Section 6. Interrupts

HIGHLIGHTS

This section of the manual contains the following topics:

Note: This family reference manual section is meant to serve as a complement to device data sheets. Depending on the device variant, this manual section may not apply to all dsPIC33F/PIC24H devices.

Please consult the note at the beginning of the **"Interrupts"** chapter in the current device data sheet to check whether this document supports the device you are using.

Device data sheets and family reference manual sections are available for download from the Microchip Worldwide Web site at: http://www.microchip.com

6.1 INTRODUCTION

The dsPIC33F/PIC24H Interrupt Controller module reduces the numerous peripheral interrupt request signals to a single interrupt request signal to the dsPIC33F/PIC24H CPU. This module consists of the following features:

- Up to eight processor exceptions and software traps
- Seven user-selectable priority levels
- Interrupt Vector Table (IVT) with up to 126 vectors
- A unique vector for each interrupt or exception source
- Fixed priority within a specified user priority level
- Alternate Interrupt Vector Table (AIVT) for debugging support
- Fixed interrupt entry and return latencies

6.1.1 Interrupt Vector Table

The Interrupt Vector Table (IVT) as shown in Figure 6-1, resides in program memory starting at location 0x000004. The IVT contains 126 vectors consisting of eight non-maskable trap vectors and up to 118 sources of interrupt. In general, each interrupt source has its own vector. Each interrupt vector contains a 24-bit-wide address. The value programmed into each interrupt vector location is the starting address of the associated Interrupt Service Routine (ISR).

6.1.2 Alternate Interrupt Vector Table

The Alternate Interrupt Vector Table (AIVT) is located after the IVT, as shown in Figure 6-1. Access to the AIVT is provided by the Enable Alternate Interrupt Vector Table (ALTIVT) control bit in Interrupt Control Register 2 (INTCON2<15>). If the ALTIVT bit is set, all interrupt and exception process use the alternate vectors instead of the default vectors. The alternate vectors are organized in the same manner as the default vectors.

The AIVT supports emulation and debugging by providing a means to switch between an application and a support environment without reprogramming the interrupt vectors. This feature also enables switching between applications for evaluation of different software algorithms at run time. If the AIVT is not needed, the AIVT should be programmed with the same addresses used in the IVT.

6.1.3 Reset Sequence

A device Reset is not a true exception because the interrupt controller is not involved in the Reset process. The dsPIC33F/PIC24H device clears its registers during Reset, which forces the Program Counter (PC) to zero. The processor then begins program execution at location 0x000000. The user programs a `GOTO` instruction at the Reset address, which redirects program execution to the appropriate start-up routine.

Note: Any unimplemented or unused vector locations in the IVT and AIVT must be programmed with the address of a default interrupt handler routine that contains a `RESET` instruction.

Figure 6-1: Interrupt Vector Table

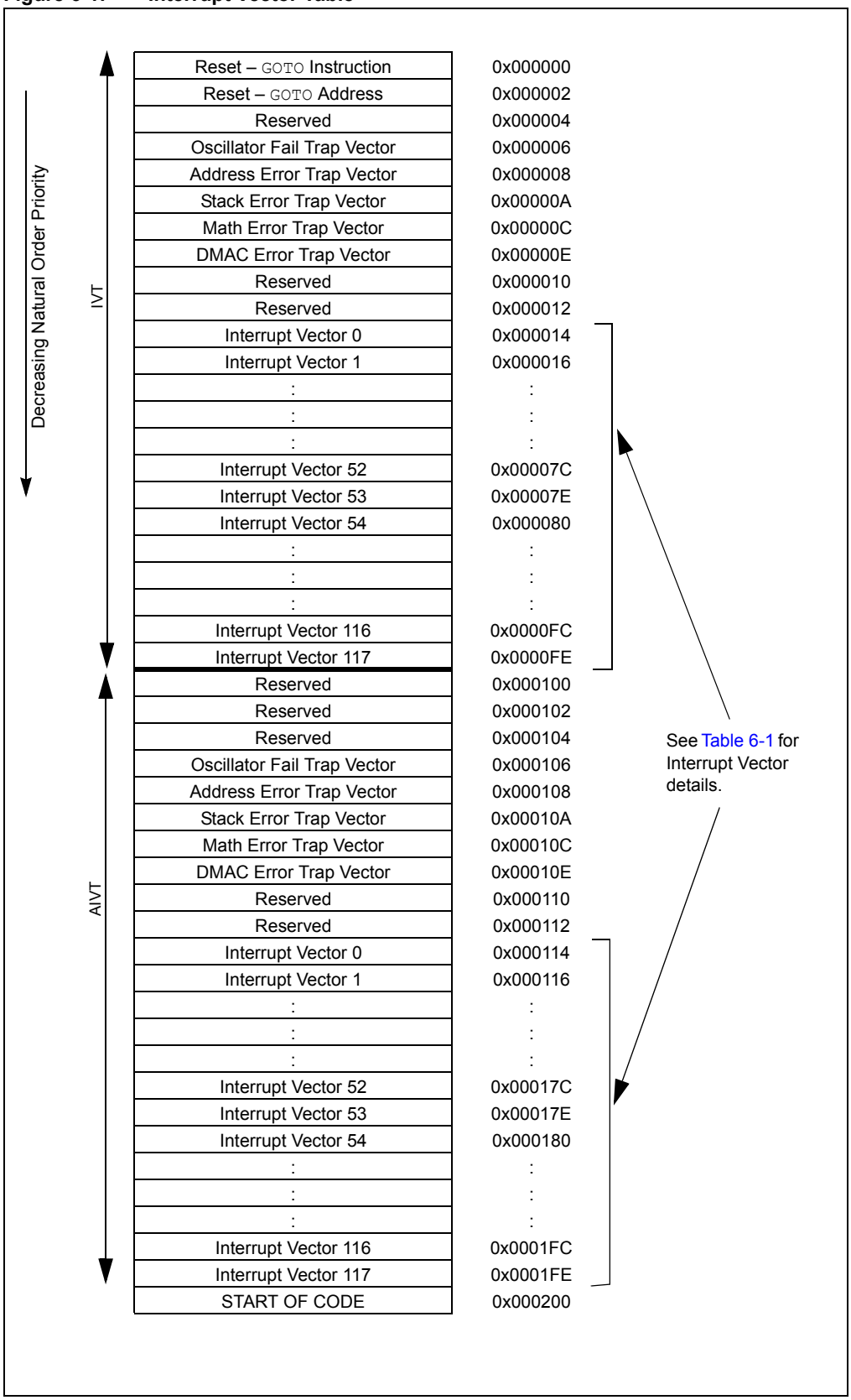


Table 6-1: Interrupt Vector Details

IRQ #	IVT Address	AIVT Address	Interrupt Source
Highest Natural Order Priority			
0	0x000004	0x000084	Reserved
1	0x000006	0x000086	Oscillator Failure
2	0x000008	0x000088	Address Error
3	0x00000A	0x00008A	Stack Error
4	0x00000C	0x00008C	Math Error
5	0x00000E	0x00008E	DMAC Error
6-7	0x000010-0x000012	0x000091-0x000092	Reserved
8	0x000014	0x000114	INT0 – External Interrupt 0
9	0x000016	0x000116	IC1 – Input Capture 1
10	0x000018	0x000118	OC1 – Output Compare 1
11	0x00001A	0x00011A	T1 – Timer1
12	0x00001C	0x00011C	DMA0 – DMA Channel 0
13	0x00001E	0x00011E	IC2 – Input Capture 2
14	0x000020	0x000120	OC2 – Output Compare 2
15	0x000022	0x000122	T2 – Timer2
16	0x000024	0x000124	T3 – Timer3
17	0x000026	0x000126	SPI1E – SPI1 Fault
18	0x000028	0x000128	SPI1 – SPI1 Transfer Done
19	0x00002A	0x00012A	U1RX – UART1 Receiver
20	0x00002C	0x00012C	U1TX – UART1 Transmitter
21	0x00002E	0x00012E	AD1 – ADC1 Convert Done
22	0x000030	0x000130	DMA1 – DMA Channel 1
23	0x000032	0x000132	Reserved
24	0x000034	0x000134	SI2C1 – I2C1 Slave Event
25	0x000036	0x000136	MI2C1 – I2C1 Master Event
26	0x000038	0x000138	Reserved
27	0x00003A	0x00013A	CN – Input Change Interrupt
28	0x00003C	0x00013C	INT1 – External Interrupt 1
29	0x00003E	0x00013E	AD2 – ADC2 Convert Done
30	0x000040	0x000140	IC7 – Input Capture 7
31	0x000042	0x000142	IC8 – Input Capture 8
32	0x000044	0x000144	DMA2 – DMA Channel 2
33	0x000046	0x000146	OC3 – Output Compare 3
34	0x000048	0x000148	OC4 – Output Compare 4
35	0x00004A	0x00014A	T4 – Timer4
36	0x00004C	0x00014C	T5 – Timer5
37	0x00004E	0x00014E	INT2 – External Interrupt 2
38	0x000050	0x000150	U2RX – UART2 Receiver
39	0x000052	0x000152	U2TX – UART2 Transmitter
40	0x000054	0x000154	SPI2E – SPI2 Fault
41	0x000056	0x000156	SPI2 – SPI2 Transfer Done
42	0x000058	0x000158	C1RX – CAN1 RX Data Ready
43	0x00005A	0x00015A	C1 – CAN1 Event
44	0x00005C	0x00015C	DMA3 – DMA Channel 3

Table 6-1: Interrupt Vector Details (Continued)

IRQ #	IVT Address	AIVT Address	Interrupt Source
45	0x00005E	0x00015E	IC3 – Input Capture 3
46	0x000060	0x000160	IC4 – Input Capture 4
47	0x000062	0x000162	IC5 – Input Capture 5
48	0x000064	0x000164	IC6 – Input Capture 6
49	0x000066	0x000166	OC5 – Output Compare 5
50	0x000068	0x000168	OC6 – Output Compare 6
51	0x00006A	0x00016A	OC7 – Output Compare 7
52	0x00006C	0x00016C	OC8 – Output Compare 8
53	0x00006E	0x00016E	Reserved
54	0x000070	0x000170	DMA4 – DMA Channel 4
55	0x000072	0x000172	T6 – Timer6
56	0x000074	0x000174	T7 – Timer7
57	0x000076	0x000176	I2C2S – I2C2 Slave Event
58	0x000078	0x000178	I2C2M – I2C2 Master Event
59	0x00007A	0x00017A	T8 – Timer8
60	0x00007C	0x00017C	T9 – Timer9
61	0x00007E	0x00017E	INT3 – External Interrupt 3
62	0x000080	0x000180	INT4 – External Interrupt 4
63	0x000082	0x000182	C2RX – CAN2 RX Data Ready
64	0x000084	0x000184	C2 – CAN2 Event
65	0x000086	0x000186	PWM – PWM Period Match
66	0x000088	0x000188	QEI – QEI Position Counter Compare
67	0x00008A	0x00018A	DCIE – DCI Fault Interrupt
68	0x00008C	0x00018C	DCI – DCI Transfer Done
69	0x00008E	0x00018E	DMA5 – DMA Channel 5
70	0x000090	0x000190	Reserved
71	0x000092	0x000192	FLTA – MPWM Fault A
72	0x000094	0x000194	FLTB – MPWM Fault B
73	0x000096	0x000196	U1E – UART1 Error Interrupt
74	0x000098	0x000198	U2E – UART2 Error Interrupt
75	0x00009A	0x00019A	Reserved
76	0x00009C	0x00019C	DMA6 – DMA Channel 6
77	0x00009E	0x00019E	DMA7 – DMA Channel 7
78	0x0000A0	0x0001A0	C1TX – CAN1 TX Data Request
79	0x0000A2	0x0001A2	C2TX – CAN2 TX Data Request
80-125	0x0000A4-0x0000FE	0x0001A4-0x0001FE	Reserved
Lowest Natural Order Priority			

6.1.4 CPU Priority Status

The CPU can operate at one of 16 priority levels that range from 0-15. An interrupt or trap source must have a priority level greater than the current CPU priority to initiate an exception process. You can program peripheral and external interrupt sources for levels 0-7. CPU priority levels 8-15 are reserved for trap sources.

A trap is a non-maskable interrupt source intended to detect hardware and software problems (see **6.2 "Non-Maskable Traps"**). The priority level for each trap source is fixed. Only one trap is assigned to a priority level. An interrupt source programmed to priority level 0 is effectively disabled, since it can never be greater than the CPU priority.

The current CPU priority level is indicated by the following status bits:

- CPU Interrupt Priority Level (IPL<2:0>) status bits in the CPU Status Register (SR<7:5>)
- CPU Interrupt Priority Level 3 (IPL3) status bit in the Core Control (CORCON<3>) register

The IPL<2:0> status bits are readable and writable, so the user application can modify these bits to disable all sources of interrupts below a given priority level. For example, if IPL<2:0> = `011`, the CPU would not be interrupted by any source with a programmed priority level of 0, 1, 2, or 3.

Trap events have higher priority than any user interrupt source. When the IPL3 bit is set, a trap event is in progress. The IPL3 bit can be cleared, but not set, by the user application. In some applications, you might need to clear the IPL3 bit when a trap has occurred and branch to an instruction other than the instruction after the one that originally caused the trap to occur.

All user interrupt sources can be disabled by setting IPL<2:0> = `111`.

Note:	The IPL<2:0> bits become read-only bits when interrupt nesting is disabled. See **6.2.4.2 "Interrupt Nesting"**, for more information.

6.1.5 Interrupt Priority

Each peripheral interrupt source can be assigned to one of seven priority levels. The user assignable interrupt priority control bits for each individual interrupt are located in the Least Significant 3 bits of each nibble within the IPCx registers. Bit 3 of each nibble is not used and is read as a '`0`'. These bits define the priority level assigned to a particular interrupt. The usable priority levels are 1 (lowest priority) through 7 (highest priority). If the IPC bits associated with an interrupt source are all cleared, the interrupt source is effectively disabled.

Note:	The application program must disable the interrupts while reconfiguring the interrupt priority levels on-the-fly. Failure to disable interrupts can produce unexpected results.

More than one interrupt request source can be assigned to a specific priority level. To resolve priority conflicts within a given user application-assigned level, each source of interrupt has a natural order priority based on its location in the IVT. Table 6-1 shows the location of each interrupt source in the IVT. The lower numbered interrupt vectors have higher natural priority, while the higher numbered vectors have lower natural priority. The overall priority level for any pending source of interrupt is first determined by the user-assigned priority of that source in the IPCx register, then by the natural order priority within the IVT.

Natural order priority is used only to resolve conflicts between simultaneous pending interrupts with the same user assigned priority level. Once the priority conflict is resolved and the exception process begins, the CPU can be interrupted only by a source with higher user-assigned priority. Interrupts with the same user-assigned priority, but a higher natural order priority that become pending during the exception process, remain pending until the current exception process completes.

Assigning each interrupt source to one of seven priority levels enables the user application to give an interrupt with a low natural order priority and a very high overall priority level. For example, the UART1 RX Interrupt can be given a priority of 7, and the External Interrupt 0 (INT0) can be assigned to priority level 1, thus giving it a very low effective priority.

Note:	The peripherals and sources of interrupt available in the IVT vary depending on the specific device. The sources of interrupt shown in this document represent a comprehensive listing of all interrupt sources found on dsPIC33F/PIC24H devices. Refer to the specific device data sheet for further details.

6.2 NON-MASKABLE TRAPS

Traps are non-maskable, nestable interrupts that adhere to a fixed priority structure. Traps provide a means to correct erroneous operation during debugging and operation of the application. If the user application does not intend to correct a trap error condition, these vectors must be loaded with the address of a software routine to reset the device. Otherwise, the user application programs the trap vector with the address of a service routine that corrects the trap condition.

The following sources of non-maskable traps are implemented in dsPIC33F/PIC24H devices:

- Oscillator Failure Trap
- Stack Error Trap
- Address Error Trap
- Math Error Trap
- DMAC Error Trap

For many of the trap conditions, the instruction that caused the trap is allowed to complete before exception processing begins. Therefore, the user application may have to correct the action of the instruction that caused the trap.

Each trap source has a fixed priority as defined by its position in the IVT. An oscillator failure trap has the highest priority, while a DMA Controller (DMAC) error trap has the lowest priority (see Figure 6-1). In addition, trap sources are classified into two distinct categories: soft traps and hard traps.

6.2.1 Soft Traps

The DMAC error trap (priority level 10), math error trap (priority level 11), and stack error trap (priority level 12) are categorized as soft trap sources. Soft traps can be treated like non-maskable sources of interrupt that adhere to the priority assigned by their position in the IVT. Soft traps are processed like interrupts and require two cycles to be sampled and acknowledged prior to exception processing. Therefore, additional instructions may be executed before a soft trap is acknowledged.

6.2.1.1 STACK ERROR TRAP (SOFT TRAP, LEVEL 12)

The stack is initialized to 0x0800 during a Reset. A stack error trap is generated, if the Stack Pointer address is less than 0x0800.

A Stack Limit (SPLIM) register associated with the Stack Pointer is uninitialized at Reset. The stack overflow check is not enabled until a word is written to the SPLIM register.

All Effective Addresses (EA) generated using W15 as a source or destination pointer are compared against the value in the SPLIM register. If the EA is greater than the contents of the SPLIM register, a stack error trap is generated. In addition, a stack error trap is generated if the EA calculation wraps over the end of data space (0xFFFF).

A stack error can be detected in software by polling the Stack Error Trap (STKERR) status bit (INTCON1<2>). To avoid re-entering the Trap Service Routine, the STKERR status flag must be cleared (in software) before the program returns from the trap (with a `RETFIE` instruction).

6.2.1.2 MATH ERROR TRAP (SOFT TRAP, LEVEL 11)

Any of the following events generate a math error trap:

- Accumulator A overflow
- Accumulator B overflow
- Catastrophic accumulator overflow
- Divide by zero
- Shift Accumulator (SFTAC) operation that exceeds ±16 bits

Three bits in the INTCON1 register enable three types of accumulator overflow traps.

- The Accumulator A Overflow Trap Flag (OVATE) control bit (INTCON1<10>) enables traps for an Accumulator A overflow event.
- The Accumulator B Overflow Trap Flag (OVBTE) control bit (INTCON1<9>) enables traps for an Accumulator B overflow event.
- The Catastrophic Overflow Trap Enable (COVTE) control bit (INTCON1<8>) enables traps for a catastrophic overflow of either accumulator. When this trap is detected, these corresponding ERROR bits are set in the INTCON1 register:
  - Accumulator A Overflow Trap Flag (OVAERR)
  - Accumulator B Overflow Trap Flag (OVBERR)
  - Accumulator A Catastrophic Overflow Trap Enable (COVAERR)
  - Accumulator B Catastrophic Overflow Trap Enable (COVBERR)

An Accumulator A or Accumulator B overflow event is defined as a carry-out from bit 31. The accumulator overflow cannot occur if the 31-bit Saturation mode is enabled for the accumulator. A catastrophic accumulator overflow is defined as a carry-out from bit 39 of either accumulator. The catastrophic overflow cannot occur if accumulator saturation (31-bit or 39-bit) is enabled.

Divide-by-zero traps cannot be disabled. The divide-by-zero check is performed during the first iteration of the REPEAT loop that executes the divide instruction. The Math Error Status (DIV0ERR) bit (INTCON1<6>) is set when this trap is detected.

Accumulator shift traps cannot be disabled. The SFTAC instruction can be used to shift the accumulator by a literal value or a value in one of the W registers. If the shift value exceeds ±16 bits, an arithmetic trap is generated and the Shift Accumulator Error Status (SFTACERR) bit (INTCON1<7>) is set. The SFTAC instruction executes, but the results of the shift are not written to the target accumulator.

A math error trap can be detected in software by polling the Math Error Status (MATHERR) bit (INTCON1<4>). To avoid re-entering the Trap Service Routine, the MATHERR status flag must be cleared (in software) before the program returns from the trap (with a RETFIE instruction). Before the MATHERR status bit can be cleared, all conditions that caused the trap to occur must also be cleared. If the trap was due to an accumulator overflow, the Accumulator Overflow (OA and OB) status bits (SR<15:14>) must be cleared. The OA and OB status bits are read-only, so the user software must perform a dummy operation on the overflowed accumulator (such as adding '0'), which will cause the hardware to clear the OA or OB status bit.

6.2.1.3 DMAC ERROR TRAP (SOFT TRAP, LEVEL 10)

A DMAC error trap occurs with these conditions:

- RAM write collision
- DMA-ready peripheral RAM write collision

Write collision errors are a serious enough threat to system integrity to warrant a non-maskable CPU trap event. If Both the CPU and a DMA channel attempt to write to a target address, the CPU is given priority and the DMA write is ignored. In this case, a DMAC error trap is generated and the DMAC Error Status (DMACERR) bit (INTCON1<5>) is set.

6.2.2 Hard Traps

Hard traps include exceptions of priority level 13 through level 15, inclusive. The address error (level 13) and oscillator error (level 14) traps fall into this category.

Like soft traps, hard traps are non-maskable sources of interrupt. The difference between hard traps and soft traps is that hard traps force the CPU to stop code execution after the instruction causing the trap has completed. Normal program execution flow does not resume until the trap has been acknowledged and processed.

6.2.2.1 TRAP PRIORITY AND HARD TRAP CONFLICTS

If a higher priority trap occurs while any lower priority trap is in progress, processing of the lower-priority trap is suspended. The higher-priority trap is acknowledged and processed. The lower-priority trap remains pending until processing of the higher priority trap completes.

Each hard trap that occurs must be acknowledged before code execution of any type can continue. If a lower-priority hard trap occurs while a higher priority trap is pending, acknowledged or is being processed, a hard-trap conflict occurs because the lower-priority trap cannot be acknowledged until processing for the higher-priority trap completes.

The device is automatically reset in a hard-trap conflict condition. The Trap Reset Flag (TRAPR) status bit in the Reset Control Register (RCON<15>) in the Reset module, is set when the Reset occurs so that the condition can be detected in software.

6.2.2.2 OSCILLATOR ERROR TRAP (HARD TRAP, LEVEL 14)

An oscillator failure trap event is generated for any of these reasons:

- The Fail-Safe Clock Monitor (FSCM) is enabled and has detected a loss of the system clock source
- A loss of PLL lock has been detected during normal operation using the PLL
- The FSCM is enabled and the PLL fails to achieve lock at a Power-on Reset (POR)

An oscillator failure trap event can be detected in software by polling the Oscillator Failure Trap (OSCFAIL) status bit (INTCON1<1>) or the Clock Fail (CF) status bit (OSCCON<3> in the Oscillator module). To avoid re-entering the Trap Service Routine, the OSCFAIL status flag must be cleared (in software) before the program returns from the trap (with a `RETFIE` instruction).

Refer to the **Section 7 “Oscillator”** (DS70186) and **Section 25 “Device Configuration”** (DS70194), for more information about the Fail-Safe Clock Monitor. Refer to the Microchip web site at www.microchip.com for the latest documentation.

6.2.2.3 ADDRESS ERROR TRAP (HARD TRAP, LEVEL 13)

Operating conditions that can generate an address error trap include:

- A misaligned data word fetch is attempted. This condition occurs when an instruction performs a word access with the Least Significant bit (LSb) of the effective address set to ‘`1`’. The dsPIC33F/PIC24H CPU requires all word accesses to be aligned to an even address boundary.
- A bit manipulation instruction uses the Indirect Addressing mode with the LSb of the effective address set to ‘`1`’
- A data fetch is attempted from unimplemented data address space.
- Execution of a `BRA #literal` instruction or a `GOTO #literal` instruction, where `literal` is an unimplemented program memory address
- Execution of instructions after the Program Counter has been modified to point to unimplemented program memory addresses. The Program Counter can be modified by loading a value into the stack and executing a `RETURN` instruction.

When an address error trap occurs, data space writes are inhibited so that data is not destroyed.

An address error can be detected in software by polling the ADDRERR status bit (INTCON1<3>). To avoid re-entering the Trap Service Routine (TSR), the ADDRERR status flag must be cleared (in software) before the program returns from the trap (with a RETFIE instruction).

Note: In the MAC class of instructions, the data space is split into X and Y spaces. In these instructions, unimplemented X space includes all of Y space, and unimplemented Y space includes all of X space.

6.2.3 Disable Interrupts Instruction

The DISI (Disable Interrupts) instruction can disable interrupts for up to 16384 instruction cycles. This instruction is useful for executing time-critical code segments.

The DISI instruction only disables interrupts with priority levels 1-6. Priority level 7 interrupts and all trap events can still interrupt the CPU when the DISI instruction is active.

The DISI instruction works in conjunction with the Disable Interrupts Count (DISICNT) register in the CPU. When the DISICNT register is non-zero, priority level 1-6 interrupts are disabled. The DISICNT register is decremented on each subsequent instruction cycle. When the DISICNT register counts down to zero, priority level 1-6 interrupts are re-enabled. The value specified in the DISI instruction includes all cycles due to PSV accesses, instruction stalls, and so on.

The DISICNT register is both readable and writable. The user application can terminate the effect of a previous DISI instruction early by clearing the DISICNT register. The time that interrupts are disabled can also be increased by writing to or adding to the DISICNT register.

If the DISICNT register is zero, interrupts cannot be disabled by simply writing a non-zero value to the register. Interrupts must first be disabled by using the DISI instruction. Once the DISI instruction has executed and DISICNT holds a non-zero value, the application can extend the interrupt disable time by modifying the contents of DISICNT.

The DISI Instruction (DISI) status bit (INTCON2<14>) is set whenever interrupts are disabled as a result of the DISI instruction.

Note: The DISI instruction can be used to quickly disable all user interrupt sources, if no source is assigned to CPU priority level 7.

6.2.4 Interrupt Operation

All interrupt event flags are sampled during each instruction cycle. A pending Interrupt Request (IRQ) is indicated by the flag bit = 1 in an IFSx register. The IRQ causes an interrupt, if the corresponding bit in the Interrupt Enable (IECx) registers is set. For the rest of the instruction cycle in which the IRQ is sampled, the priorities of all pending interrupt requests are evaluated.

No instruction is aborted when the CPU responds to the IRQ. The instruction in progress when the IRQ is sampled is completed before the Interrupt Service Routine (ISR) is executed.

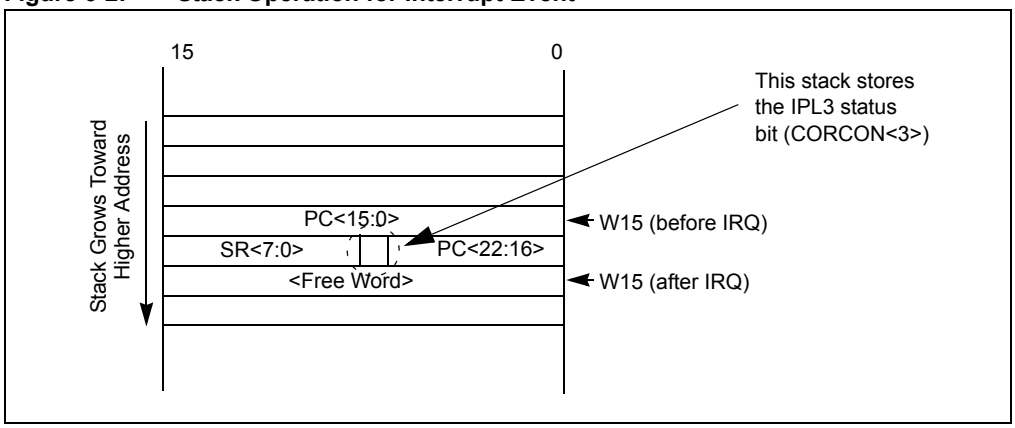
If there is a pending IRQ with a user-assigned priority level greater than the current processor priority level, indicated by the IPL<2:0> status bits (SR<7:5>), an interrupt is presented to the processor. The processor then saves the following information on the software stack:

- Current PC value
- Low byte of the Processor Status register (SRL)
- IPL3 status bit (CORCON<3>)

These three values allow the return Program Counter address value, MCU status bits and current processor priority level to be automatically saved.

After this information is saved on the stack, the CPU writes the priority level of the pending interrupt into the IPL<2:0> bit locations. This action disables all interrupts of lower or equal priority until the ISR is terminated using the RETFIE instruction.

Figure 6-2: Stack Operation for Interrupt Event



6.2.4.1 RETURN FROM INTERRUPT

The RETFIE (Return from Interrupt) instruction unstacks the PC return address, IPL3 status bit and SRL register to return the processor to the state and priority level that existed before the interrupt sequence.

6.2.4.2 INTERRUPT NESTING

Interrupts are nestable by default. Any ISR in progress can be interrupted by another source of interrupt with a higher user-assigned priority level. Interrupt nesting can be disabled by setting the Interrupt Nesting Disable (NSTDIS) control bit (INTCON1<15>). When the NSTDIS control bit is set, all interrupts in progress force the CPU priority to level 7 by setting IPL<2:0> = 111. This action effectively masks all other sources of interrupt until a RETFIE instruction is executed. When interrupt nesting is disabled, the user-assigned interrupt priority levels have no effect except to resolve conflicts between simultaneous pending interrupts.

The IPL<2:0> bits (SR<7:5>) become read-only when interrupt nesting is disabled. This prevents the user software from setting IPL<2:0> to a lower value, which would effectively re-enable interrupt nesting.

6.2.5 Wake-Up from Sleep and Idle

Any source of interrupt that is individually enabled, using its corresponding control bit in the IECx registers, can wake-up the processor from Sleep or Idle mode. When the interrupt status flag for a source is set and the interrupt source is enabled by the corresponding bit in the IEC Control registers, a wake-up signal is sent to the dsPIC33F/PIC24H CPU. When the device wakes from Sleep or Idle mode, one of two actions occur:

- If the interrupt priority level for that source is greater than the current CPU priority level, the processor will process the interrupt and branch to the ISR for the interrupt source.
- If the user-assigned interrupt priority level for the source is lower than or equal to the current CPU priority level, the processor will continue execution, starting with the instruction immediately following the PWRSAV instruction that previously put the CPU in Sleep or Idle mode.

Note:	User interrupt sources that are assigned to CPU priority level 0 cannot wake the CPU from Sleep or Idle mode, because the interrupt source is effectively disabled. To use an interrupt as a wake-up source, the program must assign the CPU priority level for the interrupt to level 1 or greater.

6.2.6 Analog-to-Digital Converter (ADC) External Conversion Request

The INT0 external interrupt request pin is shared with the ADC as an external conversion request signal. The INT0 interrupt source has programmable edge polarity, which is also available to the ADC external conversion request feature.

6.2.7 External Interrupt Support

The dsPIC33F/PIC24H supports up to five external interrupt pin sources (INT0-INT4). Each external interrupt pin has edge detection circuitry to detect the interrupt event. The INTCON2 register has five control bits (INT0EP-INT4EP) that select the polarity of the edge detection circuitry. Each external interrupt pin can be programmed to interrupt the CPU on a rising edge or falling edge event. See Register 6-4 for further details.

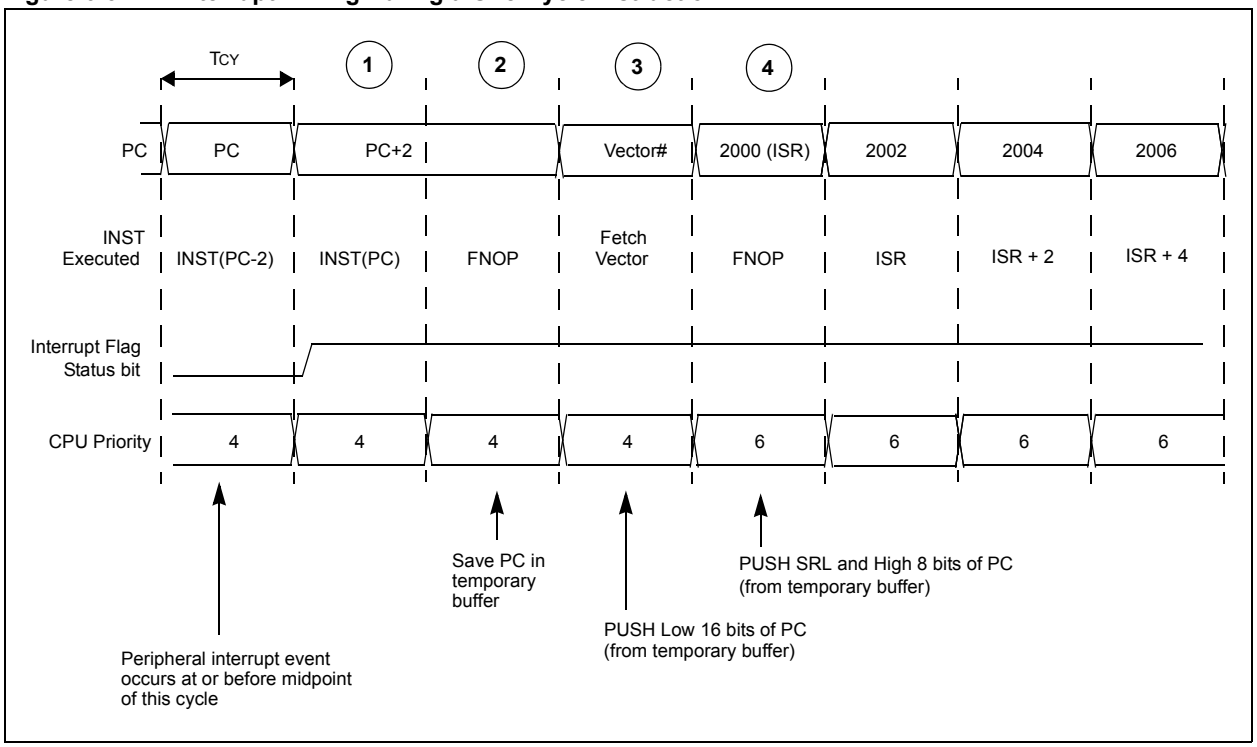
6.3 INTERRUPT PROCESSING TIMING

6.3.1 Interrupt Latency for One-Cycle Instructions

Figure 6-3 shows the sequence of events when a peripheral interrupt is asserted during a one-cycle instruction. The interrupt process takes four instruction cycles. Each cycle is numbered (in the figure) for reference.

The interrupt flag status bit is set during the instruction cycle after the peripheral interrupt occurs. The current instruction completes during this instruction cycle. In the second instruction cycle after the interrupt event, the contents of the PC and Lower Byte Status (SRL) registers are saved into a temporary buffer register. The second cycle of the interrupt process is executed as a `NOP` to maintain consistency with the sequence taken during a two-cycle instruction (see **6.3.2 "Interrupt Latency for Two-Cycle Instructions"**). In the third cycle, the PC is loaded with the vector table address for the interrupt source and the starting address of the ISR is fetched. In the fourth cycle, the PC is loaded with the ISR address. The fourth cycle is executed as a `NOP` while the first instruction, in the ISR is fetched.

Figure 6-3: Interrupt Timing During a One-Cycle Instruction



6.3.2 Interrupt Latency for Two-Cycle Instructions

The interrupt latency during a two-cycle instruction is the same as during a one-cycle instruction. The first and second cycle of the interrupt process allow the two-cycle instruction to complete execution. The timing diagram in Figure 6-4 shows the peripheral interrupt event occurring in the instruction cycle prior to execution of the two-cycle instruction.

Figure 6-5 shows the timing when a peripheral interrupt coincides with the first cycle of a two-cycle instruction. In this case, the interrupt process completes as for a one-cycle instruction (see **6.3.1 "Interrupt Latency for One-Cycle Instructions"**).

Figure 6-4: Interrupt Timing During a Two-Cycle Instruction

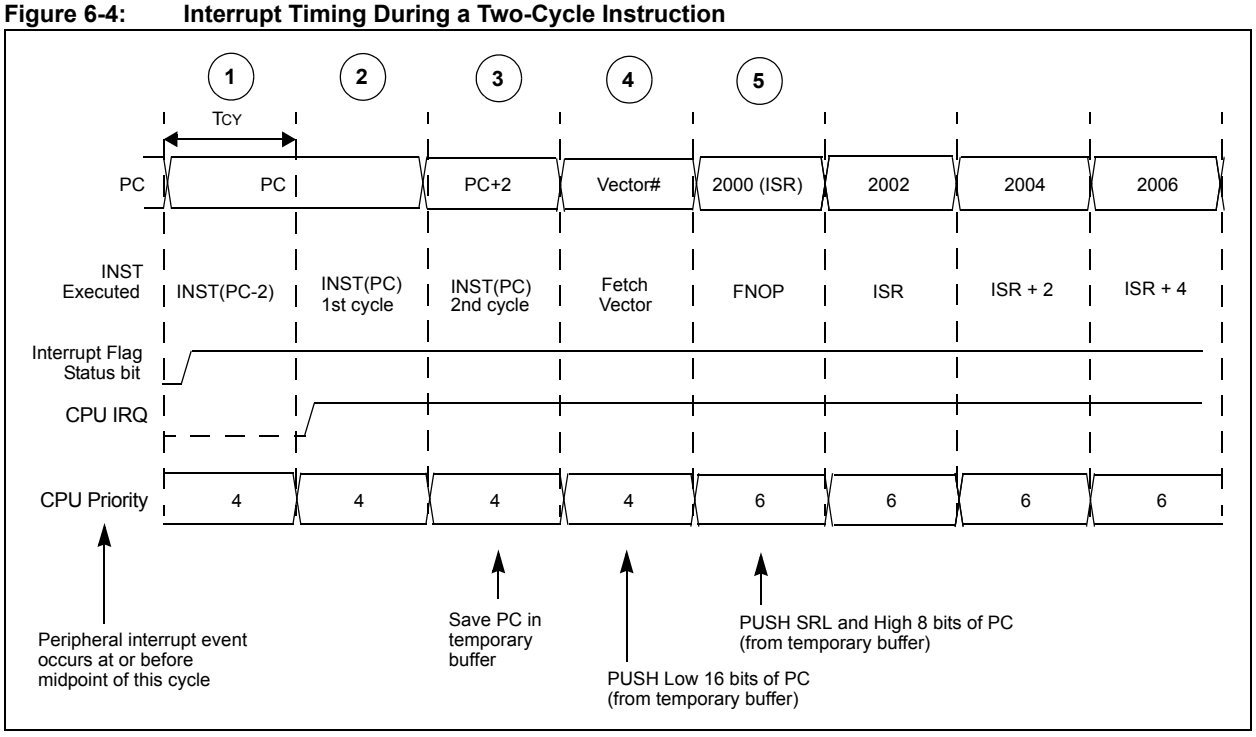
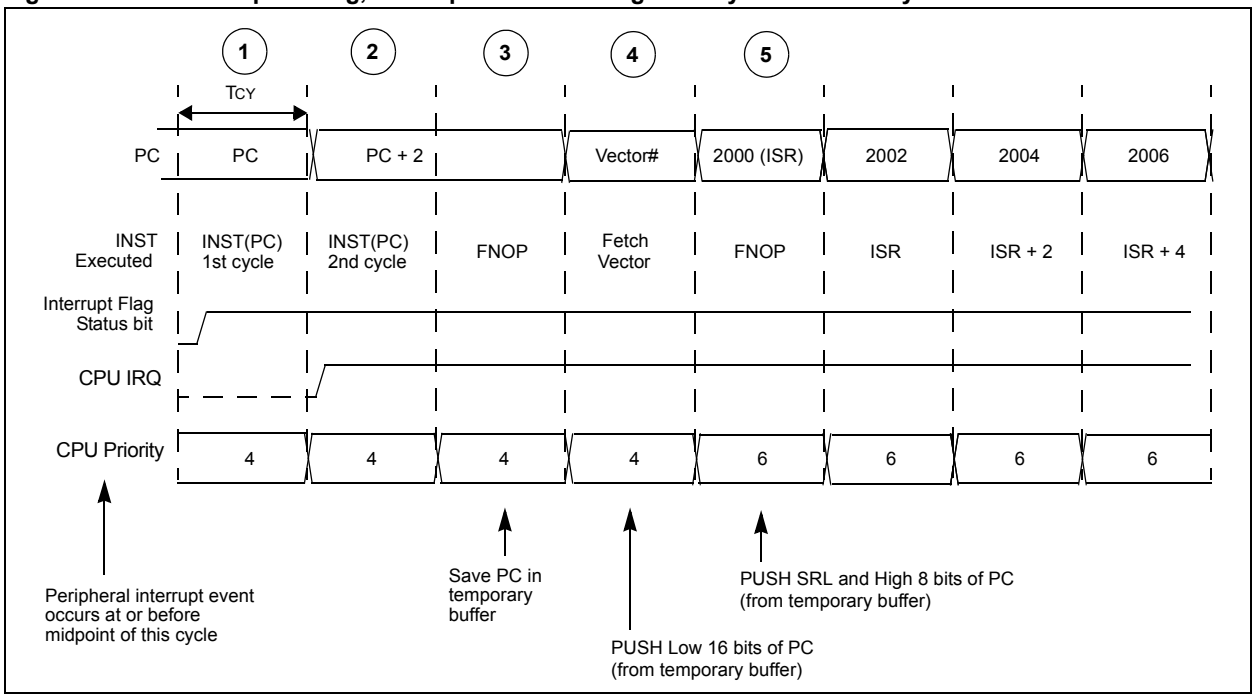


Figure 6-5: Interrupt Timing, Interrupt Occurs During First Cycle of a Two-Cycle Instruction

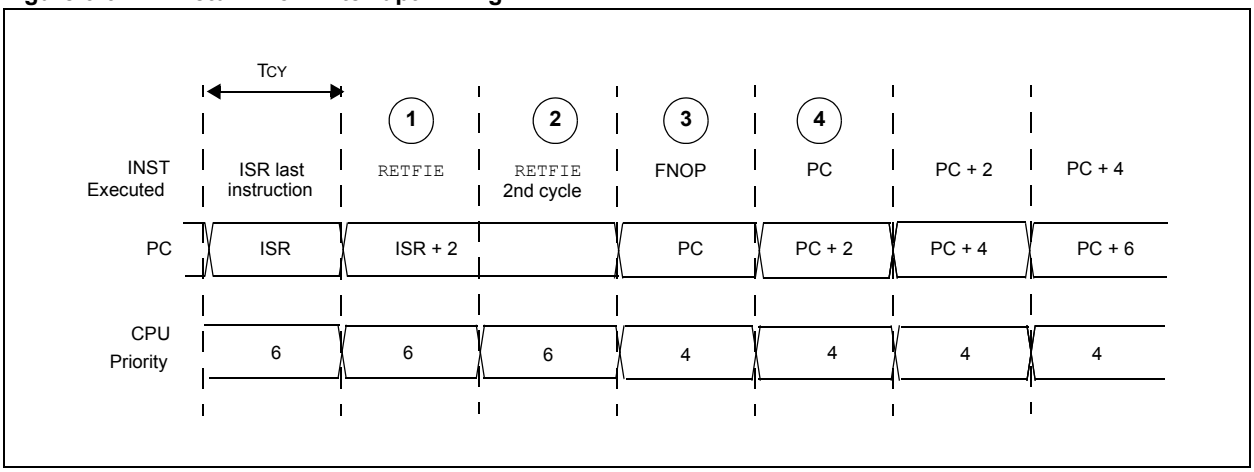


6.3.3 Returning from Interrupt

To return from an interrupt, the program must call the RETFIE instruction.

During the first two cycles of a RETFIE instruction, the contents of the PC and the SRL register are popped from the stack. The third instruction cycle is used to fetch the instruction addressed by the updated program counter. This cycle executes as a NOP instruction. On the fourth cycle, program execution resumes at the point where the interrupt occurred.

Figure 6-6: Return from Interrupt Timing



6.3.4 Special Conditions for Interrupt Latency

The dsPIC33F/PIC24H devices allow the current instruction to complete when a peripheral interrupt source becomes pending. The interrupt latency is the same for both one- and two-cycle instructions. However, certain conditions can increase interrupt latency by one cycle, depending on when the interrupt occurs. If a fixed latency is critical to the application, the following conditions should be avoided:

- Executing a MOV.D instruction uses PSV to access a value in program memory space
- Appending an instruction stall cycle to any two-cycle instruction
- Appending an instruction stall cycle to any one-cycle instruction that performs a PSV access
- A bit test and skip instruction (BTSC, BTSS) that uses PSV to access a value in the program memory space

6.4 INTERRUPT CONTROL AND STATUS REGISTERS

The following registers are associated with the interrupt controller:

- **INTCON1: Interrupt Control Register 1**

  This register contains the Interrupt Nesting Disable (NSTDIS) bit, as well as the control and status flags for the processor trap sources.

- **INTCON2: Interrupt Control Register 2**

  This register controls external interrupt request signal behavior and use of the alternate vector table.

- **IFSx: Interrupt Flag Status Registers** (Register 6-5 through Register 6-9)

  All interrupt request flags are maintained in the IFSx registers, where 'x' denotes the register number. Each source of interrupt has a status bit, which is set by the respective peripherals or external signal and cleared by software.

- **IECx: Interrupt Enable Control Registers** (Register 6-10 through Register 6-14)

  All Interrupt Enable Control bits are maintained in the IECx registers, where 'x' denotes the register number. These control bits are used to individually enable interrupts from the peripherals or external signals.

- **IPCx: Interrupt Priority Control Registers** (Register 6-15 through Register 6-32)

  Each user interrupt source can be assigned to one of eight priority levels. The IPC registers set the interrupt priority level for each source of interrupt.

- **INTTREG: Interrupt Control and Status Register**

  This register contains the associated interrupt vector number and the new CPU interrupt priority level, which are latched into the Vector Number (VECNUM<6:0>) and Interrupt Level (ILR<3:0>) bit fields. The new interrupt priority level is the priority of the pending interrupt.

- **SR: Status Register**

  This register is not specifically part of the interrupt controller hardware, but it contains the IPL<2:0> status bits (SR<7:5>) that indicate the current CPU priority level. The user application can change the current CPU priority level by writing to the IPL bits.

- **CORCON: Core Control Register**

  This register is not specifically part of the interrupt controller hardware, but it contains the IPL3 status bit, which indicates the current CPU priority level. IPL3 is a read-only bit so that trap events cannot be masked by the user software.

Each register is described in detail in the following sections.

Note:	The total number and type of interrupt sources depend on the particular device. Refer to the specific device data sheet for further details.

6.4.1 Assignment of Interrupts to Control Registers

The interrupt sources are assigned to the IFSx, IECx and IPCx registers in the same sequence that they are listed in Table 6-1. For example, the INT0 (External Interrupt 0) source has vector number and natural order priority 0. Therefore, the External Interrupt 0 Flag Status (INT0IF) bit exists in IFS0<0>. The INT0 interrupt uses bit 0 of the IEC0 register as its Enable bit. The IPC0<2:0> bits assign the interrupt priority level for the INT0 interrupt.

Register 6-1: SR: Status Register

R-0	R-0	R/C-0	R/C-0	R-0	R/C-0	R-0	R-0
OA	OB	SA	SB	OAB	SAB	DA	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
IPL<2:0>			RA	N	OV	Z	C
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-8	**Not used by the Interrupt Controller** (See the "*dsPIC30F/33F Programmer's Reference Manual*" (DS70157) for descriptions of SR bits)
bit 7-5	**IPL<2:0>:** CPU Interrupt Priority Level Status bits**(1,2)** 111 = CPU interrupt priority level is 7 (15). User interrupts disabled 110 = CPU interrupt priority level is 6 (14) 101 = CPU interrupt priority level is 5 (13) 100 = CPU interrupt priority level is 4 (12) 011 = CPU interrupt priority level is 3 (11) 010 = CPU interrupt priority level is 2 (10) 001 = CPU interrupt priority level is 1 (9) 000 = CPU interrupt priority level is 0 (8)
bit 4-0	**Not used by the Interrupt Controller** (See the "*dsPIC30F/33F Programmer's Reference Manual*" (DS70157) for descriptions of SR bits)

Note 1: The IPL<2:0> bits are concatenated with the IPL<3> bit (CORCON<3>) to form the CPU interrupt priority level. The value in parentheses indicates the IPL if IPL<3> = 1.

2: The IPL<2:0> status bits are read-only when NSTDIS = 1 (INTCON1<15>).

Register 6-2: CORCON: Core Control Register

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

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(1)	PSV	RND	IF
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-4	**Not used by the Interrupt Controller** (See "*dsPIC30F/33F Programmer's Reference Manual*" (DS70157) for descriptions of CORCON bits)
bit 3	**IPL3:** CPU Interrupt Priority Level Status bit 3(1) 1 = CPU interrupt priority level is greater than 7 0 = CPU interrupt priority level is 7 or less
bit 2-0	**Not used by the Interrupt Controller** (See "*dsPIC30F/33F Programmer's Reference Manual*" (DS70157) for descriptions of CORCON bits)

Note 1: The IPL3 bit is concatenated with the IPL<2:0> bits (SR<7:5>) to form the CPU interrupt priority level.

Register 6-3: INTCON1: Interrupt Control Register 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
NSTDIS	OVAERR	OVBERR	COVAERR	COVBERR	OVATE	OVBTE	COVTE
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	U-0
SFTACERR	DIV0ERR	DMACERR	MATHERR	ADDRERR	STKERR	OSCFAIL	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **NSTDIS:** Interrupt Nesting Disable bit
1 = Interrupt nesting is disabled
0 = Interrupt nesting is enabled

bit 14 **OVAERR:** Accumulator A Overflow Trap Flag bit
1 = Trap was caused by overflow of Accumulator A
0 = Trap was not caused by overflow of Accumulator A

bit 13 **OVBERR:** Accumulator B Overflow Trap Flag bit
1 = Trap was caused by overflow of Accumulator B
0 = Trap was not caused by overflow of Accumulator B

bit 12 **COVAERR:** Accumulator A Catastrophic Overflow Trap Flag bit
1 = Trap was caused by catastrophic overflow of Accumulator A
0 = Trap was not caused by catastrophic overflow of Accumulator A

bit 11 **COVBERR:** Accumulator B Catastrophic Overflow Trap Flag bit
1 = Trap was caused by catastrophic overflow of Accumulator B
0 = Trap was not caused by catastrophic overflow of Accumulator B

bit 10 **OVATE:** Accumulator A Overflow Trap Enable bit
1 = Trap overflow of Accumulator A
0 = Trap disabled

bit 9 **OVBTE:** Accumulator B Overflow Trap Enable bit
1 = Trap overflow of Accumulator B
0 = Trap disabled

bit 8 **COVTE:** Catastrophic Overflow Trap Enable bit
1 = Trap on catastrophic overflow of Accumulator A or B enabled
0 = Trap disabled

bit 7 **SFTACERR:** Shift Accumulator Error Status bit
1 = Math error trap was caused by an invalid accumulator shift
0 = Math error trap was not caused by an invalid accumulator shift

bit 6 **DIV0ERR:** Divide-by-zero Error Status bit
1 = Divide-by-zero error trap was caused by a divide by zero
0 = Divide-by-zero error trap was not caused by a divide by zero

bit 5 **DMACERR:** DMAC Error Status bit
1 = DMAC trap has occurred
0 = DMAC trap has not occurred

bit 4 **MATHERR:** Math Error Status bit
1 = Math error trap has occurred
0 = Math error trap has not occurred

bit 3 **ADDRERR:** Address Error Trap Status bit
1 = Address error trap has occurred
0 = Address error trap has not occurred

Register 6-3: INTCON1: Interrupt Control Register 1 (Continued)

bit 2 **STKERR:** Stack Error Trap Status bit
1 = Stack error trap has occurred
0 = Stack error trap has not occurred

bit 1 **OSCFAIL:** Oscillator Failure Trap Status bit
1 = Oscillator failure trap has occurred
0 = Oscillator failure trap has not occurred

bit 0 **Unimplemented:** Read as '0'

Register 6-4: INTCON2: Interrupt Control Register 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

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **ALTIVT:** Enable Alternate Interrupt Vector Table bit
1 = Use alternate vector table
0 = Use standard (default) vector table

bit 14 **DISI:** DISI Instruction Status bit
1 = DISI instruction is active
0 = DISI instruction is not active

bit 13-5 **Unimplemented:** Read as '0'

bit 4 **INT4EP:** External Interrupt 4 Edge Detect Polarity Select bit
1 = Interrupt on negative edge
0 = Interrupt on positive edge

bit 3 **INT3EP:** External Interrupt 3 Edge Detect Polarity Select bit
1 = Interrupt on negative edge
0 = Interrupt on positive edge

bit 2 **INT2EP:** External Interrupt 2 Edge Detect Polarity Select bit
1 = Interrupt on negative edge
0 = Interrupt on positive edge

bit 1 **INT1EP:** External Interrupt 1 Edge Detect Polarity Select bit
1 = Interrupt on negative edge
0 = Interrupt on positive edge

bit 0 **INT0EP:** External Interrupt 0 Edge Detect Polarity Select bit
1 = Interrupt on negative edge
0 = Interrupt on positive edge

Register 6-5: IFS0: Interrupt Flag Status Register 0

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	DMA1IF	AD1IF	U1TXIF	U1RXIF	SPI1IF	SPI1EIF	T3IF
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
T2IF	OC2IF	IC2IF	DMA0IF	T1IF	OC1IF	IC1IF	INT0IF
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **Unimplemented:** Read as '0'

bit 14 **DMA1IF:** DMA Channel 1 Data Transfer Complete Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 13 **AD1IF:** ADC1 Conversion Complete Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 12 **U1TXIF:** UART1 Transmitter Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 11 **U1RXIF:** UART1 Receiver Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 10 **SPI1IF:** SPI1 Event Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 9 **SPI1EIF:** SPI1 Fault Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 8 **T3IF:** Timer3 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 7 **T2IF:** Timer2 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 6 **OC2IF:** Output Compare Channel 2 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 5 **IC2IF:** Input Capture Channel 2 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 4 **DMA0IF:** DMA Channel 0 Data Transfer Complete Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 3 **T1IF:** Timer1 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

Register 6-5: IFS0: Interrupt Flag Status Register 0 (Continued)

bit 2 **OC1IF:** Output Compare Channel 1 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 1 **IC1IF:** Input Capture Channel 1 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 0 **INT0IF:** External Interrupt 0 Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

Register 6-6: IFS1: Interrupt Flag Status Register 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
U2TXIF	U2RXIF	INT2IF	T5IF	T4IF	OC4IF	OC3IF	DMA2IF
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
IC8IF	IC7IF	AD2IF	INT1IF	CNIF	—	MI2C1IF	SI2C1IF
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **U2TXIF:** UART2 Transmitter Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 14 **U2RXIF:** UART2 Receiver Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 13 **INT2IF:** External Interrupt 2 Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 12 **T5IF:** Timer5 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 11 **T4IF:** Timer4 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 10 **OC4IF:** Output Compare Channel 4 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 9 **OC3IF:** Output Compare Channel 3 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 8 **DMA2IF:** DMA Channel 2 Data Transfer Complete Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 7 **IC8IF:** Input Capture Channel 8 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 6 **IC7IF:** Input Capture Channel 7 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 5 **AD2IF:** ADC2 Conversion Complete Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 4 **INT1IF:** External Interrupt 1 Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

Register 6-6: IFS1: Interrupt Flag Status Register 1 (Continued)

bit 3 **CNIF:** Input Change Notification Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 2 **Unimplemented:** Read as '0'

bit 1 **MI2C1IF:** I2C1 Master Events Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 0 **SI2C1IF:** I2C1 Slave Events Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

Register 6-7: IFS2: Interrupt Flag Status Register 2

R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
T6IF	DMA4IF	—	OC8IF	OC7IF	OC6IF	OC5IF	IC6IF
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
IC5IF	IC4IF	IC3IF	DMA3IF	C1IF	C1RXIF	SPI2IF	SPI2EIF
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **T6IF:** Timer6 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 14 **DMA4IF:** DMA Channel 4 Data Transfer Complete Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 13 **Unimplemented:** Read as '0'

bit 12 **OC8IF:** Output Compare Channel 8 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 11 **OC7IF:** Output Compare Channel 7 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 10 **OC6IF:** Output Compare Channel 6 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 9 **OC5IF:** Output Compare Channel 5 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 8 **IC6IF:** Input Capture Channel 6 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 7 **IC5IF:** Input Capture Channel 5 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 6 **IC4IF:** Input Capture Channel 4 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 5 **IC3IF:** Input Capture Channel 3 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 4 **DMA3IF:** DMA Channel 3 Data Transfer Complete Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 3 **C1IF:** ECAN1 Event Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

Register 6-7: IFS2: Interrupt Flag Status Register 2 (Continued)

bit 2 **C1RXIF:** ECAN1 Receive Data Ready Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 1 **SPI2IF:** SPI2 Event Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 0 **SPI2EIF:** SPI2 Error Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

Register 6-8: IFS3: Interrupt Flag Status Register 3

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
FLTAIF	—	DMA5IF	DCIIF	DCIEIF	QEIIF	PWMIF	C2IF
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
C2RXIF	INT4IF	INT3IF	T9IF	T8IF	MI2C2IF	SI2C2IF	T7IF
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **FLTAIF:** PWM Fault A Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 14 **Unimplemented:** Read as '0'

bit 13 **DMA5IF:** DMA Channel 5 Data Transfer Complete Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 12 **DCIIF:** DCI Event Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 11 **DCIEIF:** DCI Error Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 10 **QEIIF:** QEI Event Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 9 **PWMIF:** PWM Error Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 8 **C2IF:** ECAN2 Event Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 7 **C2RXIF:** ECAN2 Receive Data Ready Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 6 **INT4IF:** External Interrupt 4 Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 5 **INT3IF:** External Interrupt 3 Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 4 **T9IF:** Timer9 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 3 **T8IF:** Timer8 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

Register 6-8: IFS3: Interrupt Flag Status Register 3 (Continued)

bit 2 **MI2C2IF:** I2C2 Master Events Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 1 **SI2C2IF:** I2C2 Slave Events Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 0 **T7IF:** Timer7 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

Register 6-9: IFS4: Interrupt Flag Status Register 4

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

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
C2TXIF	C1TXIF	DMA7IF	DMA6IF	—	U2EIF	U1EIF	FLTBIF
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-8 **Unimplemented:** Read as '0'

bit 7 **C2TXIF:** ECAN2 Transmit Data Request Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 6 **C1TXIF:** ECAN1 Transmit Data Request Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 5 **DMA7IF:** DMA Channel 7 Data Transfer Complete Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 4 **DMA6IF:** DMA Channel 6 Data Transfer Complete Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 3 **Unimplemented:** Read as '0'

bit 2 **U2EIF:** UART2 Error Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 1 **U1EIF:** UART1 Error Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 0 **FLTBIF:** PWM Fault B Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

Register 6-10: IEC0: Interrupt Enable Control Register 0

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	DMA1IE	AD1IE	U1TXIE	U1RXIE	SPI1IE	SPI1EIE	T3IE
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
T2IE	OC2IE	IC2IE	DMA0IE	T1IE	OC1IE	IC1IE	INT0IE
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **Unimplemented:** Read as '0'

bit 14 **DMA1IE:** DMA Channel 1 Data Transfer Complete Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 13 **AD1IE:** ADC1 Conversion Complete Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 12 **U1TXIE:** UART1 Transmitter Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 11 **U1RXIE:** UART1 Receiver Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 10 **SPI1IE:** SPI1 Event Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 9 **SPI1EIE:** SPI1 Error Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 8 **T3IE:** Timer3 Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 7 **T2IE:** Timer2 Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 6 **OC2IE:** Output Compare Channel 2 Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 5 **IC2IE:** Input Capture Channel 2 Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 4 **DMA0IE:** DMA Channel 0 Data Transfer Complete Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 3 **T1IE:** Timer1 Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

Register 6-10: IEC0: Interrupt Enable Control Register 0 (Continued)

bit 2 **OC1IE:** Output Compare Channel 1 Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 1 **IC1IE:** Input Capture Channel 1 Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 0 **INT0IE:** External Interrupt 0 Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

Register 6-11: IEC1: Interrupt Enable Control Register 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
U2TXIE	U2RXIE	INT2IE	T5IE	T4IE	OC4IE	OC3IE	DMA2IE
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
IC8IE	IC7IE	AD2IE	INT1IE	CNIE	—	MI2C1IE	SI2C1IE
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **U2TXIE:** UART2 Transmitter Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 14 **U2RXIE:** UART2 Receiver Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 13 **INT2IE:** External Interrupt 2 Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 12 **T5IE:** Timer5 Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 11 **T4IE:** Timer4 Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 10 **OC4IE:** Output Compare Channel 4 Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 9 **OC3IE:** Output Compare Channel 3 Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 8 **DMA2IE:** DMA Channel 2 Data Transfer Complete Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 7 **IC8IE:** Input Capture Channel 8 Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 6 **IC7IE:** Input Capture Channel 7 Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 5 **AD2IE:** ADC2 Conversion Complete Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 4 **INT1IE:** External Interrupt 1 Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

Register 6-11: IEC1: Interrupt Enable Control Register 1 (Continued)

bit 3 **CNIE:** Input Change Notification Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 2 **Unimplemented:** Read as '0'

bit 1 **MI2C1IE:** I2C1 Master Events Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 0 **SI2C1IE:** I2C1 Slave Events Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

Register 6-12: IEC2: Interrupt Enable Control Register 2

R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
T6IE	DMA4IE	—	OC8IE	OC7IE	OC6IE	OC5IE	IC6IE
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
IC5IE	IC4IE	IC3IE	DMA3IE	C1IE	C1RXIE	SPI2IE	SPI2EIE
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **T6IE:** Timer6 Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 14 **DMA4IE:** DMA Channel 4 Data Transfer Complete Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 13 **Unimplemented:** Read as '0'

bit 12 **OC8IE:** Output Compare Channel 8 Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 11 **OC7IE:** Output Compare Channel 7 Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 10 **OC6IE:** Output Compare Channel 6 Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 9 **OC5IE:** Output Compare Channel 5 Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 8 **IC6IE:** Input Capture Channel 6 Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 7 **IC5IE:** Input Capture Channel 5 Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 6 **IC4IE:** Input Capture Channel 4 Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 5 **IC3IE:** Input Capture Channel 3 Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 4 **DMA3IE:** DMA Channel 3 Data Transfer Complete Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 3 **C1IE:** ECAN1 Event Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

Register 6-12: IEC2: Interrupt Enable Control Register 2 (Continued)

bit 2 **C1RXIE:** ECAN1 Receive Data Ready Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 1 **SPI2IE:** SPI2 Event Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 0 **SPI2EIE:** SPI2 Error Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

Register 6-13: IEC3: Interrupt Enable Control Register 3

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
FLTAIE	—	DMA5IE	DCIIE	DCIEIE	QEIIE	PWMIE	C2IE
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
C2RXIE	INT4IE	INT3IE	T9IE	T8IE	MI2C2IE	SI2C2IE	T7IE
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **FLTAIE:** PWM Fault A Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 14 **Unimplemented:** Read as '0'

bit 13 **DMA5IE:** DMA Channel 5 Data Transfer Complete Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 12 **DCIIE:** DCI Event Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 11 **DCIEIE:** DCI Error Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 10 **QEIIE:** QEI Event Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 9 **PWMIE:** PWM Error Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 8 **C2IE:** ECAN2 Event Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 7 **C2RXIE:** ECAN2 Receive Data Ready Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 6 **INT4IE:** External Interrupt 4 Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 5 **INT3IE:** External Interrupt 3 Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 4 **T9IE:** Timer9 Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 3 **T8IE:** Timer8 Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

Register 6-13: IEC3: Interrupt Enable Control Register 3 (Continued)

bit 2 **MI2C2IE:** I2C2 Master Events Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 1 **SI2C2IE:** I2C2 Slave Events Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 0 **T7IE:** Timer7 Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

Register 6-14: IEC4: Interrupt Enable Control Register 4

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

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
C2TXIE	C1TXIE	DMA7IE	DMA6IE	—	U2EIE	U1EIE	FLTBIE
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-8 **Unimplemented:** Read as '0'

bit 7 **C2TXIE:** ECAN2 Transmit Data Request Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 6 **C1TXIE:** ECAN1 Transmit Data Request Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 5 **DMA7IE:** DMA Channel 7 Data Transfer Complete Enable Status bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 4 **DMA6IE:** DMA Channel 6 Data Transfer Complete Enable Status bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 3 **Unimplemented:** Read as '0'

bit 2 **U2EIE:** UART2 Error Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 1 **U1EIE:** UART1 Error Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 0 **FLTBIE:** PWM Fault B Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

Register 6-15: IPC0: Interrupt Priority Control Register 0

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

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

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **Unimplemented:** Read as '0'

bit 14-12 **T1IP<2:0>:** Timer1 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 11 **Unimplemented:** Read as '0'

bit 10-8 **OC1IP<2:0>:** Output Compare Channel 1 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 7 **Unimplemented:** Read as '0'

bit 6-4 **IC1IP<2:0>:** Input Capture Channel 1 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 3 **Unimplemented:** Read as '0'

bit 2-0 **INT0IP<2:0>:** External Interrupt 0 Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

Register 6-16: IPC1: Interrupt Priority Control Register 1

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	T2IP<2:0>			—	OC2IP<2:0>		
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
—	IC2IP<2:0>			—	DMA0IP<2:0>		
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **Unimplemented:** Read as '0'

bit 14-12 **T2IP<2:0>:** Timer2 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 11 **Unimplemented:** Read as '0'

bit 10-8 **OC2IP<2:0>:** Output Compare Channel 2 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 7 **Unimplemented:** Read as '0'

bit 6-4 **IC2IP<2:0>:** Input Capture Channel 2 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 3 **Unimplemented:** Read as '0'

bit 2-0 **DMA0IP<2:0>:** DMA Channel 0 Data Transfer Complete Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

Register 6-17: IPC2: Interrupt Priority Control Register 2

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 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
—	SPI1EIP<2:0>			—	T3IP<2:0>		
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **Unimplemented:** Read as '0'

bit 14-12 **U1RXIP<2:0>:** UART1 Receiver Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 11 **Unimplemented:** Read as '0'

bit 10-8 **SPI1IP<2:0>:** SPI1 Event Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 7 **Unimplemented:** Read as '0'

bit 6-4 **SPI1EIP<2:0>:** SPI1 Error Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 3 **Unimplemented:** Read as '0'

bit 2-0 **T3IP<2:0>:** Timer3 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

Register 6-18: IPC3: Interrupt Priority Control Register 3

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	DMA1IP<2:0>		
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
—	AD1IP<2:0>			—	U1TXIP<2:0>		
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-11 **Unimplemented:** Read as '0'

bit 10-8 **DMA1IP<2:0>:** DMA Channel 1 Data Transfer Complete Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 7 **Unimplemented:** Read as '0'

bit 6-4 **AD1IP<2:0>:** ADC1 Conversion Complete Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 3 **Unimplemented:** Read as '0'

bit 2-0 **U1TXIP<2:0>:** UART1 Transmitter Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

Register 6-19: IPC4: Interrupt Priority Control Register 4

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

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	MI2C1IP<2:0>			—	SI2C1IP<2:0>		
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **Unimplemented:** Read as '0'

bit 14-12 **CNIP<2:0>:** Change Notification Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 11-7 **Unimplemented:** Read as '0'

bit 6-4 **MI2C1IP<2:0>:** I2C1 Master Events Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 3 **Unimplemented:** Read as '0'

bit 2-0 **SI2C1IP<2:0>:** I2C1 Slave Events Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

Register 6-20: IPC5: Interrupt Priority Control Register 5

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	IC8IP<2:0>			—	IC7IP<2:0>		
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
—	AD2IP<2:0>			—	INT1IP<2:0>		
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **Unimplemented:** Read as '0'

bit 14-12 **IC8IP<2:0>:** Input Capture Channel 8 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 11 **Unimplemented:** Read as '0'

bit 10-8 **IC7IP<2:0>:** Input Capture Channel 7 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 7 **Unimplemented:** Read as '0'

bit 6-4 **AD2IP<2:0>:** ADC2 Conversion Complete Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 3 **Unimplemented:** Read as '0'

bit 2-0 **INT1IP<2:0>:** External Interrupt 1 Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

Register 6-21: IPC6: Interrupt Priority Control Register 6

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	T4IP<2:0>			—	OC4IP<2:0>		
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
—	OC3IP<2:0>			—	DMA2IP<2:0>		
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **Unimplemented:** Read as '0'

bit 14-12 **T4IP<2:0>:** Timer4 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 11 **Unimplemented:** Read as '0'

bit 10-8 **OC4IP<2:0>:** Output Compare Channel 4 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 7 **Unimplemented:** Read as '0'

bit 6-4 **OC3IP<2:0>:** Output Compare Channel 3 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 3 **Unimplemented:** Read as '0'

bit 2-0 **DMA2IP<2:0>:** DMA Channel 2 Data Transfer Complete Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

Register 6-22: IPC7: Interrupt Priority Control Register 7

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	U2TXIP<2:0>			—	U2RXIP<2:0>		
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
—	INT2IP<2:0>			—	T5IP<2:0>		
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **Unimplemented:** Read as '0'

bit 14-12 **U2TXIP<2:0>:** UART2 Transmitter Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 11 **Unimplemented:** Read as '0'

bit 10-8 **U2RXIP<2:0>:** UART2 Receiver Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 7 **Unimplemented:** Read as '0'

bit 6-4 **INT2IP<2:0>:** External Interrupt 2 Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 3 **Unimplemented:** Read as '0'

bit 2-0 **T5IP<2:0>:** Timer5 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

Register 6-23: IPC8: Interrupt Priority Control Register 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	C1IP<2:0>			—	C1RXIP<2:0>		
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
—	SPI2IP<2:0>			—	SPI2EIP<2:0>		
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **Unimplemented:** Read as ‘0’

bit 14-12 **C1IP<2:0>:** ECAN1 Event Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 11 **Unimplemented:** Read as ‘0’

bit 10-8 **C1RXIP<2:0>:** ECAN1 Receive Data Ready Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 7 **Unimplemented:** Read as ‘0’

bit 6-4 **SPI2IP<2:0>:** SPI2 Event Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 3 **Unimplemented:** Read as ‘0’

bit 2-0 **SPI2EIP<2:0>:** SPI2 Error Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

Register 6-24: IPC9: Interrupt Priority Control Register 9

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	IC5IP<2:0>			—	IC4IP<2:0>		
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
—	IC3IP<2:0>			—	DMA3IP<2:0>		
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **Unimplemented:** Read as '0'

bit 14-12 **IC5IP<2:0>:** Input Capture Channel 5 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 11 **Unimplemented:** Read as '0'

bit 10-8 **IC4IP<2:0>:** Input Capture Channel 4 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 7 **Unimplemented:** Read as '0'

bit 6-4 **IC3IP<2:0>:** Input Capture Channel 3 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 3 **Unimplemented:** Read as '0'

bit 2-0 **DMA3IP<2:0>:** DMA Channel 3 Data Transfer Complete Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

Register 6-25: IPC10: Interrupt Priority Control Register 10

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	OC7IP<2:0>			—	OC6IP<2:0>		
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
—	OC5IP<2:0>			—	IC6IP<2:0>		
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **Unimplemented:** Read as '0'

bit 14-12 **OC7IP<2:0>:** Output Compare Channel 7 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 11 **Unimplemented:** Read as '0'

bit 10-8 **OC6IP<2:0>:** Output Compare Channel 6 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 7 **Unimplemented:** Read as '0'

bit 6-4 **OC5IP<2:0>:** Output Compare Channel 5 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 3 **Unimplemented:** Read as '0'

bit 2-0 **IC6IP<2:0>:** Input Capture Channel 6 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

Register 6-26: IPC11: Interrupt Priority Control Register 11

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	T6IP<2:0>			—	DMA4IP<2:0>		
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	OC8IP<2:0>		
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **Unimplemented:** Read as '0'

bit 14-12 **T6IP<2:0>:** Timer6 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 11 **Unimplemented:** Read as '0'

bit 10-8 **DMA4IP<2:0>:** DMA Channel 4 Data Transfer Complete Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 7-3 **Unimplemented:** Read as '0'

bit 2-0 **OC8IP<2:0>:** Output Compare Channel 8 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

Register 6-27: IPC12: Interrupt Priority Control Register 12

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	T8IP<2:0>			—	MI2C2IP<2:0>		
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
—	SI2C2IP<2:0>			—	T7IP<2:0>		
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **Unimplemented:** Read as '0'

bit 14-12 **T8IP<2:0>:** Timer8 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 11 **Unimplemented:** Read as '0'

bit 10-8 **MI2C2IP<2:0>:** I2C2 Master Events Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 7 **Unimplemented:** Read as '0'

bit 6-4 **SI2C2IP<2:0>:** I2C2 Slave Events Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 3 **Unimplemented:** Read as '0'

bit 2-0 **T7IP<2:0>:** Timer7 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

Register 6-28: IPC13: Interrupt Priority Control Register 13

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	C2RXIP<2:0>			—	INT4IP<2:0>		
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
—	INT3IP<2:0>			—	T9IP<2:0>		
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **Unimplemented:** Read as '0'

bit 14-12 **C2RXIP<2:0>:** ECAN2 Receive Data Ready Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 11 **Unimplemented:** Read as '0'

bit 10-8 **INT4IP<2:0>:** External Interrupt 4 Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 7 **Unimplemented:** Read as '0'

bit 6-4 **INT3IP<2:0>:** External Interrupt 3 Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 3 **Unimplemented:** Read as '0'

bit 2-0 **T9IP<2:0>:** Timer9 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

Register 6-29: IPC14: Interrupt Priority Control Register 14

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	DCIEIP<2:0>			—	QEIIP<2:0>		
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
—	PWMIP<2:0>			—	C2IP<2:0>		
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **Unimplemented:** Read as '0'

bit 14-12 **DCIEIP<2:0>:** DCI Error Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 11 **Unimplemented:** Read as '0'

bit 10-8 **QEIIP<2:0>:** QEI Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 7 **Unimplemented:** Read as '0'

bit 6-4 **PWMIP<2:0>:** PWM Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 3 **Unimplemented:** Read as '0'

bit 2-0 **C2IP<2:0>:** ECAN2 Event Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

Register 6-30: IPC15: Interrupt Priority Control Register 15

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	FLTAIP<2:0>			—	—	—	—
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
—	DMA5IP<2:0>			—	DCIIP<2:0>		
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **Unimplemented:** Read as '0'

bit 14-12 **FLTAIP<2:0>:** PWM Fault A Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 11-7 **Unimplemented:** Read as '0'

bit 6-4 **DMA5IP<2:0>:** DMA Channel 5 Data Transfer Complete Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 3 **Unimplemented:** Read as '0'

bit 2-0 **DCIIP<2:0>:** DCI Event Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

Register 6-31: IPC16: Interrupt Priority Control Register 16

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	U2EIP<2:0>		
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
—	U1EIP<2:0>			—	FLTBIP<2:0>		
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-11 **Unimplemented:** Read as '0'

bit 10-8 **U2EIP<2:0>:** UART2 Error Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 7 **Unimplemented:** Read as '0'

bit 6-4 **U1EIP<2:0>:** UART1 Error Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 3 **Unimplemented:** Read as '0'

bit 2-0 **FLTBIP<2:0>:** PWM Fault B Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

Register 6-32: IPC17: Interrupt Priority Control Register 17

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	C2TXIP<2:0>			—	C1TXIP<2:0>		
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
—	DMA7IP<2:0>			—	DMA6IP<2:0>		
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **Unimplemented:** Read as '0'

bit 14-12 **C2TXIP<2:0>:** ECAN2 Transmit Data Request Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 11 **Unimplemented:** Read as '0'

bit 10-8 **C1TXIP<2:0>:** ECAN1 Transmit Data Request Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 7 **Unimplemented:** Read as '0'

bit 6-4 **DMA7IP<2:0>:** DMA Channel 7 Data Transfer Complete Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 3 **Unimplemented:** Read as '0'

bit 2-0 **DMA6IP<2:0>:** DMA Channel 6 Data Transfer Complete Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

Register 6-33: INTTREG: Interrupt Control and Status Register

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

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

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-12 **Unimplemented:** Read as '0'

bit 11-8 **ILR<3:0>:** New CPU Interrupt Priority Level bits

1111 = CPU Interrupt Priority Level is 15
•
•
•
0001 = CPU Interrupt Priority Level is 1
0000 = CPU Interrupt Priority Level is 0

bit 7 **Unimplemented:** Read as '0'

bit 6-0 **VECNUM<6:0>:** Vector Number of Pending Interrupt bits

1111111 = Interrupt vector pending is number 135
•
•
•
0000001 = Interrupt vector pending is number 9
0000000 = Interrupt vector pending is number 8

6.5 INTERRUPT SETUP PROCEDURES

6.5.1 Initialization

To configure an interrupt source:

1. Set the NSTDIS control bit (INTCON1<15>), if you do not plan to use nested interrupts.
2. Select the user-assigned priority level for the interrupt source by writing the control bits in the appropriate IPCx Control register. The priority level depends on the specific application and type of interrupt source. If you do not plan to use multiple priority levels, you can program the IPCx register control bits for all enabled interrupt sources to the same non-zero value.

Note:	At a device Reset, the IPC registers are initialized with all user interrupt sources assigned to priority level 4.

3. Clear the interrupt flag status bit associated with the peripheral in the associated IFSx Status register.
4. Enable the interrupt source by setting the interrupt enable control bit associated with the source in the appropriate IECx Control register.

6.5.2 Interrupt Service Routine

The method used to declare an ISR and initialize the Interrupt Vector Table (IVT) with the correct vector address depends on the programming language (C or Assembler) and the language development tool suite used to develop the application. In general, the user application must clear the interrupt flag in the appropriate IFSx register for the source of interrupt that the ISR handles. Otherwise, the application will immediately re-enter the ISR after it exits the routine. If you code the ISR in Assembler language, it must be terminated using a `RETFIE` instruction to unstack the saved PC value, SRL value and old CPU priority level.

6.5.3 Trap Service Routine

A Trap Service Routine (TSR) is coded like an ISR, except that the code must clear the appropriate trap status flag in the INTCON1 register to avoid re-entry into the TSR.

6.5.4 Interrupt Disable

To disable interrupts:

1. Push the current SR value onto the software stack using the `PUSH` instruction.
2. Force the CPU to priority level 7 by inclusive ORing the value `0xE0` with SRL.

To enable user interrupts, you can use the `POP` instruction to restore the previous SR value.

Note:	Only user interrupts with a priority level of 7 or less can be disabled. Trap sources (level 8-level 15) cannot be disabled.

The `DISI` instruction disables interrupts of priority levels 1-6 for a fixed period of time. Level 7 interrupt sources are not disabled by the `DISI` instruction.

6.5.5 Code Example

Example 6-1 illustrates code that enables nested interrupts, sets up Timer1, Timer2, Timer3, Timer4, and change notice peripherals to priority levels 2, 5, 6, 3, and 4, respectively. It also illustrates how interrupts can be enabled and disabled using the Status register. Sample ISR illustrates interrupt clearing.

Example 6-1: Interrupt Setup Code Example

```
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 6 (level 7 is highest) */
    IPC2bits.T3IP = 6;

    /* Set Timer2 interrupt priority to 5 */
    IPC1bits.T2IP = 5;

    /* Set Change Notice interrupt priority to 4 */
    IPC4bits.CNIP = 4;

    /* Set Timer4 interrupt priority to 3 */
    IPC6bits.T4IP = 3;

    /* 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;

    /* Reset Timer4 interrupt flag */
    IFS1bits.T4IF = 0;

    /* Enable CN interrupts */
    IEC1bits.CNIE = 1;
```

Example 6-1: Interrupt Setup Code Example (Continued)

```
    /* Enable Timer1 interrupt */
    IEC0bits.T1IE = 1;

    /* Enable Timer2 interrupt (PWM time base) */
    IEC0bits.T2IE = 1;

    /* Enable Timer3 interrupt */
    IEC0bits.T3IE = 1;

    /* Enable Timer4 interrupt (replacement for Timer 2 */
    IEC1bits.T4IE = 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;
}

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__)) _T4Interrupt(void)
{
    /* Insert ISR Code Here*/

    /* Clear Timer4 interrupt */
    IFS1bits.T4IF = 0;
}

void __attribute__((__interrupt__)) _CNInterrupt(void)
{
    /* Insert ISR Code Here*/

    /* Clear CN interrupt */
    IFS1bits.CNIF = 0;
}
```

Table 6-2: Interrupt Controller Register Map

File Name	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	All Resets
INTCON1	NSTDIS	OVAERR	OVBERR	COVAERR	COVBERR	OVATE	OVBTE	COVTE	SFTACERR	DIV0ERR	DMACERR	MATHERR	ADDRERR	STKERR	OSCFAIL	—	0000
INTCON2	ALTIVT	DISI	—	—	—	—	—	—	—	—	—	INT4EP	INT3EP	INT2EP	INT1EP	INT0EP	0000
IFS0	—	DMA1IF	AD1IF	U1TXIF	U1RXIF	SPI1IF	SPI1EIF	T3IF	T2IF	OC2IF	IC2IF	DMA0IF	T1IF	OC1IF	IC1IF	INT0IF	0000
IFS1	U2TXIF	U2RXIF	INT2IF	T5IF	T4IF	OC4IF	OC3IF	DMA2IF	IC8IF	IC7IF	AD2IF	INT1IF	CNIF	—	MI2C1IF	SI2C1IF	0000
IFS2	T6IF	DMA4IF	—	OC8IF	OC7IF	OC6IF	OC5IF	IC6IF	IC5IF	IC4IF	IC3IF	DMA3IF	C1IF	C1RXIF	SPI2IF	SPI2EIF	0000
IFS3	FLTAIF	—	DMA5IF	DCIIF	DCIEIF	QEIIF	PWMIF	C2IF	C2RXIF	INT4IF	INT3IF	T9IF	T8IF	MI2C2IF	SI2C2IF	T7IF	0000
IFS4	—	—	—	—	—	—	—	—	C2TXIF	C1TXIF	DMA7IF	DMA6IF	—	U2EIF	U1EIF	FLTBIF	0000
IEC0	—	DMA1IE	AD1IE	U1TXIE	U1RXIE	SPI1IE	SPI1EIE	T3IE	T2IE	OC2IE	IC2IE	DMA0IE	T1IE	OC1IE	IC1IE	INT0IE	0000
IEC1	U2TXIE	U2RXIE	INT2IE	T5IE	T4IE	OC4IE	OC3IE	DMA2IE	IC8IE	IC7IE	AD2IE	INT1IE	CNIE	—	MI2C1IE	SI2C1IE	0000
IEC2	T6IE	DMA4IE	—	OC8IE	OC7IE	OC6IE	OC5IE	IC6IE	IC5IE	IC4IE	IC3IE	DMA3IE	C1IE	C1RXIE	SPI2IE	SPI2EIE	0000
IEC3	FLTAIE	—	DMA5IE	DCIIE	DCIEIE	QEIIE	PWMIE	C2IE	C2RXIE	INT4IE	INT3IE	T9IE	T8IE	MI2C2IE	SI2C2IE	T7IE	0000
IEC4	—	—	—	—	—	—	—	—	C2TXIE	C1TXIE	DMA7IE	DMA6IE	—	U2EIE	U1EIE	FLTBIE	0000
IPC0	—	T1IP<2:0>			—	OC1IP<2:0>			—	IC1IP<2:0>			—	INT0IP<2:0>			4444
IPC1	—	T2IP<2:0>			—	OC2IP<2:0>			—	IC2IP<2:0>			—	DMA0IP<2:0>			4444
IPC2	—	U1RXIP<2:0>			—	SPI1IP<2:0>			—	SPI1EIP<2:0>			—	T3IP<2:0>			4444
IPC3	—	—	—	—	—	DMA1IP<2:0>			—	AD1IP<2:0>			—	U1TXIP<2:0>			4444
IPC4	—	CNIP<2:0>			—	—	—	—	—	MI2C1IP<2:0>			—	SI2C1IP<2:0>			4444
IPC5	—	IC8IP<2:0>			—	IC7IP<2:0>			—	AD2IP<2:0>			—	INT1IP<2:0>			4444
IPC6	—	T4IP<2:0>			—	OC4IP<2:0>			—	OC3IP<2:0>			—	DMA2IP<2:0>			4444
IPC7	—	U2TXIP<2:0>			—	U2RXIP<2:0>			—	INT2IP<2:0>			—	T5IP<2:0>			4444
IPC8	—	C1IP<2:0>			—	C1RXIP<2:0>			—	SPI2IP<2:0>			—	SPI2EIP<2:0>			4444
IPC9	—	IC5IP<2:0>			—	IC4IP<2:0>			—	IC3IP<2:0>			—	DMA3IP<2:0>			4444
IPC10	—	OC7IP<2:0>			—	OC6IP<2:0>			—	OC5IP<2:0>			—	IC6IP<2:0>			4444
IPC11	—	T6IP<2:0>			—	DMA4IP<2:0>			—	—	—	—	—	OC8IP<2:0>			4444
IPC12	—	T8IP<2:0>			—	MI2C2IP<2:0>			—	SI2C2IP<2:0>			—	T7IP<2:0>			4444
IPC13	—	C2RXIP<2:0>			—	INT4IP<2:0>			—	INT3IP<2:0>			—	T9IP<2:0>			4444
IPC14	—	DCIEIP<2:0>			—	QEIIP<2:0>			—	PWMIP<2:0>			—	C2IP<2:0>			4444
IPC15	—	FLTAIP<2:0>			—	—	—	—	—	DMA5IP<2:0>			—	DCIIP<2:0>			4444
IPC16	—	—	—	—	—	U2EIP<2:0>			—	U1EIP<2:0>			—	FLTBIP<2:0>			4444
IPC17	—	C2TXIP<2:0>			—	C1TXIP<2:0>			—	DMA7IP<2:0>			—	DMA6IP<2:0>			4444

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

6.6 DESIGN TIPS

Question 1: ***What happens when two sources of interrupt become pending at the same time and have the same user-assigned priority level?***

Answer: The interrupt source with the highest natural order priority will take precedence. The natural order priority is determined by the Interrupt Vector Table (IVT) address for that source. Interrupt sources with a lower IVT address have a higher natural order priority.

Question 2: ***Can the `DISI` instruction be used to disable all sources of interrupt and traps?***

Answer: The `DISI` instruction does not disable traps or priority level 7 interrupt sources. However, the `DISI` instruction can be used as a convenient way to disable all interrupt sources, if no priority level 7 interrupt sources are enabled in the user's application.

Question 3: ***What happens when a peripheral interrupt is used as a DMA request?***

Answer: The user application can designate any peripheral interrupt to be a DMA request. A DMA request is an IRQ that is directed to the DMA. When the DMA channel is configured to respond to a particular interrupt as a DMA request, the application should disable the corresponding CPU interrupt. Otherwise, a CPU interrupt would also be requested.

6.7 RELATED APPLICATION NOTES

This section lists application notes that are related to this section of the manual. These application notes may not be written specifically for the dsPIC33F/PIC24H Product Family, but the concepts are pertinent and could be used with modification and possible limitations. The current application notes related to the Interrupts module are:

Title	Application Note #
No related application notes at this time.	N/A

Note: Please visit the Microchip web site (www.microchip.com) for additional Application Notes and code examples for the dsPIC33F/PIC24H family of devices.

6.8 REVISION HISTORY

Revision A (January 2007)

This is the initial released version of this document.

Revision B (July 2008)

This revision incorporates the following updates:

- Examples:
  - The term “Programmable Low-Voltage Detect (PLVD)” in the example, in **6.1.5 “Interrupt Priority”** has been corrected as “UART1 RX Interrupt”
- Registers:
  - The bit descriptions for bit 4, bit 6, bit 11, and bit 12 in the (INTCON1): Interrupt Control Register 1 have been corrected (see Register 6-3)
  - The bit descriptions for bit 5, bit 6, bit 7, and bit 8 in the IEC1: Interrupt Enable Control Register 1 have been corrected (see Register 6-11)
  - The bit descriptions for bit 0, bit 2, bit 3, bit 4, and bit 14 in the IEC2: Interrupt Enable Control Register 2 have been corrected (see Register 6-12)
  - The bit descriptions for all the bits in the IEC3: Interrupt Enable Control Register 3 have been corrected (see Register 6-13)
  - The bit descriptions for all the bits in the IEC4: Interrupt Enable Control Register 4 have been corrected (see Register 6-14)
  - Added new register “INTTREG: Interrupt Control and Status Register” (see Register 6-33)
- Notes:
  - Added a note after the first paragraph in **6.1.5 “Interrupt Priority”**, which provides information on changing the interrupt priority levels “on-the-fly”
- Tables:
  - Updated the IVT Address and AIVT Address for the IRQ numbers 83-124, in Table 6-1
- Additional minor corrections such as language and formatting updates are incorporated throughout the document

Revision C (September 2011)

This revision includes the following updates:

- Added the CPU IRQ to the two-cycle interrupt timing diagrams (see Figure 6-4 and Figure 6-5)
- Removed the Address column and the unused registers from the Interrupt Controller Register Map (see Table 6-2)
- Additional minor corrections such as language and formatting updates were incorporated throughout the document

NOTES:

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Trademarks

ISBN: 978-1-61341-617-4

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
ISO/TS 16949:2009

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Cleveland
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

Santa Clara
Santa Clara, CA
Tel: 408-961-6444
Fax: 408-961-6445

Toronto
Mississauga, Ontario,
Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Hangzhou
Tel: 86-571-2819-3187
Fax: 86-571-2819-3189

China - Hong Kong SAR
Tel: 852-2401-1200
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8203-2660
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

Japan - Yokohama
Tel: 81-45-471- 6166
Fax: 81-45-471-6122

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-5778-366
Fax: 886-3-5770-955

Taiwan - Kaohsiung
Tel: 886-7-536-4818
Fax: 886-7-330-9305

Taiwan - Taipei
Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

UK - Wokingham
Tel: 44-118-921-5869
Fax: 44-118-921-5820

08/02/11