第 28 章 中断 （第 2 部分）

主要内容

手册的这一章包含下面这些主要内容：

28.1 简介 28-2
28.2 不可屏蔽的陷阱 28-6
28.3 中断处理时序 28-11
28.4 中断控制和状态寄存器 28-14
28.5 中断设置步骤 28-40
28.6 设计技巧 28-44
28.7 相关应用笔记 28-45
28.8 版本历史 28-46

28.1 简介

dsPIC® 开关电源（Switched Mode Power Supply，SMPS）DSC 系列的中断控制器将大量的外设中断请求信号简化为送到 CPU 的单一中断请求信号，它具有以下特性：

- 多达 8 个处理器异常和软件陷阱
- 8 个可由用户选择的优先级
- 具有多达 62 个向量的中断向量表 （Interrupt Vector Table， IVT）
- 每个中断源或异常源对应一个唯一的向量
- 在一个指定的用户优先级内具有固定的优先级
- 用于支持调试功能的备用中断向量表 （Alternate Interrupt Vector Table， AIVT）
- 固定的中断进入延时和中断返回延时

28.1.1 中断向量表

中断向量表（IVT）如图 28-1 所示。IVT 位于程序存储器中，起始单元地址是 0x000004。IVT 包含 62 个向量，由 8 个不可屏蔽陷阱向量和多达 54 个中断源组成。一般来说，每个中断源都有自己的中断向量。每个中断向量都包含一个 24 位的地址。每个中断向量单元中设定的值是其相应的中断服务程序 （Interrupt Service Routine， ISR）的起始地址。

28.1.2 备用向量表

备用中断向量表（Alternate Interrupt Vector Table， AIVT）位于 IVT 之后。由中断控制寄存器 2 中的备用中断向量表使能控制位 ALTIVT （INTCON2<15>）控制对 AIVT 的访问。如果 ALTIVT 位置 1，则所有的中断和异常处理都使用备用向量，而非默认向量。备用向量与默认向量的组织方式相同。

AIVT提供了一种无需重新编程中断向量就可在应用程序和支持环境之间切换的方法，来支持仿真和调试功能。此特性也支持运行时在应用程序之间切换以便评估不同的软件算法。如果不需要 AIVT，则需要用 IVT 中使用的相同地址编程 AIVT。

28.1.3 复位过程

器件复位不是真正的异常，因为复位过程中并不涉及到中断控制器。作为对复位的响应， dsPIC SMPS DSC 器件清零其寄存器，同时强制 PC 为零。然后处理器从地址 0x000000 处开始执行程序。用户应用程序可以在复位地址中写入 GOTO 指令，将程序执行重定向到相应的启动程序。

注：	应该使用包含 RESET 指令的默认中断处理程序的入口地址编程 IVT 和 AIVT 中所有未实现或未使用的向量单元。

© 2008 Microchip Technology Inc.

图 28-1：　中断向量表

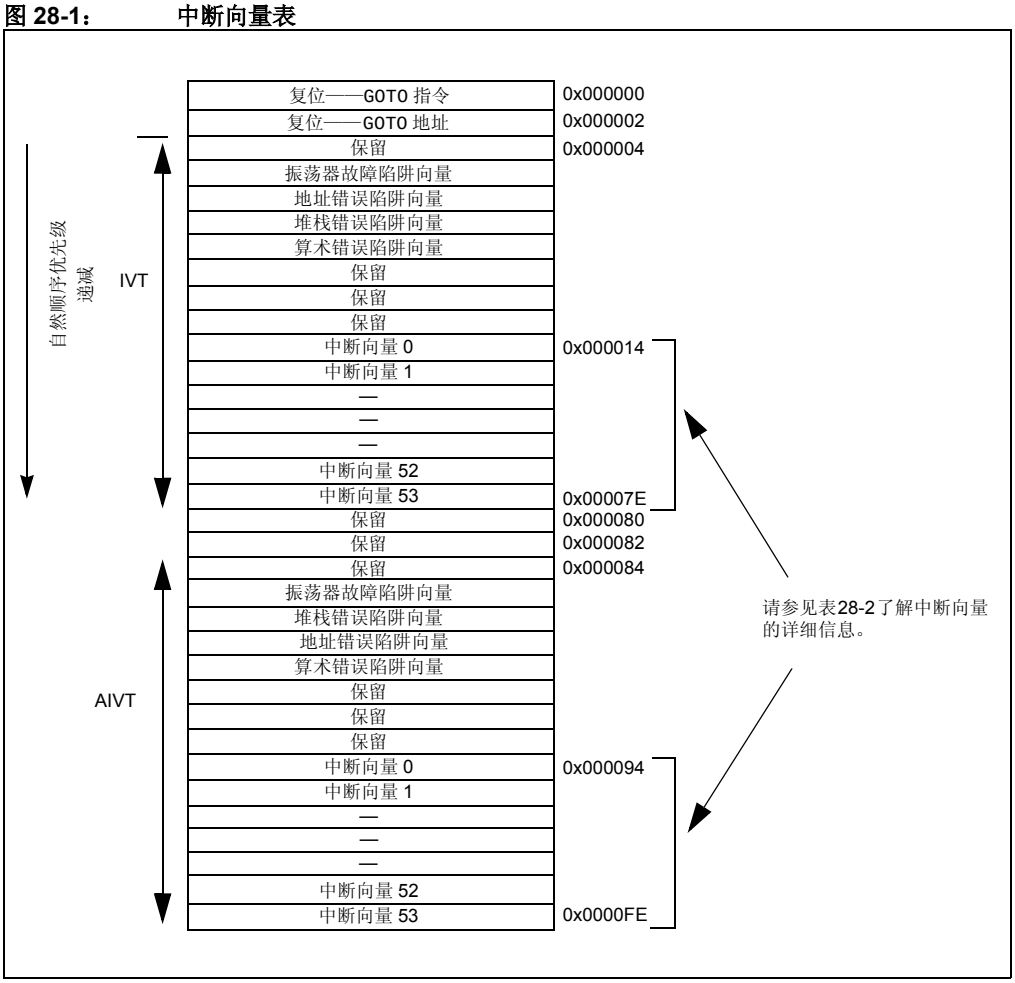


表 28-1：　陷阱向量详细信息

向量编号	IVT 地址	AIVT 地址	陷阱源
0	0x000004	0x000084	保留
1	0x000006	0x000086	振荡器故障
2	0x000008	0x000088	地址错误
3	0x00000A	0x00008A	堆栈错误
4	0x00000C	0x00008C	算术错误
5	0x00000E	0x00008E	保留
6	0x000010	0x000090	保留
7	0x000012	0x000092	保留

表 28-2： 中断向量详细信息

向量编号	IVT 地址	AIVT 地址	中断源
8	0x000014	0x000094	INT0——外部中断 0
9	0x000016	0x000096	IC1——输入捕捉 1
10	0x000018	0x000098	OC1——输出比较 1
11	0x00001A	0x00009A	T1——Timer1
12	0x00001C	0x00009C	保留
13	0x00001E	0x00009E	OC2——输出比较 2
14	0x000020	0x0000A0	T2——Timer2
15	0x000022	0x0000A2	T3——Timer3
16	0x000024	0x0000A4	SPI1
17	0x000026	0x0000A6	U1RX——UART1 接收器
18	0x000028	0x0000A8	U1TX——UART1 发送器
19	0x00002A	0x0000AA	ADC——ADC 转换完成
20	0x00002C	0x0000AC	NVM——NVM 写完成
21	0x00002E	0x0000AE	SI2C——I^2C™ 从事件
22	0x000030	0x0000B0	MI2C——I^2C 主事件
23	0x000032	0x0000B2	保留
24	0x000034	0x0000B4	INT1——外部中断 1
25	0x000036	0x0000B6	INT2——外部中断 2
26	0x000038	0x0000B8	PWM 特殊事件触发
27	0x00003A	0x0000BA	PWM 发生器 1
28	0x00003C	0x0000BC	PWM 发生器 2
29	0x00003E	0x0000BE	PWM 发生器 3
30	0x000040	0x0000C0	PWM 发生器 4
31	0x000042	0x0000C2	保留
32	0x000044	0x0000C4	保留
33	0x000046	0x0000C6	保留
34	0x000048	0x0000C8	保留
35	0x00004A	0x0000CA	CN——输入电平变化通知
36	0x00004C	0x0000CC	保留
37	0x00004E	0x0000CE	模拟比较器 1
38	0x000050	0x0000D0	模拟比较器 2
39	0x000052	0x0000D2	模拟比较器 3
40	0x000054	0x0000D4	模拟比较器 4
41	0x000056	0x0000D6	保留
42	0x000058	0x0000D8	保留
43	0x00005A	0x0000DA	保留
44	0x00005C	0x0000DC	保留
45	0x00005E	0x0000DE	ADC 输入对 0 转换完成
46	0x000060	0x0000E0	ADC 输入对 1 转换完成
47	0x000062	0x0000E2	ADC 输入对 2 转换完成
48	0x000064	0x0000E4	ADC 输入对 3 转换完成
49	0x000066	0x0000E6	ADC 输入对 4 转换完成
50	0x000068	0x0000E8	ADC 输入对 5 转换完成
51	0x00006A	0x0000EA	保留
52	0x00006C	0x0000EC	保留
53-61	0x00006E-0x00007E	0x0000EE-0x0000FE	保留
最低自然顺序优先级			

© 2008 Microchip Technology Inc.

28.1.4 CPU 优先级状态

CPU 可以工作在 16 个优先级 （0-15）之一。为了开始异常处理，中断源或者陷阱源的优先级必须高于当前的 CPU 优先级。外设中断源和外部中断源可以编程为优先级 0-7，而 CPU 优先级 8-15 保留给陷阱源。陷阱是不可屏蔽的中断源，用于检测硬件和软件问题 （参见**第 28.2 节 “不可屏蔽的陷阱”**）。每个陷阱源的优先级是固定的。一个优先级只可分配给一个陷阱。编程为优先级 0 的中断源实际上是被禁止的，因为它的优先级永远不会大于 CPU 的优先级。

以下状态位用于指示当前的 CPU 优先级：

- 状态寄存器中的 CPU 中断优先级状态位 IPL<2:0> （SR<7:5>）
- 内核控制寄存器中的 CPU 中断优先级状态位 IPL3 （CORCON<3>）

IPL<2:0> 状态位是可读写的，因此用户应用程序可以修改这些位以禁止所有优先级低于给定优先级的中断源。例如，如果 IPL<2:0> = 3， CPU 就不会被任何优先级编程为 0、 1、 2 或 3 的中断源中断。

陷阱事件的优先级比任何用户中断源都高。当 IPL3 位被置 1 时表示正在处理陷阱事件。用户应用程序可以清零 IPL3 位，但不能将其置 1。在某些应用中，可能会倾向于在发生陷阱时将 IPL3 位清零，并跳转到另外一条指令，而不是原先导致陷阱发生的那条指令的下一条指令。

可通过设置 IPL<2:0> = 111 禁止所有用户中断源。

注：	当禁止中断嵌套时，IPL<2:0> 位变成只读位。更多信息，请参见**第 28.2.4.2 节 “中断嵌套”**。

28.1.5 中断优先级

可以为每个外设中断源分配 7 个优先级之一。每个中断的用户可分配中断优先级控制位位于中断优先级控制 （IPCx）寄存器中每半字节的低 3 位中。每个半字节的 bit 3 不使用且读为 0。这些位定义了分配给特定中断的优先级。可用的优先级为从 1 （最低优先级）到 7 （最高优先级）。如果与中断源相关的 IPC 位全部被清零，则中断源被有效禁止。

注：	如果应用程序要实时地重新配置中断优先级，则在执行此操作时必须禁止中断。未能禁止中断可能会产生不可预料的结果。

由于一个特定的优先级可被分配给多个中断请求源，因此提供了一种解决给定用户分配优先级内优先级冲突的方法。每个中断源都有一个自然顺序优先级，这由其在 IVT 中的位置决定。表 28-2 给出了每个中断源在 IVT 中的位置。中断向量的编号越低，其自然优先级越高，而向量的编号越高，其自然优先级越低。任何待处理的中断源的总优先级都首先由该中断源在中断优先级控制（IPCx）寄存器中的用户分配优先级决定，然后由 IVT 中的自然顺序优先级决定。

自然顺序优先级仅用于解决具有相同用户分配优先级而同时等待处理的中断之间的冲突。一旦解决了优先级冲突，异常处理过程就开始了，CPU 只能被具有更高用户分配优先级的中断源中断。在异常处理过程开始后才等待处理的中断，如果它与当前正在处理的中断具有相同的用户分配优先级，即便具有较高的自然顺序优先级，也将保持待处理状态直到当前的异常处理过程结束。

为每个中断源分配七个优先级之一使用户应用程序可以给低自然顺序优先级的中断分配非常高的总优先级。例如，可以给 Timer2 分配优先级 7，并给外部中断 （INT0）分配优先级 1，这样就给了它一个很低的有效优先级。

注： IVT 中的外设中断源和外部中断源随特定 dsPIC SMPS DSC 器件而有所不同。此文档中所列的中断源为 dsPIC SMPS DSC 器件的所有中断源的综合列表。更多详细信息请参见特定的器件数据手册。

28.2 不可屏蔽的陷阱

陷阱是不可屏蔽的可嵌套中断，遵循固定的优先级结构。陷阱提供了一种方法来纠正在调试和运行应用程序时的错误操作。如果用户应用程序不打算纠正陷阱错误条件，那么必须在这些陷阱向量中装入将使器件复位的软件程序地址。否则，用户应用程序将陷阱向量编程为纠正陷阱条件的服务程序的地址。

dsPIC SMPS DSC 器件实现了四个不可屏蔽陷阱源：

- 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

对于许多陷阱条件，允许引起陷阱的指令在异常处理开始前执行完毕。因此，用户应用程序可能不得不纠正导致陷阱的指令的操作。

每个陷阱源具有固定的优先级，由其在 IVT 中的位置定义。振荡器故障陷阱具有最高优先级。此外，陷阱源被分为两种不同的类别：硬陷阱和软陷阱。

28.2.1 软陷阱

算术错误陷阱 （优先级为 11）和堆栈错误陷阱 （优先级为 12）归类为软陷阱源。软陷阱可视为不可屏蔽的中断源，采用由其在 IVT 中位置指定的优先级。软陷阱的处理过程和中断类似，在异常处理之前需要两个周期对其进行采样和响应。因此，在软陷阱被响应之前可以执行一些额外的指令。

28.2.1.1 堆栈错误陷阱 （软陷阱，优先级 12）

复位时堆栈被初始化为 0x0800。如果堆栈指针地址小于 0x0800，就会产生堆栈错误陷阱。

与堆栈指针相关的堆栈限制寄存器 （SPLIM）在复位时不被初始化。在将一个字写入 SPLIM 之前，不使能堆栈溢出检测。

使用 W15 作为源指针或目标指针生成的所有有效地址（EA）与 SPLIM 寄存器中的值进行比较。如果 EA 大于 SPLIM 寄存器中的内容，将产生堆栈错误陷阱。此外，如果 EA 计算值超过了数据空间的结束地址 （0xFFFF），也会产生堆栈错误陷阱。

可通过在软件中查询中断控制寄存器 1 中的堆栈错误陷阱状态位 STKERR （INTCON1<2>）检测堆栈错误。要避免再次进入陷阱服务程序，必须在退出堆栈错误陷阱之前用软件清零 STKERR 状态标志位。

© 2008 Microchip Technology Inc.

28.2.1.2 算术错误陷阱 （软陷阱，优先级 11）

以下任一事件都会产生算术错误陷阱：

- 累加器 A 溢出
- 累加器 B 溢出
- 灾难性累加器溢出
- 被 0 除
- 移位累加器 （SFTAC）操作超过 ±16 位

INTCON1 寄存器中有三位可允许三种类型的累加器溢出陷阱：

- 累加器 A 溢出陷阱允许控制位 OVATE （INTCON1<10>）用于允许累加器 A 溢出事件的陷阱。
- 累加器 B 溢出陷阱允许控制位 OVBTE （INTCON1<9>）用于允许累加器 B 溢出事件的陷阱。
- 灾难性溢出陷阱允许控制位 COVTE （INTCON1<8>）用于允许任一累加器的灾难性溢出陷阱。当检测到此陷阱时， INTCON1 寄存器中的相应错误标志位就会被置 1：
  - 累加器 A 溢出陷阱标志 （OVAERR）
  - 累加器 B 溢出陷阱标志 （OVBERR）
  - 累加器 A 灾难性溢出陷阱标志 （COVAERR）
  - 累加器 B 灾难性溢出陷阱标志 （COVBERR）

累加器 A 或累加器 B 溢出事件定义为从 bit 31 进位。注意如果使能了累加器的 31 位饱和模式就不会发生累加器溢出。灾难性累加器溢出定义为从任一累加器的 bit 39 进位。如果使能了累加器饱和 （31 位或 39 位）就不会发生灾难性溢出。

不能禁止被 0 除陷阱。在执行除法指令的 REPEAT 循环的第一次迭代中执行被 0 除检测。当检测到此陷阱时，算术错误状态位 DIV0ERR （INTCON1<6>）被置 1。

不能禁止累加器移位陷阱。 SFTAC 指令可用于将累加器移位一个立即数值或某个 W 寄存器中的值。如果移位值超过 ±16 位，将产生算术错误陷阱，并且累加器移位错误状态位 SFTAERR （INTCON1<7>）被置 1。 SFTAC 指令仍会执行，但移位的结果不会写入目标累加器。

可通过查询算术错误状态位 MATHERR（INTCON1<4>）用软件检测算术错误陷阱。要避免再次进入陷阱服务程序，必须在退出算术错误陷阱之前用软件清零 MATHERR 状态标志位。在将 MATHERR 状态位清零之前，所有导致产生陷阱的条件也必须被清除。如果陷阱是由于累加器溢出而产生的， OA 和 OB 状态位 （SR<15:14>）必须清零。 OA 和 OB 状态位是只读的，因此用户软件必须对溢出的累加器执行一个空操作（比如加 0），从而使得硬件清零 OA 或 OB 状态位。

28.2.2 硬陷阱

硬陷阱包括优先级为 13 到 15 的异常 （含优先级 13 和优先级 15）。地址错误 （优先级 13）和振荡器错误 （优先级 14）陷阱都属于此类。

和软陷阱一样，硬陷阱也可以看作不可屏蔽的中断源。硬陷阱和软陷阱之间的区别在于，引起陷阱的指令执行完之后，硬陷阱会强制 CPU 停止代码执行。在陷阱被响应和处理之前，正常程序执行流不会恢复。

28.2.2.1 陷阱优先级和硬陷阱冲突

如果在处理任何一个优先级较低的陷阱时发生优先级较高的陷阱，低优先级陷阱的处理会暂停，高优先级陷阱将被响应并处理。在高优先级陷阱处理完成之前，低优先级陷阱将保持在等待处理状态。

在能够继续进行任何类型的代码执行之前，每个发生的硬陷阱均必须先被响应。如果在优先级较高的陷阱处于待处理、响应或正在处理的过程中发生了较低优先级的硬陷阱，就会产生硬陷阱冲突。产生硬陷阱冲突的原因是，在较高优先级的陷阱处理完成之前，不能响应较低优先级的陷阱。

在硬陷阱冲突条件下，器件会自动复位。在发生复位时，复位模块的复位控制寄存器中的陷阱复位标志状态位 TRAPR （RCON<15>）被置 1，这样就可以用软件检测到冲突条件。

28.2.2.2 振荡器故障陷阱 （硬陷阱，优先级 14）

以下任何一个原因都将产生振荡器故障陷阱事件：

- 故障保护时钟监视器 （Fail-Safe Clock Monitor， FSCM）被使能并检测到系统时钟源丢失。
- 在使用 PLL 的正常工作期间检测到 PLL 失锁。
- FSCM 被使能且 PLL 在上电复位 （POR）时锁定失败。

通过查询振荡器模块中的振荡器故障陷阱状态位 OSCFAIL （INTCON1<1>）或时钟失效状态位 CF（OSCCON<3>），可用软件检测振荡器故障陷阱事件。要避免再次进入陷阱服务程序，必须在退出振荡器故障陷阱之前用软件清零 OSCFAIL 状态标志位。

28.2.2.3 地址错误陷阱 （硬陷阱，优先级 13）

以下工作情形会产生地址错误陷阱：

- 试图取不对齐的数据字。当一条指令执行了一个有效地址的 LSb 被置为 1 的字访问时会发生这种情况。 dsPIC SMPS DSC 器件要求所有字访问与偶地址边界对齐。
- 一条位操作指令使用有效地址的 LSb 被置为 1 的间接寻址模式。
- 试图从未实现的数据地址空间取数据。
- 执行 BRA #literal 指令或 GOTO #literal 指令，其中 literal 是未实现的程序存储器地址。
- 修改程序计数器 （PC）使其指向未实现的程序存储器地址后，执行指令。通过将值装入堆栈并执行 RETURN 指令可修改程序计数器。

发生地址错误陷阱时，数据空间写操作被禁止，这样数据就不会被破坏。

可以通过查询地址错误陷阱状态位 ADDRERR （INTCON1<3>）用软件检测地址错误。要避免再次进入陷阱服务程序，必须在退出地址错误陷阱之前用软件清零 ADDRERR 状态标志位。

注： 在 MAC 类指令中，数据空间被分成 X 空间和 Y 空间。在这些指令中，未实现的 X 空间包括所有 Y 空间，并且未实现的 Y 空间包括所有 X 空间。

© 2008 Microchip Technology Inc.

28.2.3 禁止中断指令

DISI （禁止中断）指令能够将中断禁止长达 16384 个指令周期。当执行时间要求严格的代码段时，此指令很有用处。

DISI 指令只禁止优先级为 1-6 的中断。当 DISI 指令有效时，优先级为 7 的中断和所有陷阱事件仍然可以中断 CPU。

DISI 指令与 CPU 中的禁止中断计数（DISICNT）寄存器配合工作。当 DISICNT 寄存器非零时，优先级为 1-6 的中断被禁止。 DISICNT 寄存器在每个后续的指令周期中递减。当 DISICNT 寄存器计数到零时，优先级为 1-6 的中断被重新允许。DISI 指令中指定的值包括由于 PSV 访问和指令停顿等所花费的所有周期。

DISICNT 寄存器是可读写的。用户应用程序可通过清零 DISICNT 寄存器提前终止前一条 DISI 指令的影响。可以通过写入或向 DISICNT 寄存器加一个值来延长中断被禁止的时间。

如果 DISICNT 寄存器为零，简单地写一个非零值到寄存器不能禁止中断。必须首先使用 DISI 指令禁止中断。一旦DISI指令被执行，且DISICNT的值为非零值，应用程序即可通过修改DISICNT寄存器的内容来延长中断禁止时间。

注： 不推荐通过软件修改 DISICNT 寄存器。

每当由于执行 DISI 指令使中断被禁止， DISI 指令状态位 DISI （INTCON2<14>）就被置 1。

注： 如果未分配 CPU 优先级为 7 的中断源，则可以用 DISI 指令快速禁止所有用户中断源。

28.2.4 中断操作

在每个指令周期中都对所有中断事件标志进行采样。中断标志状态（IFSx）寄存器中的某个标志位等于 1 表示有等待处理的中断请求 （Interrupt Request， IRQ）。如果中断允许 （IECx）寄存器中相应的位被置 1，IRQ 将会导致产生中断。在对 IRQ 采样的指令周期的剩余时间中，将评估所有待处理的中断请求的优先级。

当 CPU 响应 IRQ 时，指令不会被中止。采样 IRQ 时正在执行的指令将会继续执行完毕，此后才会执行 ISR。

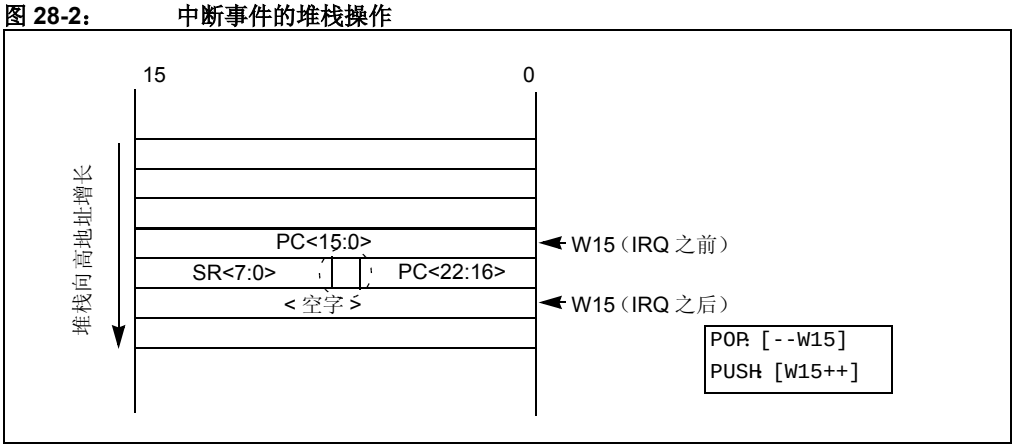
当待处理 IRQ 的用户分配优先级大于当前的处理器优先级 （由 IPL<2:0> 状态位 （SR<7:5>）指示）时，会将中断请求送到处理器。然后处理器将以下信息保存到软件堆栈中：

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

这三个值允许自动保存返回 PC 地址值、 MCU 状态位和当前的处理器优先级。

在以上信息被保存到堆栈之后， CPU 将待处理中断的优先级写入 IPL<2:0> 位。此操作将禁止所有优先级小于或等于此优先级的中断，直到使用 RETFIE （从中断返回）指令终止 ISR。

图 28-2：　　中断事件的堆栈操作



28.2.4.1　从中断返回

RETFIE（从中断返回）指令将使 PC 返回地址、IPL3 状态位和 SRL 寄存器弹出堆栈以使处理器返回到中断处理过程之前的状态和优先级。

28.2.4.2　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。可以通过置 1 中断嵌套禁止控制位 NSTDIS （INTCON1<15>）禁止中断嵌套。当 NSTDIS 控制位被置 1 时，所有正在处理的中断将通过设置 IPL<2:0> = 111 强制 CPU 优先级为 7。此操作将有效屏蔽所有其他中断源，直到执行 RETFIE 指令。当中断嵌套被禁止时，用户分配的中断优先级无效，除非是为了解决同时等待处理的中断之间的冲突。

当中断嵌套被禁止时， IPL<2:0> 位 （SR<7:5>）变成只读位。这将防止用户软件将 IPL<2:0> 设置为一个较低的值，防止其有效地重新使能中断嵌套。

28.2.5　从休眠和空闲模式唤醒

使用中断允许控制（IECx）寄存器中相应的控制位单独允许的任何中断源，都可以将处理器从休眠或空闲模式唤醒。当中断源的中断状态标志位被置 1，且通过 IECx 寄存器中相应的位允许此中断源时，唤醒信号被送到 CPU。当器件从休眠或空闲模式唤醒时，会发生以下两种操作之一：

1. 如果该中断源的中断优先级大于当前的 CPU 优先级，处理器将处理该中断并转移到该中断源的 ISR。
2. 如果中断源的用户分配中断优先级小于或等于当前的 CPU 优先级，处理器将继续执行，从先前使 CPU 进入休眠或空闲模式的 PWRSAV 指令后紧跟的那条指令开始执行。

注：	分配为 CPU 优先级 0 的用户中断源不能将 CPU 从休眠或空闲模式唤醒，因为中断源实际上是被禁止的。要使用一个中断作为唤醒源，此中断的 CPU 优先级必须被分配为 CPU 优先级 1 或更高。

28.2.6　A/D 转换器外部转换请求

INT0 外部中断请求引脚与 A/D 转换器的外部转换请求信号引脚复用。INT0 中断源具有可编程的边沿极性，它也可用于 A/D 转换器外部转换请求功能。

© 2008 Microchip Technology Inc.

28.2.7 外部中断支持

dsPIC SMPS DSC 器件支持最多 3 个外部中断引脚源 （INT0-INT2）。每个外部中断引脚都有边沿检测电路来检测中断事件。INTCON2 寄存器有 3 个控制位（INT0EP-INT2EP）用于选择边沿检测电路的极性。每个外部中断引脚都可以被编程为在上升沿或下降沿事件中断 CPU。

28.3 中断处理时序

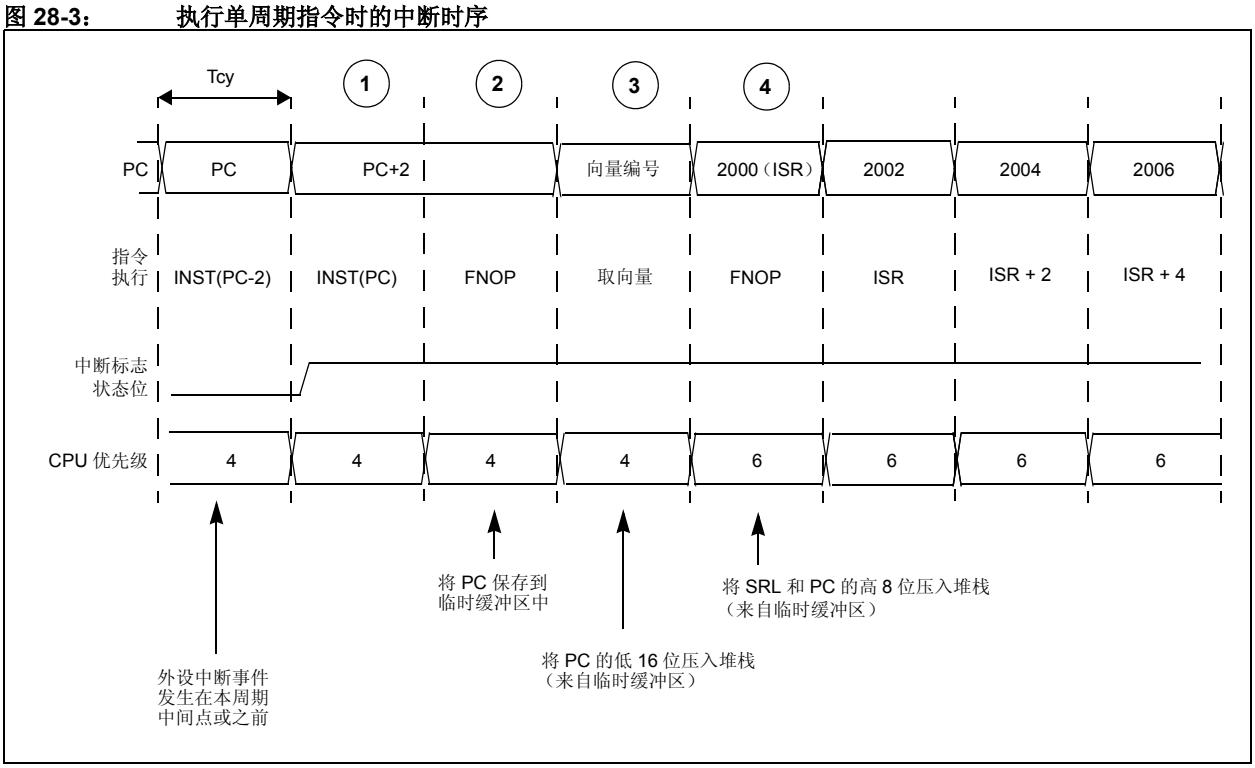
下面各节描述了中断处理时序。

28.3.1 执行单周期指令时的中断延时

图 28-3 所示为在执行单周期指令过程中发生外设中断时的事件时序。中断处理需要四个指令周期。每个周期都在图中编号以供参考。

在外设中断发生后的指令周期中，中断标志状态位被置 1。在此指令周期中，当前指令执行完毕。在中断事件后的第二个指令周期中， PC 和 SRL 寄存器的内容被保存到临时缓冲寄存器。中断处理的第二个周期执行一条 NOP 指令，以保持与双周期指令中时序的一致性（参见**第 28.3.2 节“执行双周期指令时的中断延时”**）。在第三个周期中，PC 被装入中断源的向量表地址并取出 ISR 的起始地址。在第四个周期中， PC 被装入 ISR 的地址。第四个周期执行一条 NOP 指令，同时取出 ISR 中的第一条指令。

图 28-3： 执行单周期指令时的中断时序



28.3.2 执行双周期指令时的中断延时

执行双周期指令时的中断延时和单周期指令相同。中断处理的第一个和第二个周期允许双周期指令执行完毕。图 28-4 中的时序图给出了在执行双周期指令之前的指令周期中发生外设中断事件的情况。

图 28-5 给出了外设中断发生在双周期指令的第一个周期时的时序。在这种情况下，中断处理的完成情况与单周期指令相同（见**第 28.3.1 节 “执行单周期指令时的中断延时”**）。

图 28-4： 执行双周期指令时的中断时序

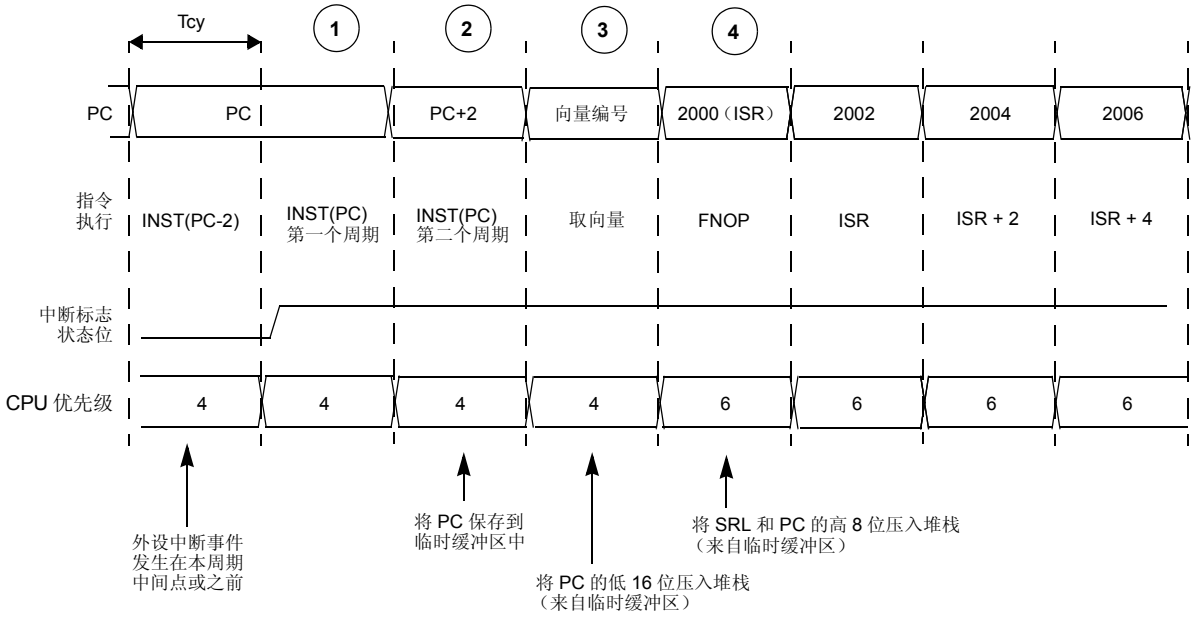
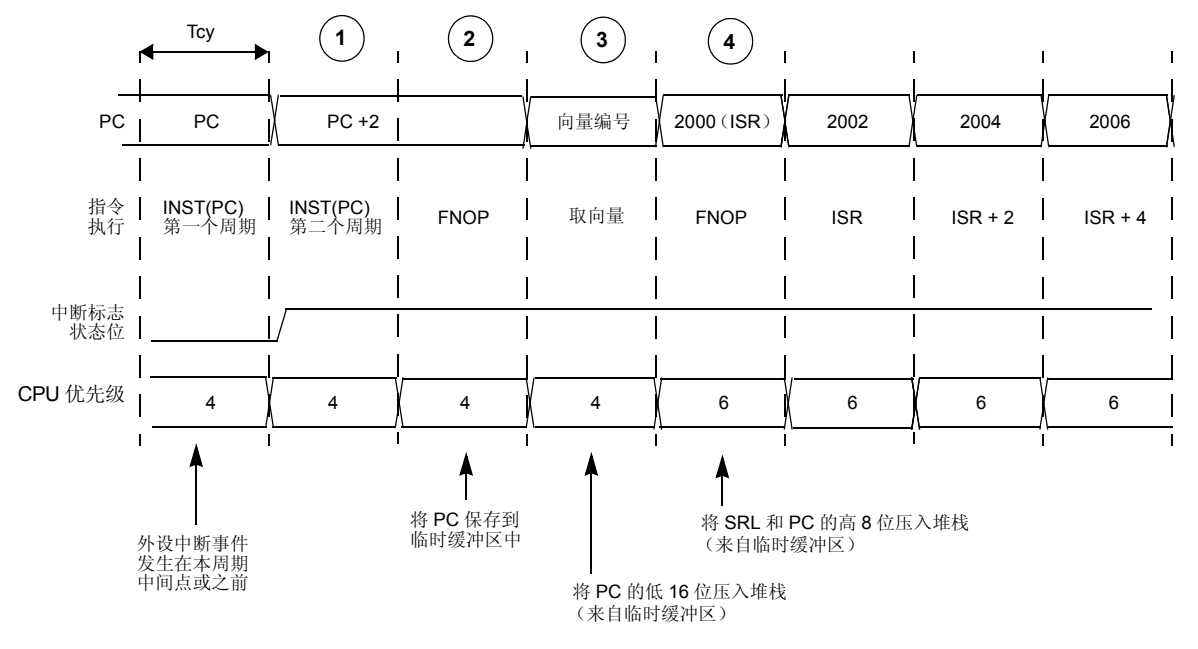


图 28-5： 中断时序，中断在双周期指令的第一个周期发生



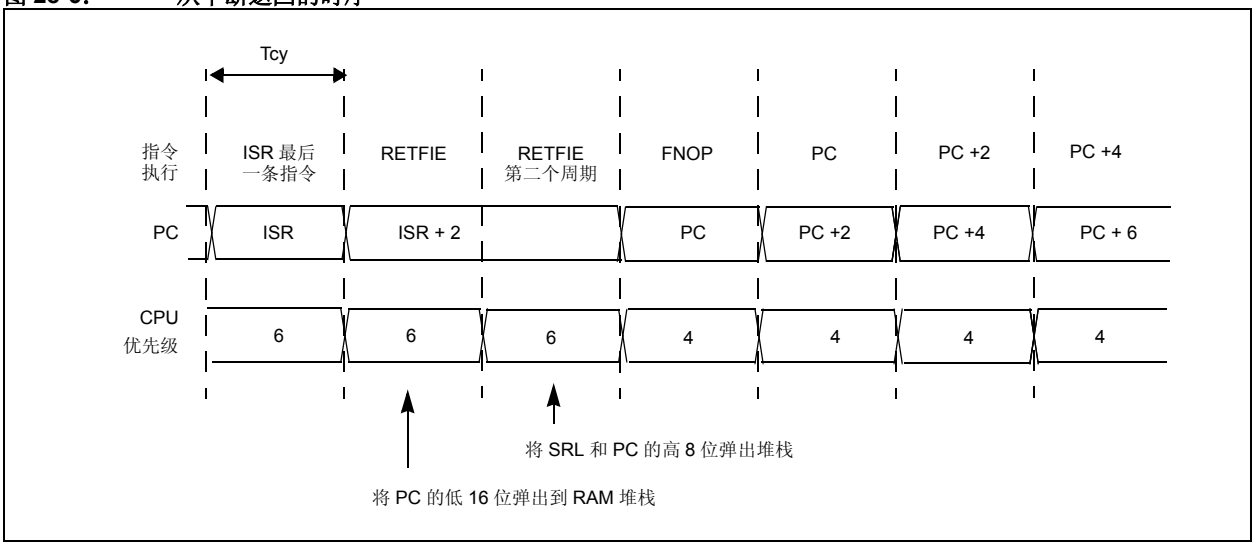
© 2008 Microchip Technology Inc.

28.3.3 从中断返回

要从中断返回，程序必须调用 RETFIE 指令。

在 RETFIE 指令的前两个周期中，PC 和 SRL 寄存器的内容被弹出堆栈。第三个指令周期用于取出由更新的程序计数器寻址的指令。第四个周期执行一条 NOP 指令。程序执行返回到发生中断处继续执行。

图 28-6： 从中断返回的时序



28.3.4 中断延时的特殊情况

当一个外设中断源等待处理时，dsPIC SMPS DSC 器件允许执行完当前指令。对于单周期或双周期指令，中断延时是相同的。但是，根据中断发生的时间，某些情况可能使中断延时增加一个周期。如果固定的中断延时对应用非常关键的话，用户应该避免出现下列这些情况：

- 执行了使用 PSV 访问程序存储空间中值的 MOV.D 指令
- 给任何双周期指令附加一个指令停顿周期
- 给任何执行 PSV 访问的单周期指令附加一个指令停顿周期
- 使用 PSV 访问程序存储空间中值的位测试和跳过指令 （BTSC 和 BTSS）

28.4 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**
  这两个寄存器提高全局中断控制功能。 INTCON1 包含中断嵌套禁止 （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的操作以及备用中断向量表的使用。
- **IFSx：中断标志状态寄存器**
  所有中断请求标志都在 IFSx 寄存器中，其中 “x”表示寄存器编号。每个中断源都有一个状态位，由各自的外设或外部信号置 1，而由软件清零。
- **IECx：中断允许控制寄存器**
  所有中断允许控制位都在 IECx 寄存器中，其中 “x”表示寄存器编号。这些控制位用于单独允许外设中断或外部信号中断。
- **IPCx：中断优先级控制寄存器**
  每个用户中断源都可以分配为 8 个优先级之一。IPC 寄存器用于设置每个中断源的中断优先级。
- **SR：CPU 状态寄存器**
  SR 并不是中断控制器硬件中的特定组成部分，但它包含 IPL<2:0> 状态位 （SR<7:5>），用来指示当前的 CPU 中断优先级。用户可以通过写这些 IPL 位来更改当前 CPU 优先级。
- **CORCON：内核控制寄存器**
  CORCON 也不是中断控制器硬件的特定组成部分，但它包含 IPL3 状态位，用来指示当前的 CPU 中断优先级。 IPL3 是只读位，所以用户软件不能屏蔽陷阱事件。
- **INTTREG：中断控制和状态寄存器**
  INTTREG 寄存器包含相关的中断向量编号和新的 CPU 中断优先级，分别锁存在 INTTREG 寄存器中的向量编号 （VECNUM<6:0>）和中断优先级 （ILR<3:0>）位域中。新的中断优先级是等待处理中断的优先级。

接下来的章节将对每个寄存器进行详细描述。

注：	中断源的总数和类型取决于具体的器件。更多详细信息请参见特定的器件数据手册。

28.4.1 控制寄存器的中断分配

中断源按表 28-2 中的顺序分配给 IFSx、 IECx 和 IPCx 寄存器。例如，所示的 INT0 （外部中断 0）带有向量编号并且其自然顺序优先级为 0。因此 INT0IF 状态位位于 IFS0<0>。INT0 中断使用 IEC0 寄存器的 bit 0 作为其允许位，同时 IPC0<2:0> 位为 INT0 中断分配中断优先级。

© 2008 Microchip Technology Inc.

寄存器 28-1： SR：状态寄存器（在 CPU 中）

高字节：							
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
OA	OB	SA	SB	OAB	SAB	DA	DC
bit 15							bit 8

低字节：							
R/W-0	R/W-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
IPL<2:0>			RA	N	OV	Z	C
bit 7							bit 0

bit 15-8 中断控制器不使用
（有关 SR 寄存器中各位的描述，请参见 《dsPIC30F/33F 程序员参考手册》（DS70157B_CN）。）

bit 7-5 **IPL<2:0>：**CPU 中断优先级状态位
111 = CPU 中断优先级为 7（15）。禁止用户中断。
110 = CPU 中断优先级为 6（14）
101 = CPU 中断优先级为 5（13）
100 = CPU 中断优先级为 4（12）
011 = CPU 中断优先级为 3（11）
010 = CPU 中断优先级为 2（10）
001 = CPU 中断优先级为 1（9）
000 = CPU 中断优先级为 0（8）

注 1：IPL<2:0> 位与 IPL<3> 位（CORCON<3>）组合形成 CPU 中断优先级。如果 IPL<3> = 1，那么括号中的值表示中断优先级。

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

bit 4-0 中断控制器不使用
（有关 SR 寄存器中各位的描述，请参见 《dsPIC30F/33F 程序员参考手册》（DS70157B_CN）。）

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

寄存器 28-2： CORCON：内核控制寄存器

高字节：							
U-0	U-0	U-0	R/W-0	R/W-0	R-0	R-0	R-0
—	—	—	US	EDT	DL<1:0>		
bit 15							bit 8

低字节：							
R/W-0	R/W-0	R/W-1	R/W-0	R/C-0	R/W-0	R/W-0	R/W-0
SATA	SATB	SATDW	ACCSAT	IPL3	PSV	RND	IF
bit 7							bit 0

bit 15-4 中断控制器不使用
（有关 CORCON 寄存器中各位的描述，请参见 《dsPIC30F/33F 程序员参考手册》（DS70157B_CN）。）

bit 3 **IPL3**：CPU 中断优先级状态位 3
1 = CPU 中断优先级大于 7
0 = CPU 中断优先级等于或小于 7

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

bit 2-0 中断控制器不使用
（有关 CORCON 寄存器中各位的描述，请参见 《dsPIC30F/33F 程序员参考手册》（DS70157B_CN）。）

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

© 2008 Microchip Technology Inc.

寄存器 28-3： INTCON1：中断控制寄存器 1

高字节：							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
NSTDIS	OVAERR	OVBERR	COVAERR	COVBERR	OVATE	OVBTE	COVTE
bit 15							bit 8

低字节：							
R/W-0-0	R/W-0-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
SFTACERR	DIV0ERR	—	MATHERR	ADDRERR	STKERR	OSCFAIL	—
bit 7							bit 0

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 **OVAERR**：累加器 A 溢出陷阱标志位
1 = 陷阱是由累加器 A 溢出引起的
0 = 陷阱不是由累加器 A 溢出引起的

bit 13 **OVBERR**：累加器 B 溢出陷阱标志位
1 = 陷阱是由累加器 B 溢出引起的
0 = 陷阱不是由累加器 B 溢出引起的

bit 12 **COVAERR**：累加器 A 灾难性溢出陷阱标志位
1 = 陷阱是由累加器 A 灾难性溢出引起的
0 = 陷阱不是由累加器 A 灾难性溢出引起的

bit 11 **COVBERR**：累加器 B 灾难性溢出陷阱标志位
1 = 陷阱是由累加器 B 灾难性溢出引起的
0 = 陷阱不是由累加器 B 灾难性溢出引起的

bit 10 **OVATE**：累加器 A 溢出陷阱允许位
1 = 允许累加器 A 溢出陷阱
0 = 禁止累加器 A 溢出陷阱

bit 9 **OVBTE**：累加器 B 溢出陷阱允许位
1 = 允许累加器 B 溢出陷阱
0 = 禁止累加器 B 溢出陷阱

bit 8 **COVTE**：灾难性溢出陷阱允许位
1 = 允许累加器 A 或 B 的灾难性溢出陷阱
0 = 禁止累加器 A 或 B 的灾难性溢出陷阱

bit 7 **SFTACERR**：累加器移位错误状态位
1 = 算术错误陷阱是由非法的累加器移位引起的
0 = 算术错误陷阱不是由非法的累加器移位引起的

bit 6 **DIV0ERR**：算术错误状态位
1 = 算术错误陷阱是由被零除引起的
0 = 算术错误陷阱不是由被零除引起的

bit 5 **未实现**：读为 0

bit 4 **MATHERR**：算术错误状态位
1 = 发生了算术错误陷阱
0 = 未发生算术错误陷阱

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

寄存器 28-3： INTCON1：中断控制寄存器 1（续）

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

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

© 2008 Microchip Technology Inc.

寄存器 28-4： INTCON2：中断控制寄存器 2

高字节：							
R/W-0	R-0	U-0	U-0	U-0	U-0	U-0	U-0
ALTIVT	DISI	—	—	—	—	—	—
bit 15							bit 8

低字节：							
U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	INT2EP	INT1EP	INT0EP
bit 7							bit 0

bit 15 **ALTIVT**：备用中断向量表使能位
1 = 使用备用中断向量表
0 = 使用标准 （默认）中断向量表

bit 14 **DISI**：禁止中断 （DISI）指令状态位
1 = DISI 指令有效
0 = DISI 指令无效

bit 13-3 **未实现**：读为 0

bit 2 **INT2EP**：外部中断 2 边沿检测极性选择位
1 = 下降沿中断
0 = 上升沿中断

bit 1 **INT1EP**：外部中断 1 边沿检测极性选择位
1 = 下降沿中断
0 = 上升沿中断

bit 0 **INT0EP**：外部中断 0 边沿检测极性选择位
1 = 下降沿中断
0 = 上升沿中断

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

寄存器 28-5： IFS0：中断标志状态寄存器 0

高字节：							
U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	MI2CIF	SI2CIF	NVMIF	ADIF	U1TXIF	U1RXIF	SPI1IF
bit 15							bit 8

低字节：							
R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
T3IF	T2IF	OC2IF	—	T1IF	OC1IF	IC1IF	INT0IF
bit 7							bit 0

bit 15 **未实现**：读为 0

bit 14 **MI2CIF**：I^2C 主事件中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求

bit 13 **SI2CIF**：I^2C 从事件中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求

bit 12 **NVMIF**：非易失性存储器写完成中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求

bit 11 **ADIF**：ADC 转换完成中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求

bit 10 **U1TXIF**：UART1 发送器中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求

bit 9 **U1RXIF**：UART1 接收器中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求

bit 8 **SPI1IF**：SPI1 中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求

bit 7 **T3IF**：Timer3 中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求

bit 6 **T2IF**：Timer2 中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求

bit 5 **OC2IF**：输出比较通道 2 中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求

bit 4 **未实现**：读为 0

bit 3 **T1IF**：Timer1 中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求

bit 2 **OC1IF**：输出比较通道 1 中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求

© 2008 Microchip Technology Inc.

寄存器 28-5： IFS0：中断标志状态寄存器 0（续）

bit 1 **IC1IF**：输入捕捉通道 1 中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求

bit 0 **INT0IF**：外部中断 0 标志状态位
1 = 产生了中断请求
0 = 未产生中断请求

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

寄存器 28-6： IFS1：中断标志状态寄存器 1

高字节：							
R/W-0	R/W-0	R/W-0	U-0	R/W-0	U-0	U0	U-0
AC3IF	AC2IF	AC1IF	—	CNIF	—	—	—
bit 15							bit 8

低字节：							
U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	PWM4IF	PWM3IF	PWM2IF	PWM1IF	PSEMIF	INT2IF	INT1IF
bit 7							bit 0

bit 15 **AC3IF**：模拟比较器 3 中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求

bit 14 **AC2IF**：模拟比较器 2 中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求

bit 13 **AC1IF**：模拟比较器 1 中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求

bit 12 **未实现**：读为 0

bit 11 **CNIF**：输入电平变化通知中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求

bit 10-7 **未实现**：读为 0

bit 6 **PWM4IF**：脉宽调制发生器 4 中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求

bit 5 **PWM3IF**：脉宽调制发生器 3 中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求

bit 4 **PWM2IF**：脉宽调制发生器 2 中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求

bit 3 **PWM1IF**：脉宽调制发生器 1 中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求

bit 2 **PSEMIF**：PWM 特殊事件匹配中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求

bit 1 **INT2IF**：外部中断 2 标志状态位
1 = 产生了中断请求
0 = 未产生中断请求

bit 0 **INT1IF**：外部中断 1 标志状态位
1 = 产生了中断请求
0 = 未产生中断请求

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

© 2008 Microchip Technology Inc.

寄存器 28-7： IFS2：中断标志状态寄存器 2

高字节：							
U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	ADCP5IF	ADCP4IF	ADCP3IF
bit 15							bit 8

低字节：							
R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0	R/W-0
ADCP2IF	ADCP1IF	ADCP0IF	—	—	—	—	AC4IF
bit 7							bit 0

bit 15-11 **未实现：**读为 0

bit 10 **ADCP5IF：**ADC 输入对 5 转换完成中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求

bit 9 **ADCP4IF：**ADC 输入对 4 转换完成中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求

bit 8 **ADCP3IF：**ADC 输入对 3 转换完成中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求

bit 7 **ADCP2IF：**ADC 输入对 2 转换完成中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求

bit 6 **ADCP1IF：**ADC 输入对 1 转换完成中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求

bit 5 **ADCP0IF：**ADC 输入对 0 转换完成中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求

bit 4-1 **未实现：**读为 0

bit 0 **AC4IF：**模拟比较器 4 中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求

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

寄存器 28-8： IEC0：中断允许控制寄存器 0

高字节：							
U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	MI2CIE	SI2CIE	NVMIE	ADIE	U1TXIE	U1RXIE	SPI1IE
bit 15							bit 8

低字节：							
R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
T3IE	T2IE	OC2IE	—	T1IE	OC1IE	IC1IE	INT0IE
bit 7							bit 0

bit 15 **未实现：**读为 0

bit 14 **MI2CIE：**I^2C 主事件中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 13 **SI2CIE：**I^2C 从事件中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 12 **NVMIE：**非易失性存储器写完成中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 11 **ADIE：**ADC 转换完成中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 10 **U1TXIE：**UART1 发送器中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 9 **U1RXIE：**UART1 接收器中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 8 **SPI1IE：**SPI1 中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 7 **T3IE：**Timer3 中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 6 **T2IE：**Timer2 中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 5 **OC2IE：**输出比较通道 2 中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 4 **未实现：**读为 0

bit 3 **T1IE：**Timer1 中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 2 **OC1IE：**输出比较通道 1 中断允许位
1 = 允许中断请求
0 = 禁止中断请求

© 2008 Microchip Technology Inc.

寄存器 28-8： IEC0：中断允许控制寄存器 0（续）

bit 1 **IC1IE**：输入捕捉通道 1 中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 0 **INT0IE**：外部中断 0 允许位
1 = 允许中断请求
0 = 禁止中断请求

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

寄存器 28-9： IEC1：中断允许控制寄存器 1

高字节：							
R/W-0	R/W-0	R/W-0	U-0	R/W-0	U-0	U-0	U-0
AC3IE	AC2IE	AC1IE	—	CNIE	—	—	—
bit 15							bit 8

低字节：							
U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	PWM4IE	PWM3IE	PWM2IE	PWM1IE	PSEMIE	INT2IE	INT1IE
bit 7							bit 0

bit 15 **AC3IE**：模拟比较器 3 中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 14 **AC2IE**：模拟比较器 2 中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 13 **AC1IE**：模拟比较器 1 中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 12 **未实现**：读为 0

bit 11 **CNIE**：输入电平变化通知中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 10-7 **未实现**：读为 0

bit 6 **PWM4IE**：脉宽调制发生器 4 中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 5 **PWM3IE**：脉宽调制发生器 3 中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 4 **PWM2IE**：脉宽调制发生器 2 中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 3 **PWM1IE**：脉宽调制发生器 1 中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 2 **PSEMIE**：PWM 特殊事件匹配中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 1 **INT2IE**：外部中断 2 允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 0 **INT1IE**：外部中断 1 允许位
1 = 允许中断请求
0 = 禁止中断请求

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

© 2008 Microchip Technology Inc.

寄存器 28-10： IEC2：中断允许控制寄存器 2

高字节：							
U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	ADCP5IE	ADCP4IE	ADCP3IE
bit 15							bit 8

低字节：							
R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0	R/W-0
ADCP2IE	ADCP1IE	ADCP0IE	—	—	—	—	AC4IE
bit 7							bit 0

bit 15-11 **未实现：**读为 0

bit 10 **ADCP5IE：**ADC 输入对 5 转换完成中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 9 **ADCP4IE：**ADC 输入对 4 转换完成中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 8 **ADCP3IE：**ADC 输入对 3 转换完成中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 7 **ADCP2IE：**ADC 输入对 2 转换完成中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 6 **ADCP1IE：**ADC 输入对 1 转换完成中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 5 **ADCP0IE：**ADC 输入对 0 转换完成中断允许位
1 = 允许中断请求
0 = 禁止中断请求

bit 4-1 **未实现：**读为 0

bit 0 **AC4IE：**模拟比较器 4 中断允许位
1 = 允许中断请求
0 = 禁止中断请求

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

寄存器 28-11： IPC0：中断优先级控制寄存器 0

高字节：							
U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	T1IP<2:0>			—	OC1IP<2:0>		
bit 15							bit 8

低字节：							
U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	IC1IP<2:0>			—	INT0IP<2:0>		
bit 7							bit 0

bit 15 **未实现：**读为 0

bit 14-12 **T1IP<2:0>：**Timer1 中断优先级位
111 = 中断优先级为 7（最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 11 **未实现：**读为 0

bit 10-8 **OC1IP<2:0>：**输出比较通道 1 中断优先级位
111 = 中断优先级为 7（最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 7 **未实现：**读为 0

bit 6-4 **IC1IP<2:0>：**输入捕捉通道 1 中断优先级位
111 = 中断优先级为 7（最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 3 **未实现：**读为 0

bit 2-0 **INT0IP<2:0>：**外部中断 0 优先级位
111 = 中断优先级为 7（最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

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

© 2008 Microchip Technology Inc.

寄存器 28-12： IPC1：中断优先级控制寄存器 1

高字节：							
U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	T3IP<2:0>			—	T2IP<2:0>		
bit 15							bit 8

低字节：							
U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	OC2IP<2:0>			—	—	—	—
bit 7							bit 0

bit 15 **未实现：**读为 0

bit 14-12 **T3IP<2:0>：**Timer3 中断优先级位
111 = 中断优先级为 7（最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 11 **未实现：**读为 0

bit 10-8 **T2IP<2:0>：**Timer2 中断优先级位
111 = 中断优先级为 7（最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 7 **未实现：**读为 0

bit 6-4 **OC2IP<2:0>：**输出比较通道 2 中断优先级位
111 = 中断优先级为 7（最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 3-0 **未实现：**读为 0

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

寄存器 28-13： IPC2：中断优先级控制寄存器 2

高字节：							
U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	ADIP<2:0>			—	U1TXIP<2:0>		
bit 15							bit 8

低字节：							
U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	U1RXIP<2:0>			—	SPI1IP<2:0>		
bit 7							bit 0

bit 15 **未实现：**读为 0

bit 14-12 **ADIP<2:0>：**ADC 转换完成中断优先级位
111 = 中断优先级为 7（最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 11 **未实现：**读为 0

bit 10-8 **U1TXIP<2:0>：**UART1 发送器中断优先级位
111 = 中断优先级为 7（最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 7 **未实现：**读为 0

bit 6-4 **U1RXIP<2:0>：**UART1 接收器中断优先级位
111 = 中断优先级为 7（最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 3 **未实现：**读为 0

bit 2-0 **SPI1IP<2:0>：**SPI1 中断优先级位
111 = 中断优先级为 7（最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

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

© 2008 Microchip Technology Inc.

寄存器 28-14： IPC3：中断优先级控制寄存器 3

高字节：							
U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	MI2C1IP<2:0>		
bit 15							bit 8

低字节：							
U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	SI2CIP<2:0>			—	NVMIP<2:0>		
bit 7							bit 0

bit 15-11 **未实现：**读为 0

bit 10-8 **MI2CIP<2:0>：**I^2C 主事件中断优先级位
111 = 中断优先级为 7（最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 7 **未实现：**读为 0

bit 6-4 **SI2CIP<2:0>：**I^2C 从事件中断优先级位
111 = 中断优先级为 7（最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 3 **未实现：**读为 0

bit 2-0 **NVMIP<2:0>：**非易失性存储器写中断优先级位
111 = 中断优先级为 7（最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

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

寄存器 28-15： IPC4：中断优先级控制寄存器 4

高字节：							
U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	PWM1IP<2:0>			—	PSEMIP<2:0>		
bit 15							bit 8

低字节：							
U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	INT2IP<2:0>			—	INT1IP<2:0>		
bit 7							bit 0

bit 15 **未实现：**读为 0

bit 14-12 **PWM1IP<2:0>：**PWM 发生器 1 中断优先级位
111 = 中断优先级为 7（最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 11 **未实现：**读为 0

bit 10-8 **PSEMIP<2:0>：**PWM 特殊事件匹配中断优先级位
111 = 中断优先级为 7（最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 7 **未实现：**读为 0

bit 6-4 **INT2IP<2:0>：**外部中断 2 优先级位
111 = 中断优先级为 7（最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 3 **未实现：**读为 0

bit 2-0 **INT1IP<2:0>：**外部中断 1 优先级位
111 = 中断优先级为 7（最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

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

© 2008 Microchip Technology Inc.

寄存器 28-16： IPC5：中断优先级控制寄存器 5

高字节：							
U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	PWM4IP<2:0>		
bit 15							bit 8

低字节：							
U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	PWM3IP<2:0>			—	PWM2IP<2:0>		
bit 7							bit 0

bit 15-11 **未实现：**读为 0

bit 10-8 **PWM4IP<2:0>：**PWM 发生器 4 中断优先级位
111 = 中断优先级为 7（最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 7 **未实现：**读为 0

bit 6-4 **PWM3IP<2:0>：**PWM 发生器 3 中断优先级位
111 = 中断优先级为 7（最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 3 **未实现：**读为 0

bit 2-0 **PWM2IP<2:0>：**PWM 发生器 2 中断优先级位
111 = 中断优先级为 7（最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

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

寄存器 28-17： IPC6：中断优先级控制寄存器 6

高字节：							
U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	CNIP<2:0>			—	—	—	—
bit 15							bit 8

低字节：							
U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

bit 15 **未实现：**读为 0

bit 14-12 **CNIP<2:0>：**电平变化通知中断优先级位
111 = 中断优先级为 7（最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 11-0 **未实现：**读为 0

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

© 2008 Microchip Technology Inc.

寄存器 28-18： IPC7：中断优先级控制寄存器 7

高字节：							
U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	AC3IP<2:0>			—	AC2IP<2:0>		
bit 15							bit 8

低字节：							
U-0	R/W-1	R/W-0	R/W-0	U-0	U-1	U-0	U-0
—	AC1IP<2:0>			—	—	—	—
bit 7							bit 0

bit 15 **未实现：**读为 0

bit 14-12 **AC3IP<2:0>：**模拟比较器 3 中断优先级位
111 = 中断优先级为 7（最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 11 **未实现：**读为 0

bit 10-8 **AC2IP<2:0>：**模拟比较器 2 中断优先级位
111 = 中断优先级为 7（最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 7 **未实现：**读为 0

bit 6-4 **AC1IP<2:0>：**模拟比较器 1 中断优先级位
111 = 中断优先级为 7（最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 3-0 **未实现：**读为 0

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

寄存器 28-19： IPC8：中断优先级控制寄存器 8

高字节：							
U-0	U-0	U-0	U-0	U-0	U0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

低字节：							
U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	AC4IP<2:0>		
bit 7							bit 0

bit 15-3 **未实现：**读为 0

bit 2-0 **AC4IP<2:0>：**模拟比较器 4 中断优先级位
111 = 中断优先级为 7（最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

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

© 2008 Microchip Technology Inc.

寄存器 28-20： IPC9：中断优先级控制寄存器 9

高字节：							
U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	ADCP2IP<2:0>			—	ADCP1IP<2:0>		
bit 15							bit 8

低字节：							
U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	ADCP0IP<2:0>			—	—	—	—
bit 7							bit 0

bit 15 **未实现：**读为 0

bit 14-12 **ADCP2IP<2:0>：**ADC 输入对 2 转换完成中断优先级位
111 = 中断优先级为 7（最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 11 **未实现：**读为 0

bit 10-8 **ADCP1IP<2:0>：**ADC 输入对 1 转换完成中断优先级位
111 = 中断优先级为 7（最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 7 **未实现：**读为 0

bit 6-4 **ADCP0IP<2:0>：**ADC 输入对 0 转换完成中断优先级位
111 = 中断优先级为 7（最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 3-0 **未实现：**读为 0

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

寄存器 28-21： IPC10：中断优先级控制寄存器 10

高字节：							
U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	ADCP5IP<2:0>		
bit 15							bit 8

低字节：							
U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	ADCP4IP<2:0>			—	ADCP3IP<2:0>		
bit 7							bit 0

bit 15-11 **未实现：**读为 0

bit 10-8 **ADCP5IP<2:0>：**ADC 输入对 5 转换完成中断优先级位
111 = 中断优先级为 7（最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 7 **未实现：**读为 0

bit 6-4 **ADCP4IP<2:0>：**ADC 输入对 4 转换完成中断优先级位
111 = 中断优先级为 7（最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

bit 3 **未实现：**读为 0

bit 2-0 **ADCP3IP<2:0>：**ADC 输入对 3 转换完成中断优先级位
111 = 中断优先级为 7（最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

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

© 2008 Microchip Technology Inc.

寄存器 28-22： INTTREG：中断控制和状态寄存器

高字节：							
U-0	U-0	U-0	U-0	R-0	R-0	R-0	R-0
—	—	—	—	ILR<3:0>			
bit 15							bit 8

低字节：							
U-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
—	VECNUM<6:0>						
bit 7							bit 0

bit 15-12 **未实现：**读为 0

bit 11-8 **ILR<3:0>：**新的 CPU 中断优先级位

1111 = CPU 中断优先级为 15
•
•
•
0001 = CPU 中断优先级为 1
0000 = CPU 中断优先级为 0

bit 7 **未实现：**读为 0

bit 6-0 **VECNUM<6:0>：**待处理中断的向量编号位

0111111 = 待处理中断向量的编号为 135
•
•
•
0000001 = 待处理中断向量的编号为 9
0000000 = 待处理中断向量的编号为 8

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

28.5 中断设置步骤

下面的章节描述中断设置步骤。

28.5.1 初始化

下面的步骤描述如何设置中断源：

1. 如果不需要嵌套中断，将 NSTDIS 控制位 （INTCON1<15>）置 1。
2. 通过写相应 IPCx 控制寄存器中的控制位为中断源选择用户分配优先级。优先级取决于具体的应用和中断源类型。如果不需要多个优先级，则可将所有允许中断源的 IPCx 寄存器控制位编程为相同的非零值。

注： 在器件复位时， IPC 寄存器被初始化，为所有用户中断源分配优先级 4。

3. 将相应 IFSx 状态寄存器中与外设相关的中断标志状态位清零。
4. 通过将相应 IECx 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。

28.5.2 中断服务程序

用于声明 ISR 以及使用正确的向量地址初始化 IVT 的方法，将依赖于编程语言（C 语言或汇编语言）和用于开发应用程序的语言开发工具包。一般情况下，用户必须将相应 IFSx 寄存器中与 ISR 处理的中断源相对应的中断标志清零。否则，在退出 ISR 程序后将立即再次进入 ISR。如果 ISR 用汇编语言编写，则必须使用 `RETFIE` 指令结束 ISR，以便将保存的 PC 值、 SRL 值和原先的 CPU 优先级弹出堆栈。

28.5.3 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 （Trap Service Routine， TSR）之外， TSR 使用与 ISR 类似的方式编写。

28.5.4 中断禁止

可以使用以下步骤禁止所有用户中断：

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 进行逻辑或运算来强制将 CPU 的优先级设置为 7。

要允许用户中断，则可使用 `POP` 指令恢复先前的 SR 值。

要注意只能禁止优先级小于或等于 7 的用户中断。不能禁止陷阱源 （优先级为 8-15）。

使用 `DISI` 指令可以方便地将优先级为 1-6 的中断禁止一段固定的时间。`DISI` 指令不能禁止优先级为 7 的中断源。

© 2008 Microchip Technology Inc.

28.5.5 代码示例

如下例 28-1 中的代码使能中断嵌套，并将 Timer1、 Timer2 和 Timer3 和电平变化通知外设中断的中断优先级分别设置为 2、5、3 和 4。此示例还说明了如何通过状态寄存器来允许和禁止中断。示例 ISR 说明了中断的清除。

例 28-1： 中断嵌套代码示例

```
void enableInterrupts(void)
{
/* Set CPU IPL to 0, enable level 1-7 interrupts */
/* No restoring of previous CPU IPL state performed here */
SRbits.IPL = 0;
return;
}
void disableInterrupts(void)
{
/* Set CPU IPL to 7, disable level 1-7 interrupts */
/* No saving of current CPU IPL setting performed here */
SRbits.IPL = 7;
return;
}
void initInterrupts(void)
{
/* Interrupt nesting enabled here */
INTCON1bits.NSTDIS = 0;
/* Set Timer3 interrupt priority to 3 (level 7 is highest) */
IPC1bits.T3IP = 3;
/* Set Timer2 interrupt priority to 5 */
IPC1bits.T2IP = 5;
/* Set Change Notice interrupt priority to 4 */
IPC6bits.CNIP = 4;
/* Set Timer1 interrupt priority to 2 */
IPC0bits.T1IP = 2;
/* Reset Timer1 interrupt flag */
IFS0bits.T1IF = 0;
/* Reset Timer2 interrupt flag */
IFS0bits.T2IF = 0;
/* Reset Timer3 interrupt flag */
IFS0bits.T3IF = 0;
/* Enable CN interrupts */
IEC1bits.CNIE = 1;
/* Enable Timer1 interrupt */
IEC0bits.T1IE = 1;
/* Enable Timer2 interrupt (PWM time base) */
IEC0bits.T2IE = 1;
/* Enable Timer3 interrupt (Replacement for Timer2)*/
IEC0bits.T3IE = 1;
/* Reset change notice interrupt flag */
IFS1bits.CNIF = 0;
return;
}
void __attribute__((__interrupt__)) _T1Interrupt(void)
{
/* Insert ISR Code Here*/
/* Clear Timer1 interrupt */
IFS0bits.T1IF = 0;
}
```

例 28-1：　　中断嵌套代码示例 （续）

```
void __attribute__((__interrupt__)) _T2Interrupt(void)
{
/* Insert ISR Code Here*/
/* Clear Timer2 interrupt */
IFS0bits.T2IF = 0;
}
void __attribute__((__interrupt__)) _T3Interrupt(void)
{
/* Insert ISR Code Here*/
/* Clear Timer3 interrupt */
IFS0bits.T3IF = 0;
}
void __attribute__((__interrupt__)) _CNInterrupt(void)
{
/* Insert ISR Code Here*/
/* Clear CN interrupt */
IFS1bits.CNIF = 0;

}
```


© 2008 Microchip Technology Inc.

表 28-3：　　与中断控制器相关的特殊功能寄存器

SFR 名称	地址	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	复位时的状态
INTCON1	0080	NSTDIS	OVAERR	OVBERR	COVAERR	COVBERR	OVATE	OVBTE	COVTE	SFTACERR	DIV0ERR	—	MATHERR	ADDRERR	STKERR	OSCFAIL	—	0000 0000 0000 0000
INTCON2	0082	ALTIVT	DISI	—	—	—	—	—	—	—	—	—	—	—	INT2EP	INT1EP	INT0EP	0000 0000 0000 0000
IFS0	0084	—	MI2CIF	SI2CIF	NVMIF	ADIF	U1TXIF	U1RXIF	SPI1IF	T3IF	T2IF	OC2IF	—	T1IF	OC1IF	IC1IF	INT0IF	0000 0000 0000 0000
IFS1	0086	AC3IF	AC2IF	AC1IF	—	CNIF	—	—	—	—	PWM4IF	PWM3IF	PWM2IF	PWM1IF	PSEMIF	INT2IF	INT1IF	0000 0000 0000 0000
IFS2	0088	—	—	—	—	—	ADCP5IF	ADCP4IF	ADCP3IF	ADCP2IF	ADCP1IF	ADCP0IF	—	—	—	—	AC4IF	0000 0000 0000 0000
IEC0	0094	—	MI2CIE	SI2CIE	NVMIE	ADIE	U1TXIE	U1RXIE	SPI1IE	T3IE	T2IE	OC2IE	—	T1IE	OC1IE	IC1IE	INT0IE	0000 0000 0000 0000
IEC1	0096	AC3IE	AC2IE	AC1IE	—	CNIE	—	—	—	—	PWM4IE	PWM3IE	PWM2IE	PWM1IE	PSEMIE	INT2IE	INT1IE	0000 0000 0000 0000
IEC2	0098	—	—	—	—	—	ADCP5IE	ADCP4IE	ADCP3IE	ADCP2IE	ADCP1IE	ADCP0IE	—	—	—	—	AC4IE	0000 0000 0000 0000
IPC0	00A4	—	T1IP<2:0>			—	OC1IP<2:0>			—	IC1IP<2:0>			—	INT0IP<2:0>			0100 0100 0100 0100
IPC1	00A6	—	T31P<2:0>			—	T2IP<2:0>			—	OC2IP<2:0>			—	—	—	—	0100 0100 0100 0000
IPC2	00A8	—	ADIP<2:0>			—	U1TXIP<2:0>			—	U1RXIP<2:0>			—	SPI1IP<2:0>			0100 0100 0100 0100
IPC3	00AA	—	—	—	—	—	MI2C1IP<2:0>			—	SI2CIP<2:0>			—	NVMIP<2:0>			0000 0000 0000 0100
IPC4	00AC	—	PWM1IP<2:0>			—	PSEMIP<2:0>			—	INT2IP<2:0>			—	INT1IP<2:0>			0100 0100 0100 0100
IPC5	00AE	—	—	—	—	—	PWM4IP<2:0>			—	PWM3IP<2:0>			—	PWM2IP<2:0>			0000 0000 0000 0100
IPC6	00B0	—	CNIP<2:0>			—	—	—	—	—	—	—	—	—	—	—	—	0100 0000 0000 0000
IPC7	00B2	—	AC3IP<2:0>			—	AC2IP<2:0>			—	AC1IP<2:0>			—	—	—	—	0100 0100 0100 0000
IPC8	00B4	—	—	—	—	—	—	—	—	—	—	—	—	—	AC4IP<2:0>			0000 0000 0000 0100
IPC9	00B6	—	ADCP2IP<2:0>			—	ADCP1IP<2:0>			—	ADCP0IP<2:0>			—	—	—	—	0100 0100 0100 0000
IPC10	00B8	—	—	—	—	—	ADCP5IP<2:0>			—	ADCP4IP<2:0>			—	ADCP3IP<2:0>			0000 0100 0100 0100
INTTREG	00E0	—	—	—	—	ILR<3:0>				—	VECNUM<6:0>							0000 0100 0100 0100

注：　　对于一个特定的器件，可能并不包含所有的中断源及其相关的控制位。更多详细信息，请参见特定的器件数据手册。

© 2008 Microchip Technology Inc.

28.6 设计技巧

问 1：当两个具有相同用户分配优先级的中断源同时等待处理时，将如何处理？

答：具有较高自然顺序优先级的中断源将优先处理。自然顺序优先级由中断源的中断向量表（IVT）地址确定。 IVT 地址小的中断源具有较高的自然顺序优先级。

问 2：可使用 DISI 指令来禁止所有中断源和陷阱源吗？

答：DISI 指令不能禁止陷阱以及优先级为 7 的中断源。但是，如果用户应用程序中未允许任何优先级为 7 的中断源，则可使用 DISI 来方便地禁止所有中断源。

© 2008 Microchip Technology Inc.

28.7 相关应用笔记

本节列出与手册的这一章相关的应用笔记。这些应用笔记可能并非都是专门针对 dsPIC SMPS DSC 产品系列而写的，但其概念是相近的，通过适当修改并受到一定的限制即可使用。目前与中断模块相关的应用笔记包括：

标题	应用笔记文档编号

目前没有相关的应用笔记。

注： 如需获得更多关于 dsPIC SMPS DSC 系列器件的应用笔记和代码示例，请访问 Microchip 网站 （www.microchip.com）。

28.8 版本历史

版本 A（2007 年 1 月）

这是此文档的初始版本。

版本 B（2008 年 2 月）

将中断向量 35 的名称更改为 “CN——输入电平变化通知”（见表 28-2）。添加了 “注”以指明在实时更改 IPL 时要禁止中断 （见第 28.1.5 节）。

© 2008 Microchip Technology Inc.