

第 8 章 中断

目录

本章包括下列主题：

8.1	简介.....	8-2
8.2	不可屏蔽陷阱	8-5
8.3	中断处理时序	8-9
8.4	中断控制和状态寄存器	8-12
8.5	中断设置过程	8-20
8.6	寄存器映射	8-21
8.7	设计技巧.....	8-23
8.8	相关应用笔记	8-24
8.9	版本历史.....	8-25

8.1 简介

PIC24F 中断控制器模块可以将众多的外设中断请求信号缩减为向 PIC24F CPU 作出的单个中断请求信号，并具有以下特性：

- 最多 8 个处理器异常和软件陷阱
- 7 个可由用户选择的优先级
- 中断向量表（IVT），最多可设置 118 个中断向量
- 每个中断或异常源对应一个惟一的向量
- 在所指定的用户优先级内具有固定的优先级
- 备用中断向量表（AIVT），用于支持调试
- 固定的中断入口和返回延时

8.1.1 中断向量表

中断向量表（IVT）驻留在程序存储器中，起始单元为 0x000004。IVT 中包含 126 个向量，其中有 8 个不可屏蔽陷阱向量，以及 118 个中断源向量。表 8-1 对陷阱向量的详情作了小结。通常，每个中断源具有其自身的向量。每个中断向量包含一个 24 位宽的地址。在每个中断向量单元中设定的值是关联的中断服务程序（ISR）的起始地址。

8.1.2 备用中断向量表

备用中断向量表（AIVT）位于 IVT 的后面，如图 8-1 所示。对 AIVT 的访问是通过 ALTIVT 控制位（INTCON2<15>）来提供的。如果 ALTIVT 位置 1，则所有的中断和异常处理将使用备用向量，而不是默认向量。备用向量的结构与默认向量的结构相同。

AIVT 提供了一种在应用程序和支持环境之间切换而无需重新设定中断向量的方法，以此支持仿真和调试。有些时候系统可能会有两个应用程序——引导加载程序和主应用程序。在这种情形中，引导加载程序可以使用其中一组向量，而主应用程序可以使用另一组向量。

该功能还可实现运行时在对不同软件算法进行评估的几个应用程序之间切换。如果不需要 AIVT，则应将 AIVT 设定为使用 IVT 中的相同地址。

8.1.3 复位序列

器件复位不是真正的异常，因为复位过程中并不需要使用中断控制器。复位时，PC 被强制设为 0，PIC24F 器件清零它的寄存器。然后，处理器从 0x000000 单元开始执行程序。用户在复位地址处设定一条 GOTO 指令，该指令会将程序重定向到相应的启动程序开始执行。关于复位的更多信息，请参见第 7 章“复位”。

注：	IVT 和 AIVT 中的所有未实现或未用的向量单元应设定为默认中断处理程序的地址，该地址中含有一条 RESET 指令。
-----------	--

图 8-1: 中断向量表

自然顺序 优先级递减 ↓	复位——GOTO 指令	0x000000	关于陷阱向量的 详细信息，请参 见表 8-1。
	复位——GOTO 地址	0x000002	
	保留	0x000004	
	振荡器故障陷阱向量		
	地址错误陷阱向量		
	堆栈错误陷阱向量		
	算术错误陷阱向量		
	保留		
	保留		
	保留		
	中断向量 0	0x000014	
	中断向量 1		
	~		
	~		
	~		
	中断向量 52	0x00007C	
	中断向量 53	0x00007E	
	中断向量 54	0x000080	
	~		
	~		
	~		
	中断向量 116	0x0000FC	
	中断向量 117	0x0000FE	
	保留		
	保留		
	保留		
	振荡器故障陷阱向量		
	地址错误陷阱向量		
	堆栈错误陷阱向量		
	算术错误陷阱向量		
	保留		
	保留		
	保留		
	中断向量 0	0x000114	
	中断向量 1		
	~		
	~		
	~		
	中断向量 52	0x00017C	
	中断向量 53	0x00017E	
	中断向量 54	0x000180	
	~		
	~		
	~		
	中断向量 116	0x0001FE	
	中断向量 117	0x000200	
	代码起始单元		

表 8-1: 陷阱向量详细信息

向量号	IVT 地址	AIVT 地址	陷阱源
0	0x000004	0x000104	保留
1	0x000006	0x000106	振荡器故障
2	0x000008	0x000108	地址错误
3	0x00000A	0x00010A	堆栈错误
4	0x00000C	0x00010C	算术错误
5	0x00000E	0x00010E	保留
6	0x000010	0x000110	保留
7	0x000012	0x000112	保留

8.1.4 CPU 优先级状态

CPU 可以工作于 16 种不同的优先级（0-15）。只有中断或陷阱源的优先级大于当前 CPU 优先级时，才会启动异常处理。外设和外部中断源可以设定为使用优先级 0-7，而 CPU 优先级 8-15 保留供陷阱源使用。陷阱是不可屏蔽的中断源，用于检测硬件和软件问题（见第 8.2 节“不可屏蔽陷阱”）。每个陷阱源的优先级都是固定的，并且一个优先级只分配一个陷阱。请注意，将中断源优先级设定为 0 实际上会禁止该中断源，因为它的优先级总是低于 CPU 优先级。

当前的 CPU 优先级由以下 4 个状态位指示：

- IPL<2:0> 状态位，位于 SR<7:5>
- IPL3 状态位，位于 CORCON<3>

IPL<2:0> 状态位是可读写位，所以用户可以通过修改这些位，禁止优先级低于给定优先级的所有中断源。例如，如果 IPL<2:0> = 111，则任何优先级设定为 0、1、2 或 3 的中断源都不能中断 CPU。

陷阱事件的优先级（8-15）高于所有的用户中断源。当 IPL3 位置 1 时，说明发生了陷阱事件。IPL3 位可以清零，但不能由用户置 1。在一些应用程序中，可能需要在发生陷阱后将 IPL3 位清零，并跳转到另一条指令，而不是跳转到原先导致发生陷阱的指令之后的指令处。

所有的用户中断源可以通过设置 IPL<2:0> = 111 而禁止。

注： 在禁止中断嵌套时，IPL<2:0> 位变为只读位。更多信息，请参见第 8.2.4.2 节“中断嵌套”。

8.1.5 中断优先级

每个外设中断源都可分配到 7 个优先级之一。每个中断的可由用户分配的中断优先级控制位位于 IPCn 寄存器中的每个半字节的低 3 位。每个半字节的 bit 3 均不使用，读为 0。这些位指定分配给特定中断的优先级。可用的优先级为优先级 1 至 7，优先级 1 是最低优先级，优先级 7 为最高优先级。如果与某个中断源相关的 IPCn 位全都清零，则实际上禁止了该中断源。

注： 在器件复位时，器件会初始化 IPCn 寄存器，为所有用户中断源都分配优先级 4。

因为可能会有多个中断请求源分配了同一个特定优先级，所以提供了一种以给定用户分配级别的方法来解决优先级冲突。每个中断源都具有一个自然顺序优先级，该优先级基于它在 IVT 中的单元位置。编号较低的中断向量具有较高的自然优先级，而编号较高的向量具有较低的自然优先级。例如，中断向量 0 具有最高的自然优先级，而中断向量 117 具有最低的自然优先级。所有等待处理的中断源的总体优先级首先根据用户在 IPCn 寄存器中分配的优先级确定，然后再根据 IVT 中的自然顺序优先级确定。

自然顺序优先级仅用于解决具有相同的用户分配优先级且同时等待处理的中断之间的冲突。在解决优先级冲突，异常处理开始之后，只有用户分配优先级更高的中断源才能中断 CPU。如果中断具有相同的用户分配优先级，但具有更高的自然顺序优先级，并在异常处理开始之后进入等待处理状态，则该中断将保持等待处理状态，直到当前异常进程完成为止。

用户可以为每个中断源分配 7 种优先级中的任意一种优先级，这意味着用户可以为具有较低自然顺序优先级的中断分配一个极高的总体优先级。例如，可以为中断向量 0 分配优先级 1，从而为其设置一个极低的有效优先级。

注： 本文档说明了通用的中断结构。关于外设和每个中断源的信息，请参见具体器件数据手册。

8.2 不可屏蔽陷阱

陷阱可以看作是`不可屏蔽的可嵌套中断`，它们使用固定的优先级结构。陷阱供用户用于在调试期间和在应用程序内操作时修正错误操作。如果用户不想在发生陷阱错误条件时执行修正操作，则必须在这些向量中装入将复位器件的软件程序的地址。否则，陷阱向量将设定为将修正陷阱条件的服务程序的地址。

PIC24F 具有 4 种不可屏蔽陷阱源：

- 振荡器故障陷阱
- 堆栈错误陷阱
- 地址错误陷阱
- 算术错误陷阱

在异常处理开始之前，允许导致陷阱的指令完成操作。因此，用户可能必须修正导致陷阱的指令的操作。

每个陷阱源具有固定的优先级，优先级根据它在 **IVT** 中的位置指定。振荡器故障陷阱具有最高的优先级，而算术错误陷阱具有最低的优先级（见图 8-1）。此外，陷阱源还分为两种不同的类别：“硬”陷阱和“软”陷阱。

8.2.1 软陷阱

算术错误陷阱（优先级 11）和堆栈错误陷阱（优先级 12）归类为“软”陷阱源。软陷阱可以如同不可屏蔽中断源一样对待，后者的优先级由其在 **IVT** 中的位置分配。软陷阱的处理方式与中断类似，在异常处理之前需要 2 个周期进行采样和响应。因此，在响应软陷阱之前可以执行另外的指令。

8.2.1.1 堆栈错误陷阱（软陷阱，优先级 12）

复位期间，堆栈初始化为 **0x0800**。只要堆栈指针地址小于 **0x0800**，就会产生堆栈错误陷阱。

有一个堆栈极限寄存器（**SPLIM**）和堆栈指针关联，后者在复位时不初始化。只有在向 **SPLIM** 写入一个字后才会使能堆栈溢出检查。

用 **W15** 生成的作为源或目标指针的所有有效地址（**EA**），都会和 **SPLIM** 中的值比较。如果 **EA** 大于 **SPLIM** 寄存器的值，则会产生堆栈错误陷阱。此外，如果 **EA** 计算值覆盖到数据空间末尾（**0xFFFF**）以外，则也会产生堆栈错误陷阱。

在软件中，可以通过查询 **STKERR** 状态位（**INTCON1<2>**）来检测堆栈错误。为了避免再次进入陷阱服务程序，在使用 **RETFIE** 指令从陷阱返回之前，必须在软件中先将 **STKERR** 状态标志清零。

8.2.1.2 算术错误陷阱（优先级 11）

如果试图将某个数除以 0，则会执行算术错误陷阱。在软件中，可以通过查询 **MATHERR** 状态位（**INTCON1<4>**）来检测算术错误陷阱。为了避免再次进入陷阱服务程序，在使用 **RETFIE** 指令从陷阱返回之前，必须在软件中先将 **MATHERR** 状态标志清零。

8.2.2 硬陷阱

硬陷阱包括优先级为 13 至 15 的异常。地址错误（优先级 13）和振荡器错误（优先级 14）陷阱归入此类。

类似于软陷阱，硬陷阱也可以看作是`不可屏蔽中断源`。硬陷阱和软陷阱之间的区别在于，在导致陷阱的指令执行完毕后，硬陷阱会强制 **CPU** 停止执行代码。只有在响应和处理了陷阱之后，才会继续进行正常的程序执行流。

8.2.2.1 陷阱优先级和硬陷阱冲突

如果在处理任意优先级较低的陷阱时，发生了优先级较高的陷阱，则将暂时停止处理优先级较低的陷阱，并响应和处理优先级较高的陷阱。优先级较低的陷阱将保持等待处理状态，直到优先级较高的陷阱处理完毕。

每次发生硬陷阱后，任何代码都必须在应答完硬陷阱后，才能继续执行。如果在某个优先级较高的陷阱等待处理、获得响应或正在进行处理时，发生了优先级较低的硬陷阱，则将发生硬陷阱冲突。发生冲突是因为只有在较高优先级的陷阱处理完毕后，才能响应优先级较低的陷阱。

在发生硬陷阱冲突条件时，器件将自动复位。发生复位时，TRAPR 状态位（RCON<15>）置 1，因此可以在软件中检测该条件。

8.2.2.2 振荡器故障陷阱（硬陷阱，优先级 14）

在使能故障保护时钟监视器（FSCM）时，如果检测到失去系统时钟源，则将产生振荡器故障陷阱事件。

在软件中，可以通过查询 OSCFAIL 状态位（INTCON1<1>）或 CF 状态位（OSCCON<3>）来检测振荡器故障陷阱事件。为了避免再次进入陷阱服务程序，在使用 RETFIE 指令从陷阱返回之前，必须在软件中先将 OSCFAIL 状态标志清零。

关于 FSCM 的更多信息，请参见第 6 章“振荡器”和第 32 章“配置”。

8.2.2.3 地址错误陷阱（硬陷阱，优先级 13）

以下几个段落描述了会导致产生地址错误陷阱的操作情形：

1. 试图取地址未对齐的数据字。在指令执行字访问操作，但有效地址的最低位却设为 1 时，将发生这种情况。PIC24F CPU 要求所有字访问操作都要与偶地址边界对齐。
2. 使用间接寻址模式执行位操作指令，但有效地址的最低位设为 1。
3. 试图从未实现的数据地址空间取数据。
4. 执行“BRA #literal”指令或“GOTO #literal”指令，但其中的 literal 是未实现的程序存储器地址。
5. 在将 PC 修改为指向未实现的程序存储器地址后执行指令。PC 可以通过在堆栈中装入一个值并执行 RETURN 指令而修改。

每当发生地址错误陷阱时，都将禁止数据空间写操作，从而使数据免遭毁坏。

在软件中，可以通过查询 ADDRERR 状态位（INTCON1<3>）来检测地址错误。为了避免再次进入陷阱服务程序，在使用 RETFIE 指令从陷阱返回之前，必须在软件中先将 ADDRERR 状态标志清零。

8.2.3 禁止中断指令

DISI（禁止中断）指令用于禁止中断，最长时间可达 16384 个指令周期。在必须执行对时间要求严格的代码段时，该指令非常有用。

DISI 指令仅禁止优先级为 1-6 的中断。使用 DISI 指令后，优先级为 7 的中断和所有陷阱事件依然可以中断 CPU。

DISI 指令与 DISICNT 寄存器配合使用。当 DISICNT 寄存器的值非零时，优先级为 1-6 的中断被禁止。DISICNT 寄存器值在每个后续的指令周期递减。当 DISICNT 寄存器计数至 0 时，将重新允许优先级为 1-6 的中断。在 DISI 指令中指定的值包括由于 PSV 访问、指令停止等操作而产生的所有周期。

DISICNT 寄存器是可读写的。用户可以通过清零 DISICNT 寄存器而提前终止先前的 DISI 指令的作用。禁止中断的时间量也可以通过对 DISICNT 进行写或加操作而升高。

请注意，如果 DISICNT 寄存器值为 0，则仅仅向寄存器写非零值不会禁止中断。此时，必须先使用 DISI 指令禁止中断。在执行 DISI 指令，DISICNT 保存非零值后，可以通过修改 DISICNT 的值来延长中断禁止时间。

注： 建议不要使用软件修改 DISICNT 寄存器。

每当执行 DISI 指令禁止中断之后，DISI 状态位（INTCON2<14>）置 1。

注： 如果所有用户中断源都未分配 CPU 优先级 7，则 DISI 指令可以用于快速禁止所有用户中断源。

8.2.4 中断操作

所有中断事件标志均在每个指令周期期间进行采样。在 IFSn 寄存器中，标志位等于 1 代表等待处理的中断请求（IRQ）。如果中断允许（IECn）寄存器中相应的位置 1，则 IRQ 会导致发生中断。在对 IRQ 进行采样的指令周期的剩余时间，将对所有等待处理的中断请求的优先级进行比较。

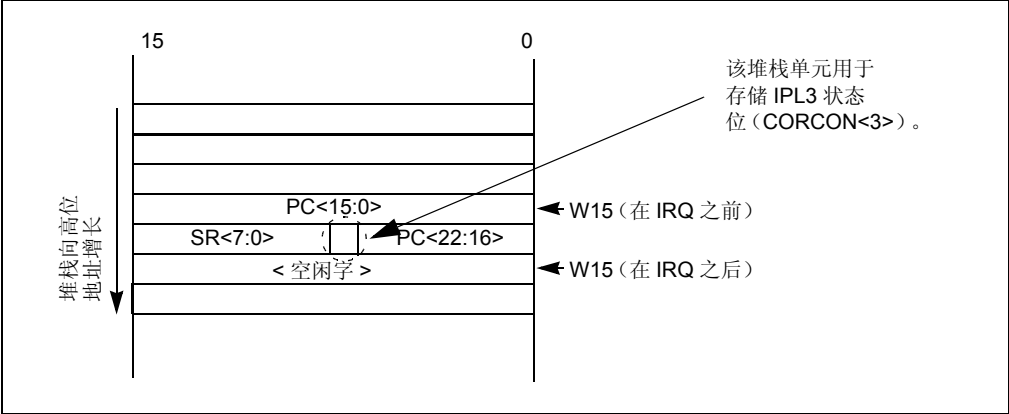
在 CPU 响应 IRQ 时，不会中止任何指令。在采样 IRQ 时正在执行的指令将先执行完毕，然后才执行 ISR。

如果有一个等待处理的 IRQ 的用户分配优先级大于当前的处理器优先级（由 IPL<2:0> 状态位 SR<7:5> 指示），则将向处理器发送中断。然后，处理器在软件堆栈中保存以下信息：

- 当前的 PC 值
- 处理器 STATUS 寄存器的低字节（SRL）
- IPL3 状态位（CORCON<3>）

保存到堆栈中的这三个值可使返回 PC 地址值、MCU 状态位和当前的处理器优先级被自动保存。在堆栈中保持以上信息之后，CPU 将等待处理的中断的优先级写入 IPL<2:0> 位单元。该操作将禁止所有优先级小于等于当前中断优先级的中断，直到使用 RETFIE 指令终止中断服务程序（ISR）为止。

图 8-2： 中断事件的堆栈操作



8.2.4.1 从中断返回

RETFIE (从中断返回) 指令会将 PC 返回地址、IPL3 状态位和 SRL 寄存器出栈, 将处理器恢复为发生中断之前的状态和优先级。

8.2.4.2 中断嵌套

在默认情况下, 中断是可嵌套的。任何正在执行的 ISR 都可以被另一个具有更高用户分配优先级的中断源中断。中断嵌套可以通过将 NSTDIS 控制位 (INTCON1<15>) 置 1 而禁止。当 NSTDIS 控制位置 1 时, 所有正在处理的中断将会通过设置 IPL<2:0> = 111 而强制设置 CPU 优先级为 7。该操作实际上会屏蔽所有其他中断源, 直到执行 RETFIE 指令为止。当禁止中断嵌套时, 除了用于解决同时等待处理的中断之间的冲突之外, 用户分配的中断优先级无其他作用。

在禁止中断嵌套时, IPL<2:0> 位变为只读位。这可以阻止用户软件将 IPL<2:0> 设置为较低值, 设置较低值的操作实际上会重新允许中断嵌套。

8.2.5 从休眠和空闲模式唤醒

任何单独允许的中断源 (使用 IECn 寄存器中的相应控制位), 都可以从休眠或空闲模式唤醒处理器。如果某个中断源的中断状态标志置 1, 而该中断源已通过 IECn 控制寄存器中的相应位允许, 则将会向 PIC24F CPU 发送唤醒信号。当器件从休眠或空闲模式唤醒时, 可能发生两种操作中的一种:

1. 如果该中断源的优先级大于当前的 CPU 优先级, 则处理器将处理该中断, 并跳转至该中断源的 ISR。
2. 如果该中断源的用户分配中断优先级小于等于当前的 CPU 优先级, 则处理器将只是继续从先前将 CPU 置为休眠或空闲模式的 PWRSAV 指令的后一条指令处开始执行代码。

注: 分配为 CPU 优先级 0 的用户中断源无法将 CPU 从休眠或空闲模式唤醒, 因为该中断源实际上被禁止。要使用某个中断作为唤醒源, 该中断的 CPU 优先级必须指定为 CPU 优先级 1 或更高。

8.2.6 A/D 转换器外部转换请求

外部中断请求引脚与 A/D 转换器共用, 转换器将它用作外部转换请求信号。中断向量 0 的中断源具有可编程的边沿极性, A/D 转换器外部转换请求功能也可使用这一特性。关于 A/D 转换器的更多信息, 请参见第 17 章 “10 位 A/D 转换器”。

8.2.7 外部中断支持

PIC24F 最多可支持 5 个外部中断引脚源 (中断向量 0 至中断向量 4)。每个外部中断引脚都具有用于检测中断事件的边沿检测电路。INTCON2 寄存器具有 5 个控制位 (INT0EP-INT4EP), 用于选择边沿检测电路的极性。每个外部中断引脚都可以设定为在发生上升沿或下降沿事件时中断 CPU。更多详细信息, 请参见寄存器 8-4。

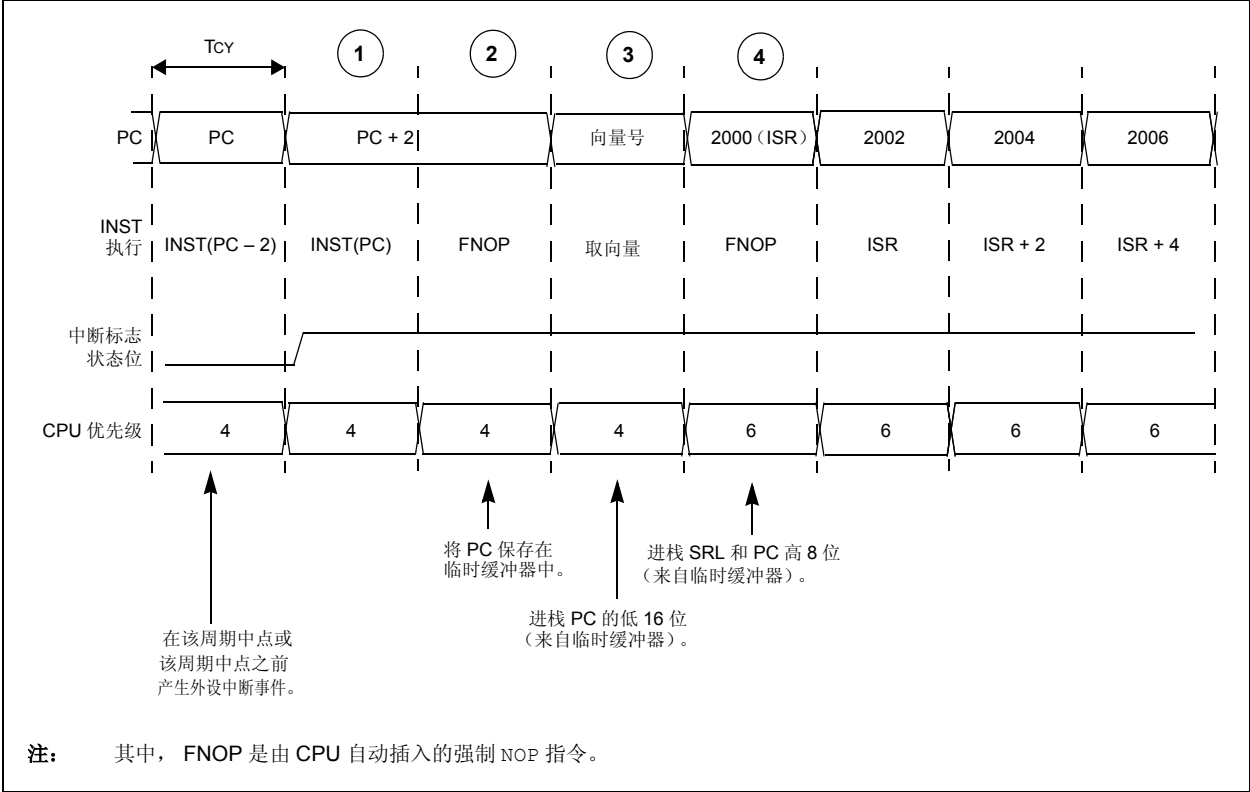
8.3 中断处理时序

8.3.1 单周期指令的中断响应延时

图 8-3 显示了在单周期指令期间产生外设中断时的事件序列。中断过程需要 4 个指令周期。图 8-3 中对每个周期进行了编号，以供参考。

在发生外设中断后，中断标志状态位在指令周期期间置 1。当前的指令在该指令周期期间完成操作。在发生中断事件后的第二个指令周期中，PC 和 SRL 寄存器的值保存到临时缓冲寄存器中。中断过程的第二个周期执行 NOP 指令，以便与双周期指令期间的序列保持一致（见第 8.3.2 节“双周期指令的中断响应延时”）。在第三个周期中，PC 中装入中断源的向量表地址，并取 ISR 的起始地址。在第四个周期中，PC 中装入 ISR 地址。在取 ISR 中的第一条指令时，第四个周期执行 NOP 指令。

图 8-3: 单周期指令期间的中断时序



8.3.2 双周期指令的中断响应延时

双周期指令期间的中断响应延时与单周期指令期间的响应延时相同。中断过程的第一个和第二个周期使双周期指令可以完成执行。图 8-5 中的时序图显示了在执行双周期指令之前的指令周期中产生外设中断事件时的情况。

图 8-6 显示了在外设中断与双周期指令的第一个周期同时发生时的时序。在此例中，中断过程与单周期指令的中断过程相同（见第 8.3.1 节“单周期指令的中断响应延时”）。

图 8-4: 双周期指令期间的中断时序

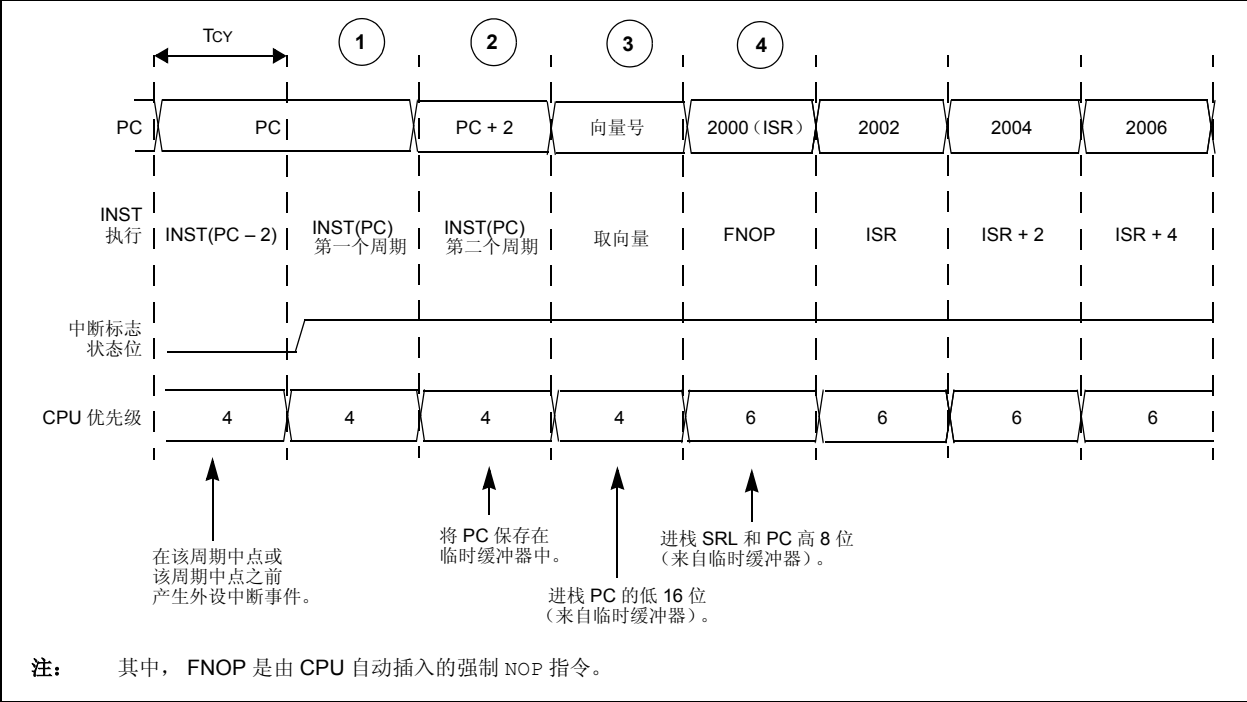
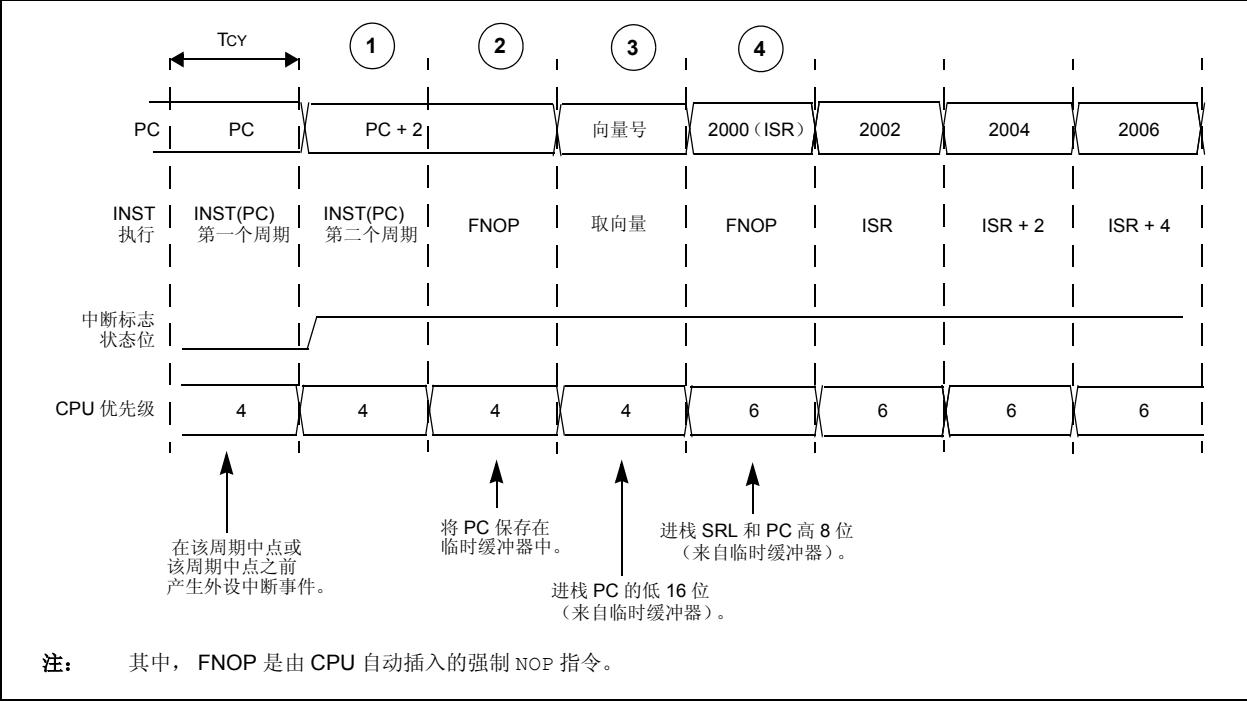


图 8-5: 中断时序，中断在双周期指令的第一个周期发生

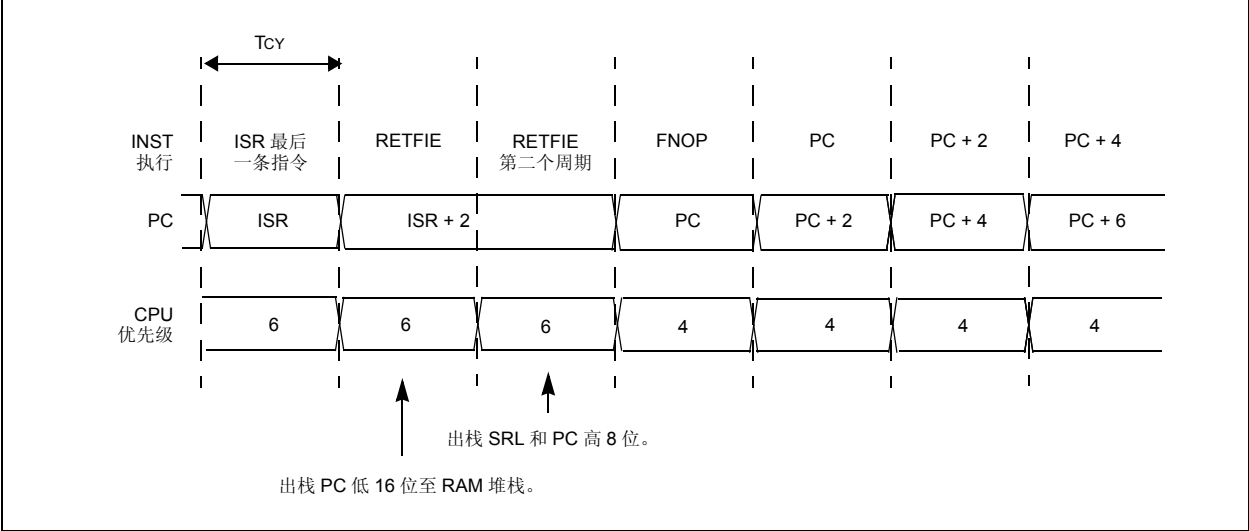


8.3.3 从中断返回

“从中断返回”指令 RETFIE 用于退出中断或陷阱程序。

在 RETFIE 指令的第一个周期期间，PC 的高位和 SRL 寄存器值从堆栈中出栈。进栈的 PC 值的低 16 位在第二个周期期间从堆栈中出栈。使用第三个指令周期取出更新后的程序计数器所寻址的指令。该周期执行 NOP 指令。

图 8-6: 从中断返回时序



8.4 中断控制和状态寄存器

以下寄存器与中断控制器相关：

- **INTCON1 和 INTCON2 寄存器**

全局中断控制功能由这两个寄存器产生。INTCON1 中含有中断嵌套禁止（NSTDIS）位，以及处理器陷阱源的控制和状态标志。INTCON2 寄存器控制外部中断请求信号行为和备用向量表的使用。

- **IFS_n：中断标志状态寄存器**

所有的中断请求标志在 IFS_n 寄存器中维护，其中“n”表示寄存器编号。每个中断源都具有一个状态位，该状态位由相应的外设或外部信号置 1，通过软件进行清零。

- **IEC_n：中断允许控制寄存器**

所有的中断允许控制位在 IEC_n 寄存器中维护，其中“n”表示寄存器编号。这些控制位用于分别允许来自外设或外部信号的中断。

- **IPC_n：中断优先级控制寄存器**

每个用户中断源都可分配到 8 个优先级之一。IPC_n 寄存器用于设置每个中断源的中断优先级。

- **SR：CPU STATUS 寄存器**

SR 不是中断控制器硬件的特定部分，但它含有 IPL<2:0> 状态位（SR<7:5>），这些状态位指示当前的 CPU 优先级。用户可以通过写 IPL 位来更改当前的 CPU 优先级。

- **CORCON：内核控制寄存器**

CORCON 不是中断控制器硬件的特定部分，但它含有 IPL3 状态位，该位指示当前的 CPU 优先级。IPL3 是只读位，所以用户软件无法屏蔽陷阱事件。

后面几页详对 SR、CORCON、INTCON1 和 INTCON2 寄存器作了说明。在以下内容中还给出了通用的中断寄存器映射。每个中断均与一个中断标志位（IF）、一个中断允许位（IE）和三个中断极性位（IP2:IP0）关联。IFS_n、IEC_n 和 IPC_n 寄存器的实际数量取决于在特定器件上实现的中断的数量。更多详细信息，请参见具体数据手册。

8.4.1 中断的控制寄存器分配

中断源按特定的顺序分配给 IFS_n、IEC_n 和 IPC_n 寄存器。例如，中断向量 0 具有的自然顺序优先级为 0。因而，中断向量 0 状态位为 IFS0<0>。中断向量 0 使用 IEC0<0> 作为它的允许位，IPC0<2:0> 位指定中断向量 0 的中断优先级。关于所有中断相关寄存器的一般概述，请参见表 8-2。

寄存器 8-1: SR: CPU STATUS 寄存器

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	DC
bit 15							bit 8
R/W-0	R/W-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
IPL2 ^(1,2)	IPL1 ^(1,2)	IPL0 ^(1,2)	RA	N	OV	Z	C
bit 7							bit 0

图注:

R = 可读位 W = 可写位 U = 未实现位, 读为 0

-n = 任何复位时的值 1 = 置 1 0 = 清零 x = 未知

bit 7-5 **IPL2:IPL0:** CPU 中断优先级状态位 ^(1,2)

111 = CPU 中断优先级是 7 (15)。禁止用户中断。

110 = CPU 中断优先级是 6 (14)

101 = CPU 中断优先级是 5 (13)

100 = CPU 中断优先级是 4 (12)

011 = CPU 中断优先级是 3 (11)

010 = CPU 中断优先级是 2 (10)

001 = CPU 中断优先级是 1 (9)

000 = CPU 中断优先级是 0 (8)

注 1: IPL<2:0> 位和 IPL<3> 位 (CORCON<3>) 组合, 形成 CPU 中断优先级。括号中的值表示 IPL<3> = 1 时的 IPL。

 2: 当 NSTDIS = 1 (INTCON1<15>) 时, IPL<2:0> 状态位是只读的。

寄存器 8-2: CORCON: 内核控制寄存器

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8
U-0	U-0	U-0	U-0	R/C-0	R/W-0	U-0	U-0
—	—	—	—	IPL3 ⁽¹⁾	PSV	—	—
bit 7							bit 0

图注:

R = 可读位 C = 可清零位 U = 未实现位, 读为 0

-n = 任何复位时的值 W = 可写位 0 = 清零 x = 未知

 1 = 置 1

bit 3 **IPL3:** CPU 中断优先级状态位 ⁽¹⁾

1 = CPU 中断优先级高于 7

0 = CPU 中断优先级等于或低于 7

注 1: IPL3 位与 IPL<2:0> 位 (SR<7:5>) 组合, 形成 CPU 中断优先级。

PIC24F 系列参考手册

寄存器 8-3: INTCON1: 中断控制寄存器 1

R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
NSTDIS	—	—	—	—	—	—	—
bit 15							bit 8
U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
—	—	—	MATHERR	ADDRERR	STKERR	OSCFAIL	—
bit 7							bit 0

图注:			
R = 可读位	W = 可写位	U = 未实现位, 读为 0	
-n = 任何复位时的值	1 = 置 1	0 = 清零	x = 未知

- bit 15 **NSTDIS:** 中断嵌套禁止位
1 = 禁止中断嵌套
0 = 使能中断嵌套
- bit 14-5 **未实现:** 读为 0
- bit 4 **MATHERR:** 算术错误陷阱状态位
1 = 发生了溢出陷阱
0 = 未发生溢出陷阱
- bit 3 **ADDRERR:** 地址错误陷阱状态位
1 = 发生了地址错误陷阱
0 = 未发生地址错误陷阱
- bit 2 **STKERR:** 堆栈错误陷阱状态位
1 = 发生了堆栈错误陷阱
0 = 未发生堆栈错误陷阱
- bit 1 **OSCFAIL:** 振荡器故障陷阱状态位
1 = 发生了振荡器故障陷阱
0 = 未发生振荡器故障陷阱
- bit 0 **未实现:** 读为 0

寄存器 8-4: INTCON2: 中断控制寄存器 2

R/W-0	R-0	U-0	U-0	U-0	U-0	U-0	U-0
ALTIVT	DISI	—	—	—	—	—	—
bit 15							bit 8
U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	INT4EP	INT3EP	INT2EP	INT1EP	INT0EP
bit 7							bit 0

图注:			
R = 可读位	W = 可写位	U = 未实现位, 读为 0	
-n = 任何复位时的值	1 = 置 1	0 = 清零	x = 未知

- bit 15 **ALTIVT:** 使能备用中断向量表位
1 = 使用备用向量表
0 = 使用标准 (默认) 向量表
- bit 14 **DISI:** DISI 指令状态位
1 = DISI 指令有效
0 = DISI 指令无效
- bit 13-5 **未实现:** 读为 0
- bit 4 **INT4EP:** 外部中断 #4 边沿检测极性选择位
1 = 负边沿处中断
0 = 正边沿处中断
- bit 3 **INT3EP:** 外部中断 #3 边沿检测极性选择位
1 = 负边沿处中断
0 = 正边沿处中断
- bit 2 **INT2EP:** 外部中断 #2 边沿检测极性选择位
1 = 负边沿处中断
0 = 正边沿处中断
- bit 1 **INT1EP:** 外部中断 #1 边沿检测极性选择位
1 = 负边沿处中断
0 = 正边沿处中断
- bit 0 **INT0EP:** 外部中断 #0 边沿检测极性选择位
1 = 负边沿处中断
0 = 正边沿处中断

PIC24F 系列参考手册

寄存器 8-5: IFSn: 中断标志状态寄存器 0 到 6 (中断向量 0 到 111) ⁽¹⁾

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
V(16n + 15)IF	V(16n + 14)IF	V(16n + 13)IF	V(16n + 12)IF	V(16n + 11)IF	V(16n + 10)IF	V(16n + 9)IF	V(16n + 8)IF
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
V(16n + 7)IF	V(16n + 6)IF	V(16n + 5)IF	V(16n + 4)IF	V(16n + 3)IF	V(16n + 2)IF	V(16n + 1)IF	V(16n)IF
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 任何复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-0 **V(16n + x)IF**: 中断向量 16n + x 的中断状态标志位 (其中 x = 位的位置编号)

1 = 已发生中断请求

0 = 未发生中断请求

注 1: 不是所有的中断向量在所有器件上都实现。请参见具体器件或器件系列的数据手册的中断向量表, 确定某一具体器件实现了哪些中断向量。

寄存器 8-6: IFSn: 中断标志状态寄存器 7 (中断向量 112 到 117) ⁽¹⁾

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0
—	—	V117IF	V116IF	V115IF	V114IF	V113IF	V112IF
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 任何复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-6 **未实现**: 读为 0

bit 5-0 **V117IF:V112IF**: 中断向量 117 到 112 的中断状态标志位

1 = 已发生中断请求

0 = 未发生中断请求

注 1: 不是所有的中断向量在所有器件上都实现。请参见具体器件或器件系列的数据手册的中断向量表, 确定某一具体器件实现了哪些中断向量。

寄存器 8-7: IECn: 中断允许寄存器 0 到 6 (中断向量 0 到 111) (1)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
V(16n + 15)IE	V(16n + 14)IE	V(16n + 13)IE	V(16n + 12)IE	V(16n + 11)IE	V(16n + 10)IE	V(16n + 9)IE	V(16n + 8)IE
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
V(16n + 7)IE	V(16n + 6)IE	V(16n + 5)IE	V(16n + 4)IE	V(16n + 3)IE	V(16n + 2)IE	V(16n + 1)IE	V(16n + 0)IE
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为 0
-n = 任何复位时的值	1 = 置 1	0 = 清零
		x = 未知

bit 15-0 V(16n + x)IF: 中断向量 16n + x 的中断允许位 (其中 x = 位的位置编号)

1 = 允许中断

0 = 禁止中断

注 1: 不是所有的中断向量在所有器件上都实现。请参见具体器件或器件系列的数据手册的中断向量表, 确定某一具体器件实现了哪些中断向量。

寄存器 8-8: IECn: 中断允许寄存器 7 (中断向量 112 到 117) (1)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	V117IF	V116IF	V115IF	V114IF	V113IF	V112IF
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为 0
-n = 任何复位时的值	1 = 置 1	0 = 清零
		x = 未知

bit 15-6 未实现: 读为 0

bit 5-0 V117IE:V112IE: 中断向量 117 到 112 的中断允许位

1 = 允许中断

0 = 禁止中断

注 1: 不是所有的中断向量在所有器件上都实现。请参见具体器件或器件系列的数据手册的中断向量表, 确定某一具体器件实现了哪些中断向量。

PIC24F 系列参考手册

寄存器 8-9: **IPCn**: 中断优先级寄存器 0 到 28 (中断向量 0 到 115) ⁽¹⁾

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	V(4n + 3)IP2	V(4n + 3)IP1	V(xn + 3)IP0	—	V(4n + 2)IP2	V(4n + 2)IP1	V(4n + 2)IP0
bit 15				bit 8			

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	V(4n + 1)IP2	V(4n + 1)IP1	V(xn + 1)IP0	—	V(4n)IP2	V(4n)IP1	V(4n)IP0
bit 7				bit 0			

图注:			
R = 可读位	W = 可写位	U = 未实现位, 读为 0	
-n = 任何复位时的值	1 = 置 1	0 = 清零	x = 未知

- bit 15 **未实现:** 读为 0
- bit 14-12 **V(4n + 3)IP2:V(4n + 3)IP0:** 中断向量 4n + 3 的中断优先级位
- 111 = 中断为优先级 7 (最高优先级中断)
-
-
-
- 001 = 中断为优先级 1
- 000 = 禁止中断源
- bit 11 **未实现:** 读为 0
- bit 10-8 **V(4n + 2)IP2:V(4n + 2)IP0:** 中断向量 4n + 2 的中断优先级位
- 111 = 中断为优先级 7 (最高优先级中断)
-
-
-
- 001 = 中断为优先级 1
- 000 = 禁止中断源
- bit 7 **未实现:** 读为 0
- bit 6-4 **V(4n + 1)IP2:V(4n + 1)IP0:** 中断向量 4n + 1 的中断优先级位
- 111 = 中断为优先级 7 (最高优先级中断)
-
-
-
- 001 = 中断为优先级 1
- 000 = 禁止中断源
- bit 3 **未实现:** 读为 0
- bit 2-0 **V(4n)IP2:V(4n)IP0:** 中断向量 4n 的中断优先级位
- 111 = 中断为优先级 7 (最高优先级中断)
-
-
-
- 001 = 中断为优先级 1
- 000 = 禁止中断源

注 1: 不是所有的中断向量在所有器件上都实现。请参见具体器件或器件系列的数据手册的中断向量表, 确定某一具体器件实现了哪些中断向量。

寄存器 8-10: **IPCn**: 中断优先级寄存器 29（中断向量 116 和 117）⁽¹⁾

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	V117IP2	V117IP1	V117IP0	—	V116IP2	V116IP1	V116IP0
bit 7							bit 0

图注:
R = 可读位 W = 可写位 U = 未实现位，读为 0
-n = 任何复位时的值 1 = 置 1 0 = 清零 x = 未知

- bit 15-7 **未实现:** 读为 0
- bit 6-4 **V117IP2:V117IP0:** 中断向量 117 的中断优先级位
 111 = 中断为优先级 7（最高优先级中断）
 •
 •
 •
 001 = 中断为优先级 1
 000 = 禁止中断源
- bit 3 **未实现:** 读为 0
- bit 2-0 **V116IP2:V116IP0:** 中断向量 116 的中断优先级位
 111 = 中断为优先级 7（最高优先级中断）
 •
 •
 •
 001 = 中断为优先级 1
 000 = 禁止中断源

注 1: 不是所有的中断向量在所有器件上都实现。请参见具体器件或器件系列的数据手册的中断向量表，确定某一具体器件实现了哪些中断向量。

8.5 中断设置过程

8.5.1 初始化

以下步骤描述如何配置中断源：

1. 如果不需要嵌套中断，则将 NSTDIS 控制位（INTCON1<15>）置 1。
2. 通过写相应 IPCn 控制寄存器中的控制位来选择每个中断源的用户分配优先级。优先级将取决于具体的应用程序和中断源类型。如果不需要多重优先级，则可以将所有允许的中断源的 IPCn 寄存器控制位设定为同一个非零值。

注： 在器件复位时，器件会初始化 IPCn 寄存器，为所有用户中断源都分配优先级 4。

3. 清零相关的 IFSn 状态寄存器中与外设相关的中断标志状态位。
4. 通过将相应的 IECn 控制寄存器中与中断源相关的中断允许控制位置 1 来允许中断源。

8.5.2 中断服务程序

用于声明 ISR 和使用正确向量地址初始化 IVT 和 AIVT 的方法取决于编程语言（即，C 语言或汇编语言）和用于开发应用程序的语言开发工具套件。一般来说，用户必须将 ISR 处理的中断源在相应 IFSn 寄存器中的中断标志清零。否则，在从 ISR 退出后将立即重新进入 ISR。如果 ISR 使用汇编语言编写，则它必须使用 RETFIE 指令来出栈所保存的 PC 值、SRL 值和原先的 CPU 优先级。

8.5.3 陷阱服务程序

除了必须清零 INTCON1 寄存器中相应的陷阱状态标志来避免重新进入陷阱服务程序（TSR）之外，TSR 使用与 ISR 类似的方式编写。

8.5.4 中断禁止

所有的用户中断均可用以下过程禁止：

1. 使用 PUSH 指令将当前的 SR 值压入软件堆栈。
2. 通过将值 0xE0 与 SRL 进行“或”运算而将 CPU 优先级强制设置为 7。

如果要允许用户中断，则可以使用 POP 指令来恢复先前的 SR 值。

请注意，只能禁止优先级小于等于 7 的用户中断。陷阱源（优先级 8- 优先级 15）无法禁止。

DISI 指令提供了一种简便方法来禁止优先级为 1-6 的中断一个固定的时间周期。DISI 指令不会禁止优先级为 7 的中断源。

8.6 寄存器映射

表 8-2 中提供了与中断控制器相关的特殊功能寄存器汇总。

表 8-2: 与中断控制器相关的特殊功能寄存器

SFR 名称	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SR	—	—	—	—	—	—	—	DC	IPL2	IPL1	IPL0	RA	N	OV	Z	C
CORCON	—	—	—	—	—	—	—	—	—	—	—	—	IPL3	PSV	—	—
INTCON1	NSTDIS	—	—	—	—	—	—	—	—	—	—	MATHERR	ADDRERR	STKERR	OSCFail	—
INTCON2	ALTIVT	DISI	—	—	—	—	—	—	—	—	—	INT4EP	INT3EP	INT2EP	INT1EP	INT0EP
IFS0	V15IF	V14IF	V13IF	V12IF	V11IF	V10IF	V09IF	V08IF	V07IF	V06IF	V05IF	V04IF	V03IF	V02IF	V01IF	V00IF
IFS1	V31IF	V30IF	V29IF	V28IF	V27IF	V26IF	V25IF	V24IF	V23IF	V22IF	V21IF	V20IF	V19IF	V18IF	V17IF	V16IF
IFS2	V47IF	V46IF	V45IF	V44IF	V43IF	V41IF	V41IF	V40IF	V39IF	V38IF	V37IF	V36IF	V35IF	V34IF	V33IF	V32IF
IFS3	V63IF	V62IF	V61IF	V60IF	V59IF	V58IF	V57IF	V56IF	V55IF	V54IF	V53IF	V52IF	V51IF	V50IF	V49IF	V48IF
IFS4	V79IF	V78IF	V77IF	V76IF	V75IF	V74IF	V73IF	V72IF	V71IF	V70IF	V69IF	V68IF	V67IF	V66IF	V65IF	V64IF
IFS5	V95IF	V94IF	V93IF	V92IF	V91IF	V90IF	V89IF	V88IF	V87IF	V86IF	V85IF	V84IF	V83IF	V82IF	V81IF	V80IF
IFS6	V111IF	V110IF	V109IF	V108IF	V107IF	V106IF	V105IF	V104IF	V103IF	V102IF	V101IF	V100IF	V99IF	V98IF	V97IF	V96IF
IFS7	—	—	—	—	—	—	—	—	—	—	V117IF	V116IF	V115IF	V114IF	V113IF	V112IF
IEC0	V15IE	V14IE	V13IE	V12IE	V11IE	V10IE	V09IE	V08IE	V07IE	V06IE	V05IE	V04IE	V03IE	V02IE	V01IE	V00IE
IEC1	V31IE	V30IE	V29IE	V28IE	V27IE	V26IE	V25IE	V24IE	V23IE	V22IE	V21IE	V20IE	V19IE	V18IE	V17IE	V16IE
IEC2	V47IE	V46IE	V45IE	V44IE	V43IE	V41IE	V41IE	V40IE	V39IE	V38IE	V37IE	V36IE	V35IE	V34IE	V33IE	V32IE
IEC3	V63IE	V62IE	V61IE	V60IE	V59IE	V58IE	V57IE	V56IE	V55IE	V54IE	V53IE	V52IE	V51IE	V50IE	V49IE	V48IE
IEC4	V79IE	V78IE	V77IE	V76IE	V75IE	V74IE	V73IE	V72IE	V71IE	V70IE	V69IE	V68IE	V67IE	V66IE	V65IE	V64IE
IEC5	V95IE	V94IE	V93IE	V92IE	V91IE	V90IE	V89IE	V88IE	V87IE	V86IE	V85IE	V84IE	V83IE	V82IE	V81IE	V80IE
IEC6	V111IE	V110IE	V109IE	V108IE	V107IE	V106IE	V105IE	V104IE	V103IE	V102IE	V101IE	V100IE	V99IE	V98IE	V97IE	V96IE
IEC7	—	—	—	—	—	—	—	—	—	—	V117IE	V116IE	V115IE	V114IE	V113IE	V112IE
IPC0	—	V03IP2	V03IP1	V03IP0	—	V02IP2	V02IP1	V02IP0	—	V01IP2	V01IP1	V01IP0	—	V00IP2	V00IP1	V00IP0
IPC1	—	V07IP2	V07IP1	V07IP0	—	V06IP2	V06IP1	V06IP0	—	V05IP2	V05IP1	V05IP0	—	V04IP2	V04IP1	V04IP0
IPC2	—	V11IP2	V11IP1	V11IP0	—	V10IP2	V10IP1	V10IP0	—	V09IP2	V09IP1	V09IP0	—	V08IP2	V08IP1	V08IP0
IPC3	—	V15IP2	V15IP1	V15IP0	—	V14IP2	V14IP1	V14IP0	—	V13IP2	V13IP1	V13IP0	—	V12IP2	V12IP1	V12IP0
IPC4	—	V19IP2	V19IP1	V19IP0	—	V18IP2	V18IP1	V18IP0	—	V17IP2	V17IP1	V17IP0	—	V16IP2	V16IP1	V16IP0
IPC5	—	V23IP2	V23IP1	V23IP0	—	V22IP2	V22IP1	V22IP0	—	V21IP2	V21IP1	V21IP0	—	V20IP2	V20IP1	V20IP0
IPC6	—	V27IP2	V27IP1	V27IP0	—	V26IP2	V26IP1	V26IP0	—	V25IP2	V25IP1	V25IP0	—	V24IP2	V24IP1	V24IP0
IPC7	—	V31IP2	V31IP1	V31IP0	—	V30IP2	V30IP1	V30IP0	—	V29IP2	V29IP1	V29IP0	—	V28IP2	V28IP1	V28IP0
IPC8	—	V35IP2	V35IP1	V35IP0	—	V34IP2	V34IP1	V34IP0	—	V33IP2	V33IP1	V33IP0	—	V32IP2	V32IP1	V32IP0
IPC9	—	V39IP2	V39IP1	V39IP0	—	V38IP2	V38IP1	V38IP0	—	V37IP2	V37IP1	V37IP0	—	V36IP2	V36IP1	V36IP0
IPC10	—	V43IP2	V43IP1	V43IP0	—	V42IP2	V42IP1	V42IP0	—	V41IP2	V41IP1	V41IP0	—	V40IP2	V40IP1	V40IP0
IPC11	—	V47IP2	V47IP1	V47IP0	—	V46IP2	V46IP1	V46IP0	—	V45IP2	V45IP1	V45IP0	—	V44IP2	V44IP1	V44IP0

注： 某个特定器件上不一定具有所有的中断源及其相关的控制位。请参见器件数据手册了解详细信息。

表 8-2: 与中断控制器相关的特殊功能寄存器（续）

SFR 名称	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IPC12	—	V51IP2	V51IP1	V51IP0	—	V50IP2	V50IP1	V50IP0	—	V49IP2	V49IP1	V49IP0	—	V48IP2	V48IP1	V48IP0
IPC13	—	V55IP2	V55IP1	V55IP0	—	V54IP2	V54IP1	V54IP0	—	V53IP2	V53IP1	V53IP0	—	V52IP2	V52IP1	V52IP0
IPC14	—	V59IP2	V59IP1	V59IP0	—	V58IP2	V58IP1	V58IP0	—	V57IP2	V57IP1	V57IP0	—	V56IP2	V56IP1	V56IP0
IPC15	—	V63IP2	V63IP1	V63IP0	—	V62IP2	V62IP1	V62IP0	—	V61IP2	V61IP1	V61IP0	—	V60IP2	V60IP1	V60IP0
IPC16	—	V67IP2	V67IP1	V67IP0	—	V66IP2	V66IP1	V66IP0	—	V65IP2	V65IP1	V65IP0	—	V64IP2	V64IP1	V64IP0
IPC17	—	V71IP2	V71IP1	V71IP0	—	V70IP2	V70IP1	V70IP0	—	V69IP2	V69IP1	V69IP0	—	V68IP2	V68IP1	V68IP0
IPC18	—	V75IP2	V75IP1	V75IP0	—	V74IP2	V74IP1	V74IP0	—	V73IP2	V73IP1	V73IP0	—	V72IP2	V72IP1	V72IP0
IPC19	—	V79IP2	V79IP1	V79IP0	—	V78IP2	V78IP1	V78IP0	—	V77IP2	V77IP1	V77IP0	—	V76IP2	V76IP1	V76IP0
IPC20	—	V83IP2	V83IP1	V83IP0	—	V82IP2	V82IP1	V82IP0	—	V81IP2	V81IP1	V81IP0	—	V80IP2	V80IP1	V80IP0
IPC21	—	V87IP2	V87IP1	V87IP0	—	V86IP2	V86IP1	V86IP0	—	V85IP2	V85IP1	V85IP0	—	V84IP2	V84IP1	V84IP0
IPC22	—	V91IP2	V91IP1	V91IP0	—	V90IP2	V90IP1	V90IP0	—	V89IP2	V89IP1	V89IP0	—	V88IP2	V88IP1	V88IP0
IPC23	—	V95IP2	V95IP1	V95IP0	—	V94IP2	V94IP1	V94IP0	—	V93IP2	V93IP1	V93IP0	—	V92IP2	V92IP1	V92IP0
IPC24	—	V99IP2	V99IP1	V99IP0	—	V98IP2	V98IP1	V98IP0	—	V97IP2	V97IP1	V97IP0	—	V96IP2	V96IP1	V96IP0
IPC25	—	V103IP2	V103IP1	V103IP0	—	V102IP2	V102IP1	V102IP0	—	V101IP2	V101IP1	V101IP0	—	V100IP2	V100IP1	V100IP0
IPC26	—	V107IP2	V107IP1	V107IP0	—	V106IP2	V106IP1	V106IP0	—	V105IP2	V105IP1	V105IP0	—	V104IP2	V104IP1	V104IP0
IPC27	—	V111IP2	V111IP1	V111IP0	—	V110IP2	V110IP1	V110IP0	—	V109IP2	V109IP1	V109IP0	—	V108IP2	V108IP1	V108IP0
IPC28	—	V115IP2	V115IP1	V115IP0	—	V114IP2	V114IP1	V114IP0	—	V113IP2	V113IP1	V113IP0	—	V112IP2	V112IP1	V112IP0
IPC29	—	—	—	—	—	—	—	—	—	V117IP2	V117IP1	V117IP0	—	V116IP2	V116IP1	V116IP0

注： 某个特定器件上不一定具有所有的中断源及其相关的控制位。请参见器件数据手册了解详细信息。

8.7 设计技巧

问 1: 当两个中断源同时等待处理并具有相同的用户分配优先级时，会发生什么？

答: 自然顺序优先级最高的中断源具有优先权。中断源的中断向量表（IVT）地址决定自然顺序优先级。IVT 地址越小的中断源自然顺序优先级越高。

问 2: 是否可以使用 DISI 指令禁止所有中断源和陷阱源？

答: DISI 指令不能禁止陷阱或优先级为 7 的中断源。但是，如果用户应用中没有允许优先级为 7 的中断源，可以使用 DISI 指令作为一种禁止所有中断源的简便方法。

8.8 相关应用笔记

本节列出了与手册本章内容相关的应用笔记。这些应用笔记可能并不是专为 PIC24F 器件系列而编写的，但其概念是相关的，通过适当修改即可使用，但在使用中可能会受到一定限制。当前与中断相关的应用笔记有：

标题	应用笔记编号
目前没有相关的应用笔记。	

注：如需获取更多 PIC24F 系列器件的应用笔记和代码示例，请访问 Microchip 网站 (www.microchip.com)。

8.9 版本历史

版本 A（2006 年 4 月）

这是本文档的初始发行版。

注: