
INT_UART0	[28]	0=允许中断;1=屏蔽中断	1
INT_IIC	[27]	0=允许中断;1=屏蔽中断	1
INT_USBH	[26]	0=允许中断;1=屏蔽中断	1
INT_USBD	[25]	0=允许中断;1=屏蔽中断	1
INT_NFCON	[24]	0=允许中断;1=屏蔽中断	1
INT_UART1	[23]	0=允许中断;1=屏蔽中断	1
INT_SPI0	[22]	0=允许中断;1=屏蔽中断	1
INT_SDI	[21]	0=允许中断;1=屏蔽中断	1
INT_DMA3	[20]	0=允许中断;1=屏蔽中断	1
INT_DMA2	[19]	0=允许中断;1=屏蔽中断	1
INT_DMA1	[18]	0=允许中断;1=屏蔽中断	1
INT_DMA0	[17]	0=允许中断;1=屏蔽中断	1
INT_LCD	[16]	0=允许中断;1=屏蔽中断	1
INT_UART2	[15]	0=允许中断;1=屏蔽中断	1

INT_TIMER4	[14]	0=允许中断;1=屏蔽中断	1
INT_TIMER3	[13]	0=允许中断;1=屏蔽中断	1
INT_TIMER2	[12]	0=允许中断;1=屏蔽中断	1
INT_TIMER1	[11]	0=允许中断;1=屏蔽中断	1
INT_TIMER0	[10]	0=允许中断;1=屏蔽中断	1
INT_WDT_AC97	[9]	0=允许中断;1=屏蔽中断	1
INT_TICK	[8]	0=允许中断;1=屏蔽中断	1
nBATT_FLT	[7]	0=允许中断;1=屏蔽中断	1
INT_CAM	[6]	0=允许中断;1=屏蔽中断	1
EINT8_23	[5]	0=允许中断;1=屏蔽中断	1
EINT4_7	[4]	0=允许中断;1=屏蔽中断	1
EINT3	[3]	0=允许中断;1=屏蔽中断	1
EINT2	[2]	0=允许中断;1=屏蔽中断	1
EINT1	[1]	0=允许中断;1=屏蔽中断	1
EINT0	[0]	0=允许中断;1=屏蔽中断	1

INTPND中断未决寄存器

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哪个中断源被服务。

寄存器	地址	读/写	描述	复位值
INTPND	0x4A000010	读写	指出中断请求的状态 0=中断未请求;1=中断已请求	0x00000000

INTPND	功能位	描述	复位值
INT_ADC	[31]	0=中断未请求;1=中断已请求	0
INT_RTC	[30]	0=中断未请求;1=中断已请求	0
INT_SPI1	[29]	0=中断未请求;1=中断已请求	0
INT_UART0	[28]	0=中断未请求;1=中断已请求	0
INT_IIC	[27]	0=中断未请求;1=中断已请求	0
INT_USBH	[26]	0=中断未请求;1=中断已请求	0
INT_USBD	[25]	0=中断未请求;1=中断已请求	0
INT_NFCON	[24]	0=中断未请求;1=中断已请求	0
INT_UART1	[23]	0=中断未请求;1=中断已请求	0
INT_SPI0	[22]	0=中断未请求;1=中断已请求	0
INT_SDI	[21]	0=中断未请求;1=中断已请求	0
INT_DMA3	[20]	0=中断未请求;1=中断已请求	0
INT_DMA2	[19]	0=中断未请求;1=中断已请求	0
INT_DMA1	[18]	0=中断未请求;1=中断已请求	0
INT_DMA0	[17]	0=中断未请求;1=中断已请求	0
INT_LCD	[16]	0=中断未请求;1=中断已请求	0
INT_UART2	[15]	0=中断未请求;1=中断已请求	0

INT_TIMER4	[14]	0=中断未请求;1=中断已请求	0
INT_TIMER3	[13]	0=中断未请求;1=中断已请求	0
INT_TIMER2	[12]	0=中断未请求;1=中断已请求	0
INT_TIMER1	[11]	0=中断未请求;1=中断已请求	0
INT_TIMER0	[10]	0=中断未请求;1=中断已请求	0
INT_WDT_AC97	[9]	0=中断未请求;1=中断已请求	0
INT_TICK	[8]	0=中断未请求;1=中断已请求	0
nBATT_FLT	[7]	0=中断未请求;1=中断已请求	0
INT_CAM	[6]	0=中断未请求;1=中断已请求	0
EINT8_23	[5]	0=中断未请求;1=中断已请求	0
EINT4_7	[4]	0=中断未请求;1=中断已请求	0
EINT3	[3]	0=中断未请求;1=中断已请求	0
EINT2	[2]	0=中断未请求;1=中断已请求	0
EINT1	[1]	0=中断未请求;1=中断已请求	0
EINT0	[0]	0=中断未请求;1=中断已请求	0

INTMOD中断模式寄存器

- ✓ 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。
- ✓ **注意仅有一个中断源能够在FIQ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	读写	中断模式寄存器 0 = IRQ 模式 1 = FIR 模式	0x00000000

INTMOD	功能位	描述	复位值
INT_ADC	[31]	0 =IRQ; 1=FIQ	0
INT_RTC	[30]	0 =IRQ; 1=FIQ	0
INT_SPI1	[29]	0 =IRQ; 1=FIQ	0
INT_UART0	[28]	0 =IRQ; 1=FIQ	0
INT_IIC	[27]	0 =IRQ; 1=FIQ	0
INT_USBH	[26]	0 =IRQ; 1=FIQ	0
INT_USBD	[25]	0 =IRQ; 1=FIQ	0
INT_NFCON	[24]	0 =IRQ; 1=FIQ	0
INT_UART1	[23]	0 =IRQ; 1=FIQ	0
INT_SPI0	[22]	0 =IRQ; 1=FIQ	0
INT_SDI	[21]	0 =IRQ; 1=FIQ	0
INT_DMA3	[20]	0 =IRQ; 1=FIQ	0
INT_DMA2	[19]	0 =IRQ; 1=FIQ	0
INT_DMA1	[18]	0 =IRQ; 1=FIQ	0
INT_DMA0	[17]	0 =IRQ; 1=FIQ	0
INT_LCD	[16]	0 =IRQ; 1=FIQ	0
INT_UART2	[15]	0 =IRQ; 1=FIQ	0

INT_TIMER4	[14]	0 =IRQ; 1=FIQ	0
INT_TIMER3	[13]	0 =IRQ; 1=FIQ	0
INT_TIMER2	[12]	0 =IRQ; 1=FIQ	0
INT_TIMER1	[11]	0 =IRQ; 1=FIQ	0
INT_TIMER0	[10]	0 =IRQ; 1=FIQ	0
INT_WDT_AC97	[9]	0 =IRQ; 1=FIQ	0
INT_TICK	[8]	0 =IRQ; 1=FIQ	0
nBATT_FLT	[7]	0 =IRQ; 1=FIQ	0
INT_CAM	[6]	0 =IRQ; 1=FIQ	0
EINT8_23	[5]	0 =IRQ; 1=FIQ	0
EINT4_7	[4]	0 =IRQ; 1=FIQ	0
EINT3	[3]	0 =IRQ; 1=FIQ	0
EINT2	[2]	0 =IRQ; 1=FIQ	0
EINT1	[1]	0 =IRQ; 1=FIQ	0
EINT0	[0]	0 =IRQ; 1=FIQ	0

中断寄存器

PRIORITY中断优先级寄存器

寄存器	地址	读/写	描述	复位值
PRIORITY	0x4A00000C	读写	中断优先级寄存器	0x0000007F

PRIORITY	功能位	描述	复位值
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

中断寄存器

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
		0 = 优先级不翻转 1 = 优先级翻转使能	

当ARB_MODEx 为0时，可以通过软件设置对应的ARB_SELx改变优先级顺序，在运行过程中顺序是固定不变的。若ARB_MODEx 为1，REQ 0和5优先级固定不变，REQ 1～4，若其中一个被响应，则其优先级降为REQ 1～4中最低。

中断寄存器

INTOFFSET中断偏移寄存器

- ✓ 中断偏移寄存器中的值显示了哪个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	只读	指出 IRQ 中断请求源	0x00000000

中断寄存器

INTOFFSET	偏移量 INTOFFSET	偏移量	INTOFFSET
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

中断寄存器

SUBSRCPND子中断源未决寄存器

子中断源未决寄存器与中断源未决寄存器(SRCPND)功能一样，它用14个位代表14个子中断的请求状态。

寄存器	地址	读/写	描述	复位值
SUBSRCPND	0x4A000018	读写	指出子中断请求的状态 0=中断未请求;1=中断已请求	0x00000000

中断寄存器

SUBSRCPND	功能位	描述	复位值
INT_AC97	[14]	0=中断未请求;1=中断已请求	0
INT_WDT	[13]	0=中断未请求;1=中断已请求	0
INT_CAM_P	[12]	0=中断未请求;1=中断已请求	0
INT_CAM_C	[11]	0=中断未请求;1=中断已请求	0
INT_ADC_S	[10]	0=中断未请求;1=中断已请求	0
INT_TC	[9]	0=中断未请求;1=中断已请求	0
INT_ERR2	[8]	0=中断未请求;1=中断已请求	0
INT_TXD2	[7]	0=中断未请求;1=中断已请求	0
INT_RXD2	[6]	0=中断未请求;1=中断已请求	0
INT_ERR1	[5]	0=中断未请求;1=中断已请求	0
INT_TXD1	[4]	0=中断未请求;1=中断已请求	0
INT_RXD1	[3]	0=中断未请求;1=中断已请求	0
INT_ERR0	[2]	0=中断未请求;1=中断已请求	0
INT_TXD0	[1]	0=中断未请求;1=中断已请求	0
INT_RXD0	[0]	0=中断未请求;1=中断已请求	0

中断寄存器

SUBINTMSK子中断掩码寄存器

子中断掩码寄存器与中断掩码寄存器(INTPND)功能一样，它用14个位代表14个子中断的屏蔽状态。

寄存器	地址	读/写	描述	复位值
SUBINTMSK	0x4A00001C	读写	决定哪个中断源被屏蔽。 0=允许中断；1=屏蔽中断	0xFFFF

中断寄存器

SUBINTMSK	功能位	描述	复位值
INT_AC97	[14]	0=允许中断;1=屏蔽中断	1
INT_WDT	[13]	0=允许中断;1=屏蔽中断	1
INT_CAM_P	[12]	0=允许中断;1=屏蔽中断	1
INT_CAM_C	[11]	0=允许中断;1=屏蔽中断	1
INT_ADC_S	[10]	0=允许中断;1=屏蔽中断	1
INT_TC	[9]	0=允许中断;1=屏蔽中断	1
INT_ERR2	[8]	0=允许中断;1=屏蔽中断	1
INT_TXD2	[7]	0=允许中断;1=屏蔽中断	1
INT_RXD2	[6]	0=允许中断;1=屏蔽中断	1
INT_ERR1	[5]	0=允许中断;1=屏蔽中断	1
INT_TXD1	[4]	0=允许中断;1=屏蔽中断	1
INT_RXD1	[3]	0=允许中断;1=屏蔽中断	1
INT_ERR0	[2]	0=允许中断;1=屏蔽中断	1
INT_TXD0	[1]	0=允许中断;1=屏蔽中断	1
INT_RXD0	[0]	0=允许中断;1=屏蔽中断	1

中断寄存器

EINTPEND外部中断未决寄存器

外部中断未决寄存器与中断源未决寄存器(SRCPND)功能一样，它用20个位代表20个外部二级中断的请求状态。

寄存器	地址	读/写	描述	复位值
EINTPEND	0x560000A8	读写	指出外部中断请求的状态 0=中断未请求;1=中断已请求	0x00000000

中断寄存器

EINTPEND	功能位	描述	复位值
EINT23	[23]	0=中断未请求;1=中断已请求	0
EINT22	[22]	0=中断未请求;1=中断已请求	0
EINT21	[21]	0=中断未请求;1=中断已请求	0
EINT20	[20]	0=中断未请求;1=中断已请求	0
EINT19	[19]	0=中断未请求;1=中断已请求	0
EINT18	[18]	0=中断未请求;1=中断已请求	0
EINT17	[17]	0=中断未请求;1=中断已请求	0
EINT16	[16]	0=中断未请求;1=中断已请求	0
EINT15	[15]	0=中断未请求;1=中断已请求	0
EINT14	[14]	0=中断未请求;1=中断已请求	0
EINT13	[13]	0=中断未请求;1=中断已请求	0
EINT12	[12]	0=中断未请求;1=中断已请求	0
EINT11	[11]	0=中断未请求;1=中断已请求	0
EINT10	[10]	0=中断未请求;1=中断已请求	0
EINT9	[9]	0=中断未请求;1=中断已请求	0
EINT8	[8]	0=中断未请求;1=中断已请求	0
EINT7	[7]	0=中断未请求;1=中断已请求	0
EINT6	[6]	0=中断未请求;1=中断已请求	0
EINT5	[5]	0=中断未请求;1=中断已请求	0
EINT4	[4]	0=中断未请求;1=中断已请求	0
保留	[3:0]	保留	0

中断寄存器

EINTMASK外部中断掩码寄存器

外部中断掩码寄存器与中断掩码寄存器(INTPND)功能一样，它用 20 个位代表 20 个外部二级中断的屏蔽状态。

寄存器	地址	读/写	描述	复位值
EINTMASK	0x560000A4	读写	决定哪个外部中断源被屏蔽。 0=允许中断；1=屏蔽中断	0x000FFFFF

EINTMASK	功能位	描述	复位值
EINT23	[23]	0=允许中断；1=屏蔽中断	1
EINT22	[22]	0=允许中断；1=屏蔽中断	1
EINT21	[21]	0=允许中断；1=屏蔽中断	1
EINT20	[20]	0=允许中断；1=屏蔽中断	1
EINT19	[19]	0=允许中断；1=屏蔽中断	1
EINT18	[18]	0=允许中断；1=屏蔽中断	1
EINT17	[17]	0=允许中断；1=屏蔽中断	1
EINT16	[16]	0=允许中断；1=屏蔽中断	1

中断寄存器

EINTMASK	功能位	描述	复位值
EINT23	[23]	0=允许中断;1=屏蔽中断	1
EINT22	[22]	0=允许中断;1=屏蔽中断	1
EINT21	[21]	0=允许中断;1=屏蔽中断	1
EINT20	[20]	0=允许中断;1=屏蔽中断	1
EINT19	[19]	0=允许中断;1=屏蔽中断	1
EINT18	[18]	0=允许中断;1=屏蔽中断	1
EINT17	[17]	0=允许中断;1=屏蔽中断	1
EINT16	[16]	0=允许中断;1=屏蔽中断	1
EINT15	[15]	0=允许中断;1=屏蔽中断	1
EINT14	[14]	0=允许中断;1=屏蔽中断	1
EINT13	[13]	0=允许中断;1=屏蔽中断	1
EINT12	[12]	0=允许中断;1=屏蔽中断	1
EINT11	[11]	0=允许中断;1=屏蔽中断	1
EINT10	[10]	0=允许中断;1=屏蔽中断	1
EINT9	[9]	0=允许中断;1=屏蔽中断	1
EINT8	[8]	0=允许中断;1=屏蔽中断	1
EINT7	[7]	0=允许中断;1=屏蔽中断	1
EINT6	[6]	0=允许中断;1=屏蔽中断	1
EINT5	[5]	0=允许中断;1=屏蔽中断	1
EINT4	[4]	0=允许中断;1=屏蔽中断	1
保留	[3:0]	保留	1111

中断寄存器

EXTINTn外部中断控制寄存器

- ✓ EXTINT0- EXTINT3用于设置外部中断的中断触发方式，并且可以设置EINT8-EINT23这16个中断的滤波功能。
- ✓ 边沿触发：即在引脚的电平由低到高变化或着由高到低变化时触发中断。
- ✓ 电平触发：即在引脚的电平由高到低或由低到高进行跳变，并保持一定时间后触发。

寄存器	地址	读/写	描述	复位值
EXTINT0	0x56000088	读写	外部中断控制寄存器 0，用于设定外部中断的触发方式	0x000000
EXTINT1	0x5600008C	读写	外部中断控制寄存器 1，用于设定外部中断的触发方式	0x000000
EXTINT2	0x56000090	读写	外部中断控制寄存器 2，用于设定外部中断的触发方式	0x000000

中断寄存器

EXTINT0	功能位	描述	复位值
EINT7	[30:28]	设置EINT7的中断触发方式 000=低电平触发 001=高电平触发 01x=下降沿触发 10x=上升沿触发 11x=上升沿和下降沿都触发	000
EINT6	[26:24]	设置EINT6的中断触发方式 000=低电平触发 001=高电平触发 01x=下降沿触发 10x=上升沿触发 11x=上升沿和下降沿都触发	000
EINT5	[22:20]	设置EINT5的中断触发方式 000=低电平触发 001=高电平触发 01x=下降沿触发 10x=上升沿触发 11x=上升沿和下降沿都触发	000
EINT4	[18:16]	设置EINT4的中断触发方式 000=低电平触发 001=高电平触发 01x=下降沿触发 10x=上升沿触发 11x=上升沿和下降沿都触发	000
EINT3	[14:12]	设置EINT3的中断触发方式 000=低电平触发 001=高电平触发 01x=下降沿触发 10x=上升沿触发 11x=上升沿和下降沿都触发	000
EINT2	[10:8]	设置EINT2的中断触发方式 000=低电平触发 001=高电平触发 01x=下降沿触发 10x=上升沿触发 11x=上升沿和下降沿都触发	000

中断寄存器

EINT1	[6:4]	设置EINT1的中断触发方式 000=低电平触发 001=高电平触发 01x=下降沿触发 10x=上升沿触发 11x=上升沿和下降沿都触发	000
EINT0	[2:0]	设置EINT0的中断触发方式 000=低电平触发 001=高电平触发 01x=下降沿触发 10x=上升沿触发 11x=上升沿和下降沿都触发	000

中断寄存器

EXTINT1	功能位	描述	复位值
FLTEN15	[31]	EINT15是否允许滤波 0-禁止滤波　1-开启滤波	0
EINT15	[30:28]	设置EINT15断触发方式 000=低电平触发 001=高电平触发 01x=下降沿触发 10x=上升沿触发 11x=上升沿和下降沿都触发	000
FLTEN14	[27]	EINT14是否允许滤波 0-禁止滤波　1-开启滤波	0
EINT14	[26:24]	设置EINT14断触发方式 000=低电平触发 001=高电平触发 01x=下降沿触发 10x=上升沿触发 11x=上升沿和下降沿都触发	000
FLTEN13	[23]	EINT13是否允许滤波 0-禁止滤波　1-开启滤波	0
EINT13	[22:20]	设置EINT13断触发方式 000=低电平触发 001=高电平触发 01x=下降沿触发 10x=上升沿触发 11x=上升沿和下降沿都触发	000
FLTEN12	[19]	EINT12是否允许滤波 0-禁止滤波　1-开启滤波	0
EINT12	[18:16]	设置EINT12断触发方式 000=低电平触发 001=高电平触发 01x=下降沿触发 10x=上升沿触发 11x=上升沿和下降沿都触发	000

中断寄存器

FLTEN11	[15]	EINT11是否允许滤波 0-禁止滤波　1-开启滤波	0
EINT11	[14:12]	设置EINT11断触发方式 000=低电平触发 001=高电平触发 01x=下降沿触发 10x=上升沿触发 11x=上升沿和下降沿都触发	000
FLTEN10	[11]	EINT10是否允许滤波 0-禁止滤波　1-开启滤波	0
EINT10	[10:8]	设置EINT10断触发方式 000=低电平触发 001=高电平触发 01x=下降沿触发 10x=上升沿触发 11x=上升沿和下降沿都触发	000
FLTEN9	[7]	EINT9是否允许滤波 0-禁止滤波　1-开启滤波	0
EINT9	[6:4]	设置EINT9中断触发方式 000=低电平触发 001=高电平触发 01x=下降沿触发 10x=上升沿触发 11x=上升沿和下降沿都触发	000
FLTEN8	[3]	EINT8是否允许滤波 0-禁止滤波　1-开启滤波	0
EINT8	[2:0]	设置EINT8中断触发方式 000=低电平触发 001=高电平触发 01x=下降沿触发 10x=上升沿触发 11x=上升沿和下降沿都触发	000

中断寄存器

EXTINT2	功能位	描述	复位值
FLTEN23	[31]	EINT23是否允许滤波 0-禁止滤波　1-开启滤波	0
EINT23	[30:28]	设置EINT23断触发方式 000=低电平触发 001=高电平触发 01x=下降沿触发 10x=上升沿触发 11x=上升沿和下降沿都触发	000
FLTEN22	[27]	EINT22是否允许滤波 0-禁止滤波　1-开启滤波	0
EINT22	[26:24]	设置EINT22断触发方式 000=低电平触发 001=高电平触发 01x=下降沿触发 10x=上升沿触发 11x=上升沿和下降沿都触发	000
FLTEN21	[23]	EINT21是否允许滤波 0-禁止滤波　1-开启滤波	0
EINT21	[22:20]	设置EINT21断触发方式 000=低电平触发 001=高电平触发 01x=下降沿触发 10x=上升沿触发 11x=上升沿和下降沿都触发	000
FLTEN20	[19]	EINT20是否允许滤波 0-禁止滤波　1-开启滤波	0
EINT20	[18:16]	设置EINT20断触发方式 000=低电平触发 001=高电平触发 01x=下降沿触发 10x=上升沿触发 11x=上升沿和下降沿都触发	000

中断寄存器

FLTEN19	[15]	EINT19是否允许滤波 0-禁止滤波　1-开启滤波	0
EINT19	[14:12]	设置EINT19断触发方式 000=低电平触发 001=高电平触发 01x=下降沿触发 10x=上升沿触发 11x=上升沿和下降沿都触发	000
FLTEN18	[11]	EINT18是否允许滤波 0-禁止滤波　1-开启滤波	0
EINT18	[10:8]	设置EINT18断触发方式 000=低电平触发 001=高电平触发 01x=下降沿触发 10x=上升沿触发 11x=上升沿和下降沿都触发	000
FLTEN17	[7]	EINT17是否允许滤波 0-禁止滤波　1-开启滤波	0
EINT17	[6:4]	设置EINT17中断触发方式 000=低电平触发 001=高电平触发 01x=下降沿触发 10x=上升沿触发 11x=上升沿和下降沿都触发	000
FLTEN16	[3]	EINT16是否允许滤波 0-禁止滤波　1-开启滤波	0
EINT16	[2:0]	设置EINT16中断触发方式 000=低电平触发 001=高电平触发 01x=下降沿触发 10x=上升沿触发 11x=上升沿和下降沿都触发	000

中断寄存器

EINTFLTn外部中断滤波寄存器

EINTFLT2	功能位	描述	复位值
FLTCLK19	[31]	EINT19的滤波时钟类型(通过OM进行配置) 0=PCLK 1=EXTCLK/OSC_CLK	
EINTFLT19	[30:24]	EINT19的滤波宽度	
FLTCLK18	[23]	EINT18的滤波时钟类型(通过OM进行配置) 0=PCLK 1=EXTCLK/OSC_CLK	
EINTFLT18	[22:16]	EINT18的滤波宽度	
FLTCLK17	[15]	EINT17的滤波时钟类型(通过OM进行配置) 0=PCLK 1=EXTCLK/OSC_CLK	
EINTFLT17	[14:8]	EINT17的滤波宽度	
FLTCLK16	[7]	EINT16的滤波时钟类型(通过OM进行配置) 0=PCLK 1=EXTCLK/OSC_CLK	
EINTFLT16	[6:0]	EINT16的滤波宽度	

中断寄存器

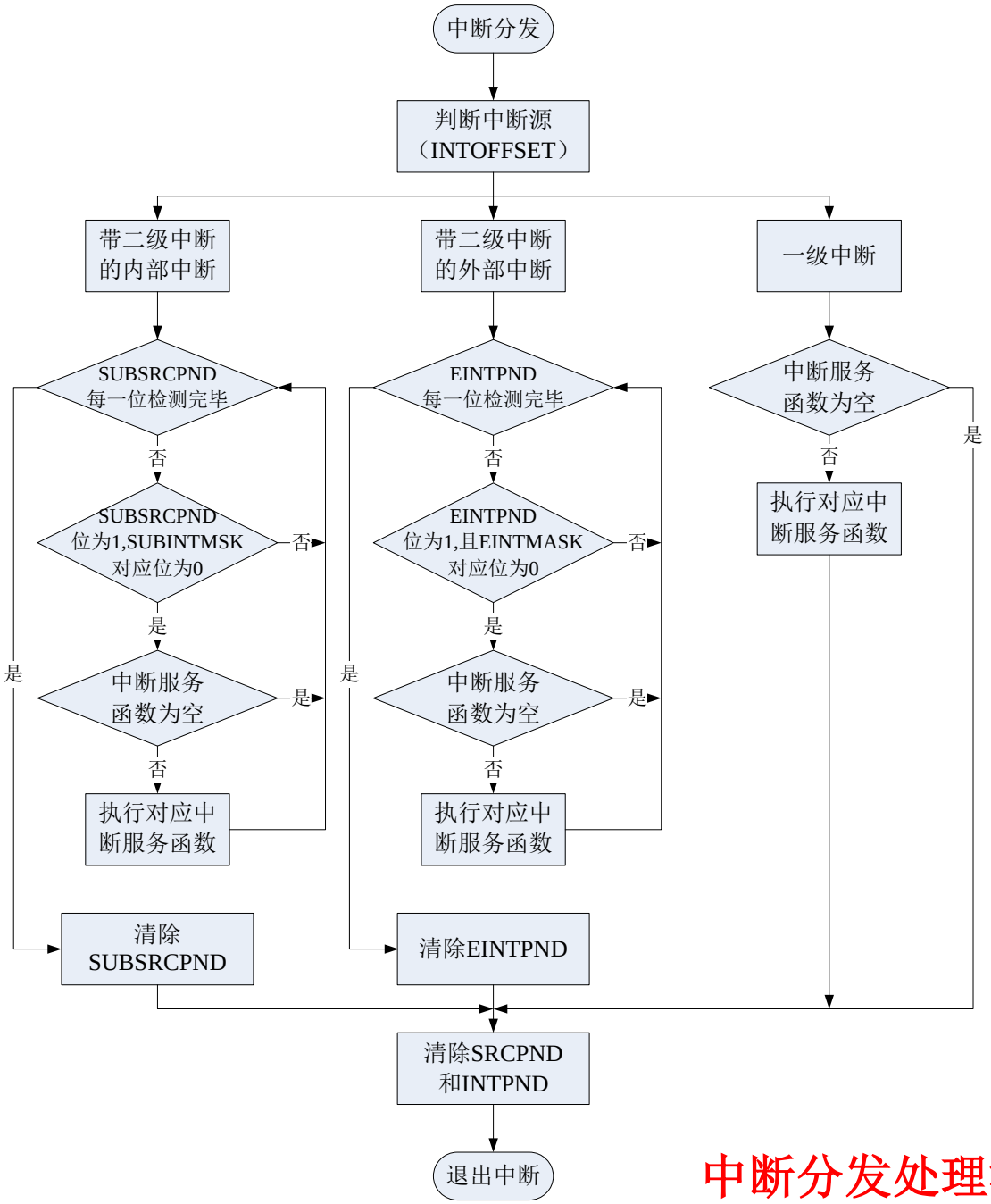
EINTFLT3	功能位	描述	复位值
FLTCLK23	[31]	EINT23的滤波时钟类型(通过OM进行配置) 0=PCLK 1=EXTCLK/OSC_CLK	
EINTFLT23	[30:24]	EINT23的滤波宽度	
FLTCLK22	[23]	EINT22的滤波时钟类型(通过OM进行配置) 0=PCLK 1=EXTCLK/OSC_CLK	
EINTFLT22	[22:16]	EINT22的滤波宽度	
FLTCLK21	[15]	EINT21的滤波时钟类型(通过OM进行配置) 0=PCLK 1=EXTCLK/OSC_CLK	
EINTFLT21	[14:8]	EINT21的滤波宽度	
FLTCLK20	[7]	EINT20的滤波时钟类型(通过OM进行配置) 0=PCLK 1=EXTCLK/OSC_CLK	
EINTFLT20	[6:0]	EINT20的滤波宽度	

注：

- I/O 口的电平变化时(例如按键按下和放开)，并不是垂直变化，都会有一些纹波抖动，一般情况下要把这些纹波滤除，以避免误操作。

中断分发处理

- 应用程序运行过程中，产生中断事件，硬件强制将PC指针指向0x00000018（IRQ）或0x0000001c（FIQ）。
- S3C2440有59种中断源，在同一时刻，最多只允许一种中断源为FIQ模式，其它可以全部为IRQ模式。
- 在IRQ模式的中断源发生中断时，程序必须识别出是哪一种中断源，根据不同中断源调用对应中断服务函数。
- 中断源的识别过程，也就是中断分发处理。


中断分发
判断中断源
（INTOFFSET）
带二级中断
的内部中断
带二级中断
的外部中断
一级中断
SUBSRCPND
每一位检测完毕
EINTPND
每一位检测完毕
中断服务
函数为空
否
是
SUBSRCPND
位为1,SUBINTMSK
对应位为0
EINTPND
位为1,且EINTMASK
对应位为0
执行对应中
断服务函数
清除
SUBSRCPND
清除EINTPND
清除SRCPND
和INTPND
退出中断

中断分发处理流程图

中断分发处理

分发处理流程：

① 进入中断分发处理后，根据INTOFFSET快速识别出中断编号（范围为0~31，表示SRCPND和INTPND的哪一位被置1）。

② 对中断编号进行分类，可分为一级中断、带二级中断的外部中断、带二级中断的内部中断。

③ 一级中断直接处理。

④ 带二级中断的外部中断和带二级中断的内部中断，处理过程一致，均要遍历二级中断的中断未决寄存器，只有在二级中断未决寄存器中的位为1（中断产生）且二级中断掩码寄存器中对应位的值为0（中断使能），才进行处理。遍历结束后，对二级中断未决寄存器所有位进行清零。

⑤ 清除SRCPND和INTPND，注意清除顺序，要先对SRCPND清零,再对INTPND清零。

外部中断编程

- 例子：结合如下电路分析中断响应过程，掌握中断函数编写。其中，左边是LED接口电路，右边是按键接口电路。要求根据按键来控制LED灯的亮灭。

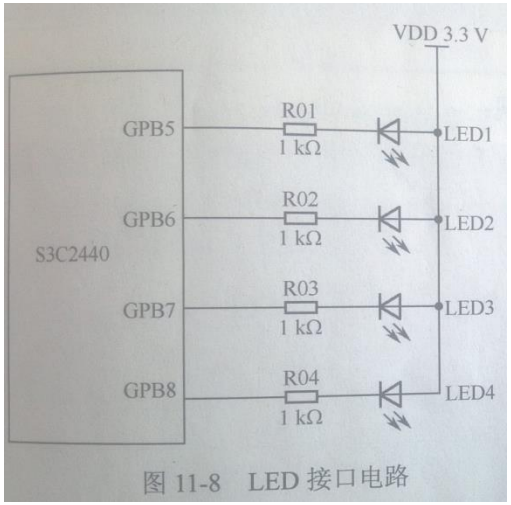


图 11-8　LED 接口电路

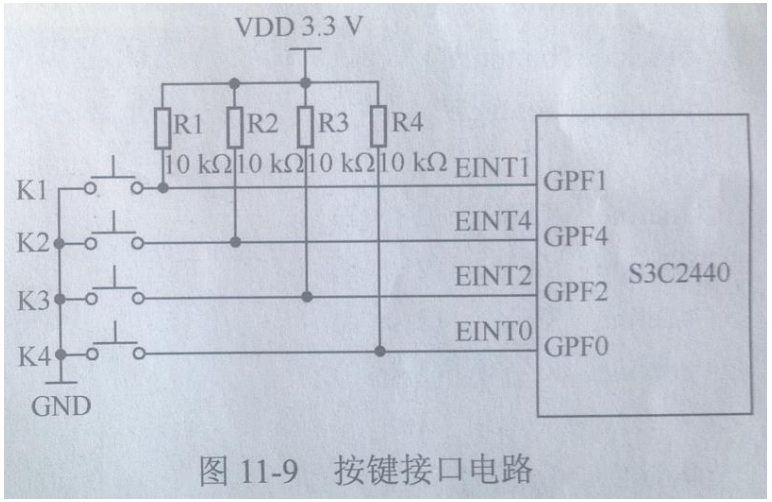


图 11-9　按键接口电路

外部中断编程

- 问题：
  - ① 如何对外部中断进行初始化？
  - ② CPU如何检测哪个中断发生了？
  - ③ 如何安装中断处理函数？
  - ④ 中断执行流程如何？ARM处理器流水结构对中断返回地址的计算有着什么影响？

  ……

谢谢！

Q & A