## S5PV210中断处理

单核CPU实际无法并行,通过中断机制，可以实现假并行,宏观上的并行，微观上实际还是串行。SOC对中断的实现机制是异常向量表，异常向量表是CPU中某些特定地址的特定定义。

在CPU设计时，就事先定义了CPU中的一些特定地址作为异常的入口地址，在发生异常时，CPU会以硬件实现的方式自动跳转到事先设定的地址去执行指令。

以上是CPU硬件设计时对异常向量表的支持，接下来就需要软件支持了，硬件已经决定了发生什么异常CPU会自动跳转PC到哪个地址去执行，软件需要做的是把处理这个异常的代码的首地址填入这个异常向量地址。

异常向量表中各个向量的相对位置是固定的，但是它们的起始地址是不固定的，各种SOC可以不一样，而且复杂ARM中还可以让用户来软件设置这个异常向量表的基地址。

异常和中断的区别和联系，

发生复位，软中断，中断，快速中断，取指令异常，数据异常等，我们都统一叫异常，所以中断其实是异常的一种，

异常的定义就是突发事件，打断了CPU的正常常规业务，CPU不得不跳转到异常向量表中去执行异常处理程序。

## 中断处理过程

当外设异常发生时触发一个中断，同时会设置相应的中断控制寄存器.中断控制寄存器会通知CPU有中断发生.CPU收到中断请求后就中断当前正在执行的程序.跳转到一个固定的地址处理这个异常,最后再返回继续执行被中断的程序。此方式实现比较复杂，但效率很高，是比较常用的方法。

不同CPU中断处理的过程一般都是类似的，大致有如下几个步骤。

(1)中断控制器负责收集各类外设发出的中断信号，然后通知CPU。

(2) CPU收到通知后保存正在执行程序的状态(即保存各寄存器等)，调用中断服务程序(Interrupt Service Routine ,ISR )来处理中断请求。

(3)在ISR中通过读取中断控制器、外设的相关寄存器来识别这是哪个中断,并进行相应的处理。

(4)通过读/写中断控制寄存器和外设相关的寄存器来清除中断。(5)最后恢复被中断程序的运行环境(即上面保存的寄存器等)。

对于S5PV210来说，中断处理过程也是一样的，也是上述这些步骤,但在实现上:与以前的S3C2440等老架构相比，有了很大的不同。在S3C2440上，上电执行的第1个程序通常写在start.S里,这个文件的开头必定是关于中断向量相关的语句,具体内容如下:

. globl\_ start

\_start: b reset

ldr pc, \_undefine\_ interrupt

Idr pc, \_software\_interrupt

ldr pc, \_prefetc\_abort

ldr pc, \_data\_abort

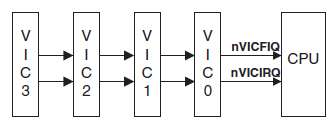
ldr pc, \_not\_used

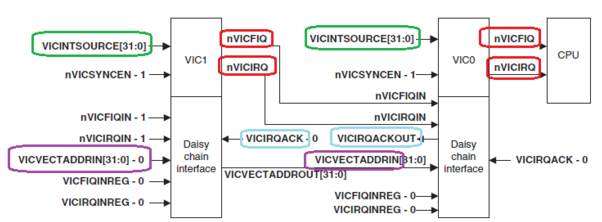
ldr pc, \_irq

ldr pc. \_fiq

这就说明了以前中断向量的入口地址是固定的，也就是通常说的0x0000\_0000或0xFFFF\_0000,S5PV210中断向量存放在从0xD003\_7400开始的128字节空间里，实际上当中断发生时CPU的PC寄存器还是先跳到从0地址开始的中断向量表中找到对应的中断，执行中断服务程序，比如跳到0x18地址处理IRQ中断

S5pv210的中断控制器由4个向量中断控制器（VIC）APM PrimeCell PL192和4个安全中断控制器（TZIC）共同组成，4个ARM PrimeCell PL192矢量中断控制器级连(daisy-chain)，每个中断寄存器都是32位，每位对应一个中断源，它们的关系是并列的，得4个寄存器一个一个去找是哪一个中断，进而确定中断编号，所以最多支持128个。S5PV210使用了其中的93个。所谓“矢量”是指当中断发生时，软件可以直接从VIC得到提前设好的中断服务程序ISR（Interrupt Service Routine）对于S5PV210这样的单核CPU来说，PL192是一种比较高效的VIC。硬件上的连接图大体如下所示：





最终连接到ARM CPU上的输入信号只有两个，IRQ和FIQ

## 中断控制寄存器介绍

以VIC0为例：

1）VIC0IRQSTATUS VIC0FIQSTATUS和VIC0RAWINTR寄存器

这是32位的寄存器。其中每一位都代表一个中断源的状态，通过寄存器的某一位可以知道相应中断是否被屏蔽，初始状态都是被屏蔽掉的，当有中断触发后，硬件上会把相应的位置1。

VICOFIQSTATUS寄存器代表快速中断源的状态，用法与VICOIRQSTATUS类似。

2) VIC0INTSELECT、VIC0INTENABLE和VICOINTENCLEAR寄存器

VICOINTSELECT寄存器中的每一位用于设置相应中断源是IRQ中断还是FIQ中断。默认是IRQ中断，如果某位被置1即为FIQ中断。

VICOINTENABLE寄存器用于开启中断，相应的位被写人1即开启，如果读此寄存器，相应的位为0表示中断禁止,反之为开启; VIC0INTENCLEAR用于清除VICOINTENABLE寄存器中相应的位,当相应的位被写入1即把VIC0INTENABLE寄存器中开启的相应中断禁止掉。需要特别注意的是,改变这两个寄存器中相应位的状态只有写人1才有效,写人0无效。

3) VICOSOFTINT和VICOSOFTINTCLEAR寄存器

VICOSOFTINT用于开启软件中断。当相应的位被写人1即开启.如果读此寄存器，相应位为0时表示对应的中断禁止反之为开启; VICOSOFTINTCLEAR寄存器用于禁止被VICOSOFTINT开启的中断，禁止的方法是往相应位写入1。需要注意的是，此两种寄存器都是写入1才有效,写人0无效。

4) VICOPROTECTION寄存器

VICOPROTECTION用于控制寄存器的访问权限。当bin[O]为1时开启保护模式，此模式下只有系统工作在特权模式下才可访问中断控制器寄存器；如果为0在用户模式和特权模式下都可访问。

5) VICOADDRESS 和VICOVECTADDR[0:31]寄存器

当中断发生时，CPU需跳到中断服务程序处执行(ISR)，对应中断服务程序的入口就由VIC0ADDRESS寄存器上报给CPU,所以当中断处理完毕，此寄存器必须清零。往寄存器中写入任何数据都会清除当前的中断.读寄存器可以知道ISR的入口地址。

VICOVECTADDRO~VICOVECTADDR31寄存器保存中断服务程序的人口地址。由于CPU每次只能有一个中断在执行。所以这32个中断根据优先级先后.每次只有一个服务程序的入口地址被自动送给VICOADDRESS寄存器.再由VIOADDRESS将中断服务程序的入口地址上报给CPU。

6) VICOSWPRIORITYMASK寄存器

VICOSWPRIORITYMASK寄存器用于启用或禁止16个中断优先级，默认是没有被屏蔽的，当写入0时即启用，写入1时即禁止，寄存器的前十六位对应十六个中断优先级。

7) VIC0VECTPRIORITY[0:31]寄存器

该寄存器为每一个中断设置优先级，VIC0VECTPRIORITY0 ~VIC0VECTPRIORITY31 寄存器的bit[3:0]用于设置中断优先级，设置范围为0~ 15。

S5PV210中，对中断优先级的设置较S3C240简单很多，只要将相应的比特位赋值即可。