

第4章 系统配置和中断模块

本章介绍LF240x DSP的**系统配置寄存器**和**中断模块**。

系统配置：用来对DSP片内的功能模块进行用户配置，根据具体用途来进行模块定制。

中断模块：主要包括—中断优先级和中断向量表、外设中断扩展控制器(PIE)、中断向量、中断响应的流程、中断响应的时间、CPU中断寄存器、外设中断寄存器、复位、无效地址检测、外部中断控制寄存器。

4.1 系统配置寄存器

4.1.1 系统控制和状态寄存器

1. 系统控制和状态寄存器1（SCSR1）

SCSR1映射到数据存储器空间的7018h，各位如下：

位15： 保留

位14： CLKSRC，为CLKOUT引脚输出时钟源的选择位

0—CLKOUT引脚输出CPU时钟；

1—CLKOUT引脚输出WDCLK时钟。

位13、12： LPM1，LPM0，低功耗模式选择，指明在执行IDLE指令后进入哪一种低功耗模式。

00—进入 IDLE1 (LPM0) 模式;

01—进入 IDLE2 (LPM1) 模式;

1x—进入 HALT (LPM2) 模式。

位11—9: CLK PS2—CLK PS0, (PLL) 时钟预定标选择位, 选择输入时钟频率 f_{in} 的倍频系数, 如表4.1 (P32) 所示。

000—4;

001—2;

010—1.33;

011—1;

100—0.8;

101—0.66;

110—0.57;

111—0.5

位8：保留

位7：ADC CLKEN, **ADC模块**时钟使能控制位

0—禁止ADC模块时钟（节能）

1—使能ADC模块时钟，且正常运行

位6：SCI CLKEN, **SCI模块**时钟使能控制位

0—禁止SCI模块时钟（节能）

1-使能SCI模块时钟，且正常运行

位5: SPICLKEN, SPI模块时钟使能控制位

0-禁止SPI模块时钟（节能）

1-使能SPI模块时钟，且正常运行

位4: CANCLKEN, CAN模块时钟使能控制位

0-禁止CAN模块时钟（节能）

1-使能CAN模块时钟，且正常运行

位3: EVBCLKEN, EVB模块时钟使能控制位

0-禁止EVB模块时钟（节能）

1-使能EVB模块时钟，且正常运行

位2: EVACLKEN, EVA模块时钟使能控制位

0-禁止EVA模块时钟（节能）

1-使能EVA模块时钟，且正常运行

位1：保留

位0：ILLADR，无效地址检测位

检测到无效地址时，该位置1。置1后需软件来清0，即向该位写0即可。复位时该位为0。

注意：任何无效的地址会导致不可屏蔽中断（NMI）事件发生。

2. 系统控制和状态寄存器2 (SCSR2)

SCSR2被映射到数据存储器空间7019h，各位意义如下

位15-7：保留位

位6：I/P QUAL，时钟输入限定，它限定输入到LF240x的CAP1-6、XINT1-2、ADCSOC、PDPINTA*/

PDPINTB*引脚上的最小脉冲宽度。脉冲宽度只有达到这个宽度之后，内部的输入状态才会改变。

0-锁存脉冲至少需要5个时钟周期；

1-锁存脉冲至少需要11个时钟周期。

如这些引脚作I/O使用，则不使用输入时钟限定电路。

位5: WD保护位, 该位**可用来禁止WD工作**。该位是一个只能清除的位, 复位后=1。通过向该位写1对其清0。

0-保护WD, **防止WD被软件禁止**。

1-复位时的默认值, **禁止WD工作**。

位4: XMIF HI-Z。该位控制外部存储器接口信号(XMIF)。

0-所有XMIF信号为正常驱动模式(非高阻态)。

1-所有XMIF信号处于高阻态。

注意: 该位仅对LF2407/ LF2407A型号有效, 对其它型号为保留位

位3: (使能位)。这位反映了BOOTEN*引脚在复位时的状态。

0-使能引导ROM。地址0000h-00FFh被片内引导ROM块占用。禁止用FLash存储器。

1-禁止引导ROM。LF2407片内Flash程序存储器映射地址范围为0000h-7FFFh。

位2: MP/MC* (微处理器 / 微控制器选择)。

0: DSP设置为微控制器方式，片内FLASH映射到程序存储器空间，地址为0000h-7FFFh。

1: DSP设置为**微处理器方式**，**程序空间0000h-7FFFh**被映射到片外程序存储器空间（必须外扩外部程序存储器）

位1-0: SARAM的程序/数据空间选择

00-地址空间不被映射，该空间被分配到外部存储器

01-SARAM 被映射到片内程序空间

10-SARAM 被映射到片内数据空间

11-SARAM 被映射到片内程序空间，又被映射到片内数据空间

4.1.2 器件标识号寄存器(DINR)

映射到数据存储器空间701Ch。各位意义：

位15-4：DIN15-DIN4。为DSP器件的器件标识号(DIN)。

位3-0: DIN3-DIN0。为所用DSP的器件的**版本、给定值**。

不同型号的DSP所对应的DIN15-DIN0的值如下:

器件	版本	DIN15-DIN0
LF2407	1.0-1.5	0510h
LF2407	1.6	0511h
LF2407A	1.0	0520h
LC2406A	1.0	0700h
LC2402A	1.0	0610h

4.2 中断优先级和中断向量表

LF2407 DSP具有3个不可屏蔽中断和6个级别的可屏蔽中断（INT1-INT6）。

对多个外设的中断需求采用了中断扩展设计来满足。在每级可屏蔽中断（INT1-INT6）中又有多个中断源，有唯一的中断入口地址向量。

表4.2 (P34) — 不可屏蔽中断源的优先级和中断入口地址向量表。

表4.3 (P34) — 可屏蔽中断源的优先级和中断入口地址向量表。

表4.2 不可屏蔽中断(3个)

中断优先级	中断名称	外设中断向量	描述
1	Reset	0000h	复位引脚和WD溢出
2	保留	0026h	仿真陷阱
3	NMI	0004h	软件中断

表4.3 INT1 (级别1)

中断优先级	中断名称	外设中断向量	描述
4	PDPINTA	0020h	功率驱动保护中断
5	PDPINTB	0019h	功率驱动保护中断
6	ADCINT	0004h	高优先级ADC中断
7	XINT1	0001h	高优先级外中断
8	XINT2	0001h	高优先级外中断
9	SPINT	0005h	高优先级SPI中断
10	RXINT	0006h	高优先级SCI接收中断
11	TXINT	0007h	高优先级SCI发送中断
12	CANMBINT	0040h	高优先级CAN邮箱中断
13	CANERINT	0041h	高优先级CAN错误中断

INT2 (级别2)

中断优先级	中断名称	外设中断向量	描述
14	CMP1INT	0021h	比较器1中断
15	CMP2INT	0022h	比较器2中断
16	CMP3INT	0023h	比较器3中断
17	T1PINT	0027h	定时器1周期中断
18	T1CINT	0028h	定时器1比较中断
19	T1UFINT	0029h	定时器1下溢中断
20	T1OFINT	0029h	定时器1上溢中断
21	CMP4INT	0024h	比较器4中断
22	CMP5INT	0025h	比较器5中断
23	CMP6INT	0026h	比较器6中断

续 INT2 (级别2)

中断优先级	中断名称	外设中断向量	描述
24	T3PINT	002Fh	定时器3周期中断
25	T3CINT	0030h	定时器3比较中断
26	T3UFINT	0031h	定时器3下溢中断
27	T10FINT	0032h	定时器3上溢中断

INT3 (级别3)

中断优先级	中断名称	外设中断向量	描述
28	T2PINT	002Bh	定时器2周期中断
29	T2CINT	002Ch	定时器2比较中断
30	T2UFINT	002Dh	定时器2下溢中断
31	T2OFINT	002Eh	定时器2上溢中断
32	T4PINT	0039h	定时器4周期中断
33	T4CINT	003Ah	定时器4比较中断
34	T4UFINT	003Bh	定时器4下溢中断
35	T4OFINT	003Ch	定时器4上溢中断

INT4 (级别4)

中断优先级	中断名称	外设中断向量	描述
36	CAP1INT	0033h	比较器1中断
37	CAP2INT	0034h	比较器2中断
38	CAP3INT	0035h	比较器3中断
39	CAP4INT	0036h	比较器1中断
40	CAP5INT	0037h	比较器2中断
41	CAP6INT	0038h	比较器3中断

INT5 (级别5)

中断优先级	中断名称	外设中断向量	描述
42	SPINT	0005h	低优先级SPI中断
43	RXINT	0006h	低优先级SCI接收中断
44	TXINT	0007h	低优先级SCI发送中断
45	CANMBINT	0040h	低优先级CAN邮箱中断
46	CANERINT	0040h	低优先级CAN错误中断

INT6 (级别6)

中断优先级	中断名称	外设中断向量	描述
47	ADCINT	0004h	低优先级ADC中断
48	XINT1	0001h	低优先级外中断
49	XINT2	0011h	低优先级外中断
N/A	TRAP	0022h	陷阱中断
N/A	假中断向量	0000h	假中断向量

4.3 外设中断扩展控制器

LF240x CPU内核提供给用户:6级可屏蔽中断INT1-INT6。

每1级别又包含多个外设中断请求，所以用一个外设中断扩展（PIE）控制器专门来管理来自各种外设或外部引脚的数十个中断请求。

图4.4为外设中断扩展模块图。

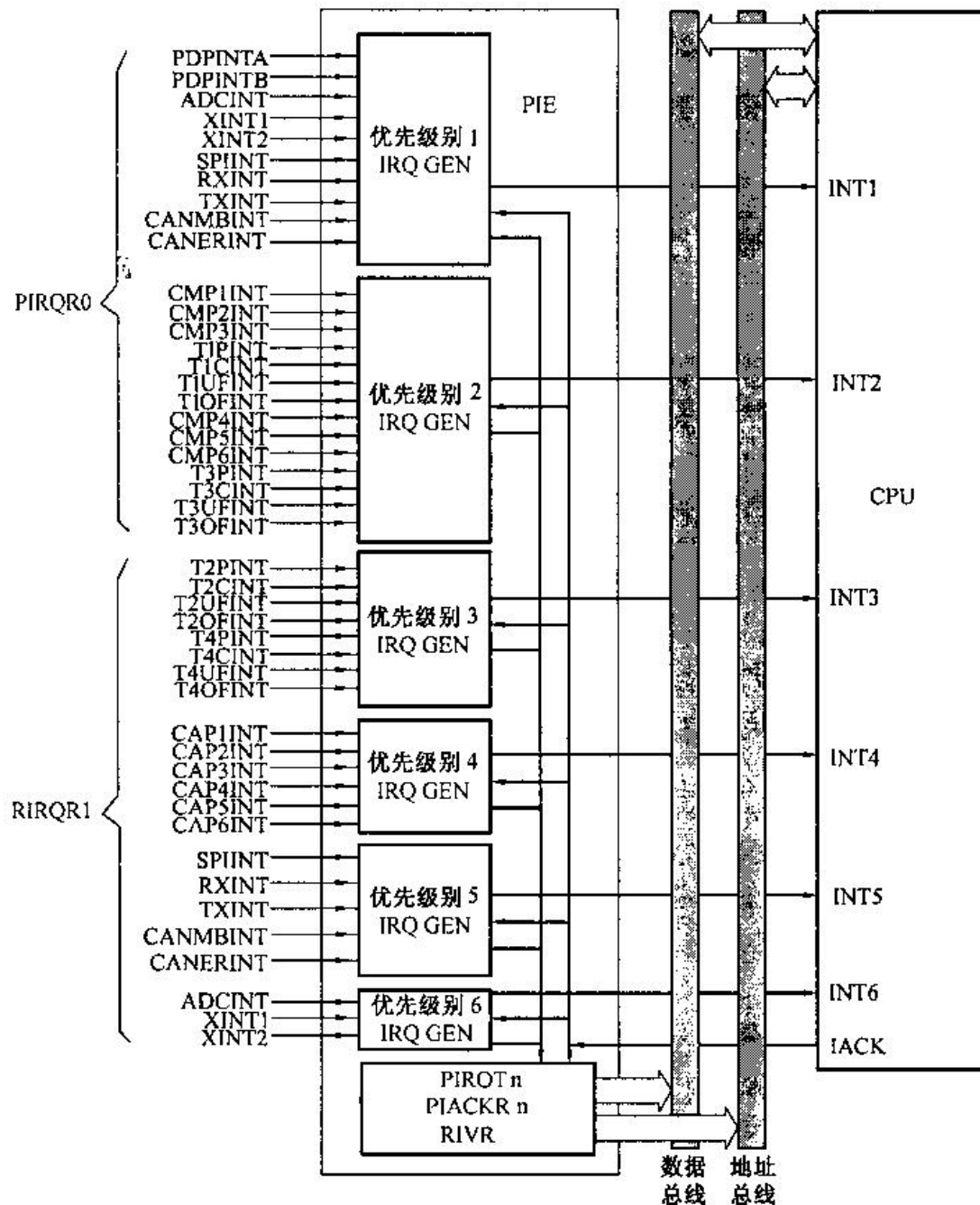


图 4.4 外设中断扩展模块图

4.3.1 中断请求层次和结构

外设中断个数很多，用一个**两级中断结构**来扩展可响应的中断个数。

中断请求/应答**硬件逻辑**和**中断服务程序软件**都有**两级层次的中断**。

在**低层次中断**，从几个外设来的外设中断请求(PIRQ)在中断控制器处进行或运算，产生一个**INT_n** ($n=1-6$) **中断请求**。

在**高层次中断**，从INT_n中断请求产生一个**到CPU的中断请求**。

如果一个引起中断的外设事件发生且相应的中断使能位被置1，则会产生一个外设到中断控制器的中断请求。

如果一个外设既可产生高级的中断请求，又可产生低级中断请求（如SCI、SPI、ADC等），对应的中断优先级位的值也被送到PIE来进行判断。

中断请求(PIRQ)标志位一直保持到中断应答自动清除或用软件将其清除。

在高层次中断，或逻辑运算的多个外设中断请求INT_n产生一个到CPU的中断请求，它是2个CPU时钟脉冲宽的低电平脉冲。

当多个外设同时发出中断请求时，CPU总是响应优先级高的中断请求。

注意： 外设中断请求标志位是在CPU响应中断时自动清除，即在高层次中断时清0，而不是在低层次中断时，清0。

4.3.2 中断向量

当CPU接受中断请求时，它并不知道是哪一外设事件引起的中断请求。

为了使CPU能够区别不同外设引起的中断事件，需经PIE译码，决定哪个中断请求被响应。

某个外设的中断请求有效时，都会产生唯一的外设中断向量，被装载到外设中断向量寄存器(PIVR)。

CPU应答外设中断请求时，从PIVR中读取相应的中断向量，并产生一个转到该中断服务子程序(GISR)入口的向量。

LF240x有两个中断向量表，CPU向量表和外设向量表。

CPU向量表用来得到响应中断请求的一级通用中断服务子程序(GISR)。

外设向量表用来获取响应某外设事件的特定中断服务子程序(SISR)。

在一级通用中断服务子程序GISR中可读出PIVR中的值，保护现场后，用PIVR中的值来产生一个转到SISR的向量。例如，可屏蔽中断XINT1(见表4.3，高级模式级别为INT1，优先级为7)产生一个中断请求，CPU对其响应。这时，0001h(XINT1的外设

中断向量) 被装载到PIVR中, CPU获取被装载到PIVR中的值之后, 用这个值来判断是哪一个外设引起的中断, 接着转移到相应的SISR。

将PIVR中的值装载入累加器时需先左移, 再加上一个固定的偏移量, 然后程序转到累加器指定的地址入口, 这个地址将指向SISR, 从而执行XINT1的中断服务子程序。

1. 假中断向量

如果一个中断应答被响应，但没有获得相应的外设的中断请求，那么就使用假中断。

假中断向量特性可以保证中断系统的完整性，从而使中断系统一直可靠安全地运行，而不会进入无法预料的中断死循环中。

以下两种情况会产生假中断：

(1) CPU执行一个软件中断指令 INTR，使用参数1-6，

用于请求服务6个可屏蔽中断（INT1-INT6）之一。

(2) 当外设发出中断请求，但是其INT_n标志位却在CPU应答请求之前已经被清0。

在上述两种情况下，并没有外设中断请求送到中断控制器，因此中断控制器不知道哪个外设中断向量装入到PIVR，此时向PIVR中装入假中断向量0000h。从而避免程序进入中断死循环中。

2. 软件层次

中断服务子程序有两级：通用中断服务子程序(GISR)

和特定中断服务子程序(SISR)。

在GISR中保存必要的上下文，从外设中断向量寄存器(PIVR)中读取外设中断向量，这个向量用来产生转移到SISR的地址入口。

程序一旦进入特定中断服务子程序后，所有的可屏蔽中断都被屏蔽。

外设中断扩展（PIE）**不包括**象复位和NMI这样的**不可屏蔽中断**。

3. 不可屏蔽中断

LF240x DSP 无NMI引脚，在**访问无效的地址时**，不可屏蔽中断(NMI)就会发出请求。当NMI被响应后，程序将转到**不可屏蔽中断向量入口地址0024h**（见表4.2）处。

4.3.3 全局中断使能

状态寄存器ST0中有一个全局中断使能位INTM，在初始化程序和主程序中，常常需要使用该位对DSP的全局中断进行打开和关闭操作。特别是初始化过程中，需要关全局中断，而在主程序开始执行时，需要开全局中断。关全局中断和开全局中断的汇编语言指令如下：

SETC INTM ; 把INTM位置1，关全局中断

CLRC INTM ; 把INTM位清0，开全局中断

执行完中断服务子程序后，一定要打开全局中断。

因为进入中断服务程序时，系统自动关中断。所以从
中断返回时需要重新打开全局中断。**注意：**不允许
中断嵌套。

4.4 中断响应的过程

下面介绍某一外设中断请求的响应过程。

- (1) 某一外设发出中断请求。
- (2) 如该外设的中断请求标志位（IF）为1，且该外设的中断使能位（IE）为1，则产生一个到PIE控制器的中断请求；如果中断没有被使能，则中

断请求标志位（IF）为1的状态保持到被软件清0。

（3） **如果不存在相同优先级（INTn）的中断请求**，那么PIRQ会使PIE控制器产生一个到CPU的中断请求（INTn），为2个CPU时钟宽度的低电平脉冲。

（4） CPU的中断请求设定**CPU的中断标志寄存器（IFR）**，如果CPU中断已被使能，中断屏蔽寄存器（IMR）CPU会中止当前的任务，将INTM置1，以屏蔽所有可屏蔽的中断，保存上下文，并且开始为高

优先级的中断（INT_n）执行通用中断服务子程序

（GISR）。CPU自动产生一个中断应答，并向与被响应的高优先级中断的相应程序地址总线（PAB）送一个中断向量值。例如，如果INT₂被响应了，它的中断向量0004h被装入PAB。

（5）外设中断扩展（PIE）控制器会对PAB的值进行译码，并产生一个外设响应应答，清除与被应答的CPU中断相关的PIRQ位。

外设中断扩展控制器然后将相应的中断向量（或假中断向量）载入**外设中断向量寄存器（PIVR）**。当GISR已经完成了现场保护，然后就可**读入PIVR**，并使用中断向量，**使程序转入到特定中断服务子程序（SISR）的入口处去执行。**

4.5 中断响应的等待时间

中断响应的等待时间包括： 外设同步接口时间、CPU响应时间、ISR转移时间。

- (1) **外设同步接口时间**是指**PIE**识别出外设发来的中断请求，经判断优先级、转换后将**中断请求**发送至**CPU的时间**。
- (2) **CPU的响应时间**指的是CPU识别出已经被使能的中断请求、响应中断、清除流水线、并且开始捕获来自CPU中断向量的第一条指令所花费的时间。**最小的CPU的响应时间是4个CPU指令周期**。
- (3) **ISR转移时间**是指为了转移ISR中特定部分而必须执行一些转移所花费的时间。该时间长短根据用户所实现的ISR的不同而有所变化。

4.6 CPU的中断寄存器

CPU中断寄存器包括：(1) 中断标志寄存器 (IFR)；(2) 中断屏蔽寄存器 (IMR)。

4.6.1 CPU中断标志寄存器(IFR)

IFR映射到数据存储器空间为0006h。各位意义如下：

位15-6：保留位。

位5-0：分别为INT6-INT1的中断标志位。

0—无INT_n (n=1-6) 的中断挂起，

1—表示有INT_n (n=1-6) 的中断挂起。

中断标志寄存器包含了所有可屏蔽中断INT6-INT1的标志位。

当一个外设发出可屏蔽中断请求时，中断标志寄存器的相应标志位被置1。如果该外设对应中断屏蔽寄存器中的**中断使能位**也为1，则该中断请求被送到CPU，此时该中断正被挂起或等待响应。

读取IFR可以识别挂起的中断，**向相应的IFR位写1将清除已挂起的中断**。

CPU响应中断或复位都能将IFR标志清除。

在对IFR操作时应注意以下几点：

- (1) 要想清除某一IFR位，必须向该位写1，而不是0；
- (2) 当一个可屏蔽中断被响应时，只有IFR位被清除，而相应的外设控制寄存器中的中断请求标志位不会被清除。如果需要清除这些标志位，应该使用软件来清除。
- (3) 当通过INTR指令来请求中断，且相应的IFR位被置1时，CPU不会自动清除该位，该位必须由软件来清除。

(4) IFR和IMR控制的是核心级的中断，所有外设在它们各自的配置/控制寄存器都有相应的中断屏蔽和标志位。

4.6.2 CPU中断屏蔽寄存器(IMR)

IMR映射在数据存储器空间中的地址为0004h，各位意义如下：

IMR中包含所有可屏蔽中断级（INT1-INT6）的屏蔽位，读IMR可以识别出已屏蔽或使能的中断级，而向IMR中写，则可屏蔽中断或使能中断。为了使能中断，

应设置相应的IMR位为1，而屏蔽中断时只需将相应的IMR位设为0。

位15-6：保留位。

位5-0：分别为INT6-INT1中断的屏蔽位。

0—中断INT_n被屏蔽。

1—中断INT_n被使能。

4.6.3 外设中断寄存器

外设中断寄存器包括如下几种：

- 外设中断请求寄存器0 (PIRQR0)
- 外设中断请求寄存器1 (PIRQR1)
- 外设中断请求寄存器2 (PIRQR2)
- 外设中断应答寄存器0 (PIACKR0)
- 外设中断应答寄存器1 (PIACKR1)
- 外设中断应答寄存器2 (PIACKR2)

外设中断请求寄存器和外设中断应答寄存器都属于外设中断扩展模块用来向CPU产生INT1-INT6中断请求的内部寄存器。这些寄存器用户只能对其读。

1. 外设中断向量寄存器 (PIVR)

外设中断向量寄存器 (PIVR) 映射在数据存储器空间中的地址为 **701Eh**，该寄存器的16位V15—V0，为最近一次被应答的外设中断的地址向量。

2. 外设中断请求寄存器0 (PIRQR0)

外设中断请求寄存器0 (PIRQR0) 映射在数据存储器空间中的地址为 **7010h**，寄存器的格式如下：

位15—0：外设请求标志位 IRQ0.15—IRQ0.0

0—无相应外设的中断请求

1—相应外设的中断请求被挂起

注：写入1会发出一个中断请求到DSP核，写入0无影响。

该寄存器16个位所对应的外设如表4.4所示。

3. 外设中断请求寄存器1 (PIRQR1)

外设中断请求寄存器1 (PIRQR1)映射在数据存储器空间中的地址为7011h，该寄存器的格式如下：

位15： 保留位，读出为0，写入无影响。

位14—0： 外设请求标志位IRQ1.14—IRQ1.0

0—无相应外设的中断请求

1—相应外设的中断请求被挂起

注：写入1会发出一个中断请求到DSP核，写入0无影响。

该寄存器16个位所对应的中断如表4.5所示。

4. 外设中断请求寄存器2 (PIRQR2)

外设中断请求寄存器2 (PIRQR2) 映射在数据存储器空间中的地址为7012h，寄存器的格式如下：

位15： 保留位

位14—0： 外设请求标志位IRQ2.14—IRQ2.0

0—无相应外设的中断请求

1—相应外设的中断请求被挂起

注：写1会发出一个中断请求到DSP核，写0无影响。

该寄存器16个位所对应的中断如表4.6所示。

5. 外设中断应答寄存器0 (PIACKR0)

外设中断应答寄存器0 (PIACKR0) 映射在数据存储器空间中的地址为**7014h**，寄存器的格式如下：

该寄存器16个位所对应的中断如**表4.6**所示。

6. 外设中断应答寄存器1 (PIACKR1)

外设中断应答寄存器1 (PIACKR1) 的映射地址**7015h**，格式如下：

该寄存器各个位所对应的中断如**表4.7**所示。

7. 外设中断应答寄存器2 (PIACKR2)

外设中断应答寄存器2 (PIACKR2) 的映射地址**7016h**,

该寄存器的格式如下:

该寄存器各个位所对应的中断如**表4.8**所示。

4.7 复位和无效地址检测

4.7.1 复位

LF2407 DSP器件有**两个复位来源**:

- (1) **外部复位引脚**的电平变化引起的复位;
- (2) **看门狗定时器溢出**引起的复位。

复位时，复位引脚被设置为输出方式，且被驱动为低，向外部电路表明LF240x器件正在自己复位。

4.7.2 无效地址检测

无效地址是不可执行的地址（例如：外设存储器映射中的保留寄存器）。

LF240x一旦检测到对无效地址的访问，就将系统控制和状态寄存器1 (SCSRI) 中的无效地址标志位 (ILLADR) 置1，从而产生一个不可屏蔽中断 (NMI)。无论何时检测到对无效地址的访问，都会产生插入一个无效地址条件，

无效地址标志位 (ILLADR) 在无效地址条件发生之后被置1，并一直保持，直到软件将其清除。

产生原因：是不正确的数据页面初始化。

4.8 外部中断控制寄存器

寄存器XINT1CR和XINT2CR为用来控制和监视XINT1和XINT2两个引脚状态的两个外部中断控制寄存器。在LF240x 中，XINT1和XINT2引脚必须被拉为低电平至少6个(或12个) CLKOUT周期才能被CPU内核识别。

4.8.1 外部中断1控制寄存器XINT1CR

映射到数据存储器空间的7070h，格式如下：

位15：XINT1标志位。在XINT1引脚上是否检测到一个所选择的中断跳变，无论中断是否使能，该位都可被置1。

0—没有检测到跳变

1—检测到跳变

位14-3：保留位

位2: XINT1极性。该读 / 写位决定是在XINT1引脚信号的上升沿还是下降沿产生中断。

0—在下**下降沿**产生中断

1—在上**上升沿**产生中断

位1: XINT1优先级。该读 / 写位决定哪一个中断优先级被请求。

0—**高**优先级

1—**低**优先级

位0: XINT1使能位。该读 / 写位可使能或屏蔽外部中断XINT1

0—屏蔽中断

1—使能中断

4.8.2 外部中断2控制寄存器XINT2CR

映射到数据存储器空间的地址为7071h，各位意义：

位15： XINT2中断请求标志位。该位表示在XINT2引脚上是否检测到一个中断请求跳变，无论该中断是否使能，该位都可以被置1。当XINT2的中断请求被应答时，该位被自动清0。

0—没有检测到跳变

1—检测到跳变

软件向该位写1(写0无效)或器件复位时，该位也被清0

位14-3： 保留位。

位2: XINT2极性。该位决定XINT2引脚信号的上升沿还是下降沿产生中断。

0—在下降沿产生中断

1—在上升沿产生中断

位1: XINT2的中断优先级。

0—高优先级

1—低优先级

位0: XINT2的中断使能位。

0—屏蔽该中断

1—使能该中断

4.9 实现可屏蔽中断的例程

通过一个例程来介绍如何编写实现可屏蔽中断的中断

服务子程序，其中包括**如何来建立中断向量表**。

LF2407有**6个**可屏蔽的中断级INT1-INT6，每一级又有

多个中断源。为正确的响应中断，由前面的介绍，

可以分**两步**来完成中断服务子程序。

本例以通用定时器1产生**1ms**的定时为例来说明如何实

现可屏蔽中断。1ms的定时器1的周期中断（就是当

计数器T1CNT计数计到与**周期寄存器T1PR**中的数相等

时，即产生中断请求。

通用定时器1的周期中断请求被响应时，CPU转移至对应INT2级别的一级通用中断服务子程序GISR2执行程序。

首先保护现场，再读取被锁存在外设中断向量寄存器（PIVR）中的该中断事件的向量地址偏移量，根据偏移量再跳到相应的中断服务子程序SISR入口。

1. 中断向量表的建立

通用定时器1的周期中断属于INT2级别。由于每一个中断都有一个其特定的中断向量入口地址，在建立

中断向量表的时候一定要注意要把所有的中断向量都列出来，否则在寻找入口地址时要出错。

下面的中断向量表列出了TMSLF2407的所有中断向量。INT2的中断向量为0004h，通用定时器1的周期中断T1PINT的外设中断向量为0027h。

.include "F2407REGS.h" ; 引用头文件

.def _c_int0

;
(1) 建立中断向量表

.sect ".vectors" ; 定义主向量段

RSVECT B _c_int0 ; **PM0** Reset Vector 1

INT1 B PHANTOM ; **PM2** Int level 1 4

INT2 B **GISR2** ; **PM4** Int level 2 5

INT3 B PHANTOM ; **PM6** Int level 3 6

INT4 B PHANTOM ; **PM8** Int level 4 7

INT5	B	PHANTOM	;	PMA	Int level 5	8
INT6	B	PHANTOM	;	PMC	Int level 6	9
RESERVED	B	PHANTOM	;	PME	(Analysis Int)	10
SW_INT8	B	PHANTOM	;	PM10	User S/W int	—
SW_INT9	B	PHANTOM	;	PM12	User S/W int	—
SW_INT10	B	PHANTOM	;	PM14	User S/W int	—
SW_INT11	B	PHANTOM	;	PM16	User S/W int	—

SW_INT12B	PHANTOM;	PM18	User S/W	int	—
SW_INT13B	PHANTOM;	PM1A	User S/W	int	—
SW_INT14B	PHANTOM;	PM1C	User S/W	int	—
SW_INT15B	PHANTOM;	PM1E	User S/W	int	—
SW_INT16B	PHANTOM;	PM20	User S/W	int	—
TRAP B	PHANTOM;	PM22	Trap	vectors	—
NMI B	PHANTOM;	PM24	User S/W	int	3
EMU_TRAP B	PHANTOM ;	PM26			2
SW_INT20	B PHANTOM ;	PM28	User S/W	int	—
SW_INT21	B PHANTOM ;	PM2A	User S/W	int	—

SW_INT22B	PHANTOM;	PM2C	User S/W	int	—
SW_INT23B	PHANTOM;	PM2E	User S/W	int	—
SW_INT24B	PHANTOM;	PM30	User S/W	int	—
SW_INT25B	PHANTOM;	PM32	User S/W	int	—
SW_INT26B	PHANTOM;	PM34	User S/W	int	—
SW_INT27B	PHANTOM;	PM36	User S/W	int	—
SW_INT28B	PHANTOM;	PM38	User S/W	int	—
SW_INT29B	PHANTOM;	PM3A	User S/W	int	—
SW_INT30B	PHANTOM;	PM3C	User S/W	int	—
SW_INT31B	PHANTOM;	PM3E	User S/W	int	—

.sect “.pvecs”; 定义各外设子向量段

PVECTORS	B	PHANTOM	;	子向量的地址偏移为0000h
	B	PHANTOM	;	子向量的地址偏移为0001h
	B	PHANTOM	;	子向量的地址偏移为0002h
	B	PHANTOM	;	子向量的地址偏移为0003h
	B	PHANTOM	;	子向量的地址偏移为0004h
	B	PHANTOM	;	子向量的地址偏移为0005h
	B	PHANTOM	;	子向量的地址偏移为0006h

B PHANTOM ; 子向量的地址偏移为0007h

B PHANTOM ; 子向量的地址偏移为0008h

B PHANTOM ; 子向量的地址偏移为000Ah

B PHANTOM ; 子向量的地址偏移为000Bh

B PHANTOM ; 子向量的地址偏移为000Ch

B PHANTOM ; 子向量的地址偏移为000Dh

B PHANTOM ; 子向量的地址偏移为000Eh

B PHANTOM ; 子向量的地址偏移为000Fh

B PHANTOM ; 子向量的地址偏移为0010h

B PHANTOM ; 子向量的地址偏移为0011h

B PHANTOM ; 子向量的地址偏移为0012h

B PHANTOM ; 子向量的地址偏移为0013h

B PHANTOM ; 子向量的地址偏移为0014h

B PHANTOM ; 子向量的地址偏移为0015h

B PHANTOM ; 子向量的地址偏移为0016h

B PHANTOM ; 子向量的地址偏移为0017h

B PHANTOM ; 子向量的地址偏移为0018h

B PHANTOM ; 子向量的地址偏移为0019h

B PHANTOM ; 子向量的地址偏移为001Ah

B PHANTOM ; 子向量的地址偏移为001Bh

B PHANTOM ; 子向量的地址偏移为001Ch

B PHANTOM ; 子向量的地址偏移为001Dh

B PHANTOM ; 子向量的地址偏移为001Eh

B PHANTOM ; 子向量的地址偏移为001Fh

B PHANTOM ; 子向量的地址偏移为0020h

B PHANTOM ; 子向量的地址偏移为0021h

B PHANTOM ; 子向量的地址偏移为0022h

B PHANTOM ; 子向量的地址偏移为0023h

B PHANTOM ; 子向量的地址偏移为0024h

B PHANTOM ; 子向量的地址偏移为0025

B PHANTOM ; 子向量的地址偏移为0026h

B T1PINT_ISR ; 子向量地址偏移为0027h, T1PINT中断

B PHANTOM ; 子向量地址偏移为0028h

B PHANTOM ; 子向量地址偏移为0029h

B PHANTOM ; 子向量地址偏移为002Ah

B PHANTOM ; 子向量地址偏移为002Bh

B PHANTOM ; 子向量地址偏移为002Dh

B PHANTOM ; 子向量地址偏移为002Eh

B PHANTOM ; 子向量地址偏移为0030h

- B PHANTOM ; 子向量的地址偏移为0031h
- B PHANTOM ; 子向量的地址偏移为0032h
- B PHANTOM ; 子向量的地址偏移为0033h
- B PHANTOM ; 子向量的地址偏移为0034h
- B PHANTOM ; 子向量的地址偏移为0035h
- B PHANTOM ; 子向量的地址偏移为0036h
- B PHANTOM ; 子向量的地址偏移为0037h
- B PHANTOM ; 子向量的地址偏移为0038h
- B PHANTOM ; 子向量的地址偏移为0039h
- B PHANTOM ; 子向量的地址偏移为003Ah

- B PHANTOM ; 子向量的地址偏移为003Bh**
- B PHANTOM ; 子向量的地址偏移为003Ch**
- B PHANTOM ; 子向量的地址偏移为003Dh**
- B PHANTOM ; 子向量的地址偏移为003Eh**
- B PHANTOM ; 子向量的地址偏移为003Fh**
- B PHANTOM ; 子向量的地址偏移为0040h**
- B PHANTOM ; 子向量的地址偏移为0041h**

； (2) 主程序

.text

_c_int0

SETC INTM ; INTM位置1, 关总中断

CLRC SXM ;

CLRC OVM ;

CLRC CNF ; B0区被配置为数据空间

LDP #0E0H ;

SPLK #81FEH, SCSR1; CLKIN=6MHz, CLKOUT=24MHz

SPLK #0E8H, WDCR ; 关闭WDT

LDP #0 ;

SPLK #02H, IMR ; 02H送IMR寄存器, 使能优先级INT2

SPLK #0FFFFH, IFR; FFFFH送IFR寄存器, 清中断标志

LDP #DP_PF2 ;

LDP #DP_EVA ;

SPLK #80H, EVAIMRA ; 使能T1PINT中断

SPLK #0FFFFh, EVAIFR ; 清EVA中断标志

SPLK #0, GPTCONA ;

SPLK #177h, TIPR ; 数177h送定时器1的周期寄存器, T1
; 1ms产生一次中断

SPLK #0, T1CNT ; T1CNT寄存器清0

SPLK #164CH, T1CON ; 数164CH送T1CON寄存器, 设置
; 定时器1输入

CLRC INTM ; INTM位清0, 开总中断

WAIT: NOP ; 循环, 等待通用定时器1的周期中断

B WAIT ;

; (3) 中断服务子程序

GISR2: ; 优先级INT2中断入口

LDP #0E0H ; 保护现场

LACC PIVR, 1 ; 读外设中断向量寄存器

; (PIVR), 并左移1位

ADD #PVECTORS ; 加上外设中断入口地址,

BACC ; 跳到相应的中断服务子程序入口

```

T1PINT_ISR: LDP    #DP_EVA        ; 通用定时器1中断服务
              ; 子程序入口
              SPLK  #0, T1CNT     ; 0送T1CNT寄存器
GISR2_RET:   ; 中断返回, 恢复现场
              CLRC  INTM         ; 开总中断, 因为一进入中
              ; 断就自动关闭总中断
              RET                ; 中断返回

```

(4) 假中断程序

PHANTOM

```

          KICK_DOG        ; 复位看门狗
          RET             ; 中断返回
          END             ; 结束

```

由程序，当定时器1的T1CNT计数到177H，即1ms时间到，就向CPU申请中断；如果这时没有其他的中断产生，则CPU接收中断申请。

先在主向量段查表查到优先级INT2的中断向量入口（GISR2），在GISR2中读取外设中断向量寄存器（PIVR）中的中断地址偏移量为0027H，再加上子向量段起始地址，通过查中断子向量段表，跳入通用定时器1的中断入口（T1PINT_ISR）。

在中断服务子程序中有对**中断现场的保护和恢复**。

中断现场主要是指DSP的状态寄存器**ST0**、**ST1**和在中断中用到的一些辅助寄存器**AR0-AR7**。

对需保护和恢复的内容，程序编写者可根据实际任务来确定。由于在本中断服务子程序仅对DP指针进行了改变，所以只需保存状态寄存器**ST0**即可。

本例程中用到**假中断PHANTOM**。假中断向量是**保持中断系统完整性的一个特性**。当一个中断请求已被响应，但却无外设将该中断向量地址的偏移量装入**外设中断向量寄存器（PIVR）**时，假中断向量（0000h）则被装入PIVR，这种缺省保证了系统按照**可控**的方式进行处理。

如果要实现其它的中断，只须将需要的中断级打开即可。