Section 32. Interrupts (Part III)

HIGHLIGHTS

This section of the manual contains the following major topics:

32.1 Introduction ............................................................................................................... 32-2
32.2 Non-maskable Traps................................................................................................. 32-7
32.3 Interrupt Processing Timing .................................................................................... 32-12
32.4 Interrupt Control and Status Registers................................................................. 32-15
32.5 Interrupt Setup Procedures...................................................................................... 32-53
32.6 Register Map ........................................................................................................... 32-56
32.7 Design Tips ............................................................................................................. 32-57
32.8 Related Application Notes....................................................................................... 32-58
32.9 Revision History....................................................................................................... 32-59
32.1 INTRODUCTION

This section describes the Interrupt Controller module specific to the low-cost, small pin count, dsPIC33F/PIC24H Digital Signal Controllers (DSCs). These devices are ideally suited for low-cost, high-performance motor control, general purpose, audio applications, and a variety of sensor applications.

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 includes 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
- 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

32.1.1 Interrupt Vector Table (IVT)

The IVT, as illustrated in Figure 32-1, resides in program memory starting at location 0x000004. The IVT contains up to 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).

32.1.2 Alternate Interrupt Vector Table (AIVT)

The AIVT is located after the IVT, as illustrated in Figure 32-1. Access to the AIVT is provided by the Enable Alternate Interrupt Vector Table control bit (ALTIVT) in the Interrupt Control Register 2 (INTCON2<15>). If the ALTIVT bit (INTCON2<15>) is set, all interrupt and exception processes 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 requiring the interrupt vectors to be reprogrammed. This feature also enables switching between applications for evaluation of different software algorithms at run time. If the AIVT is not required, the AIVT should be programmed with the same addresses that are used in the IVT.

32.1.3 Reset Sequence

A device Reset is not a true exception because the interrupt controller is not involved in the Reset process. During Reset, the dsPIC33F/PIC24H device clears its registers, which forces the Program Counter (PC) to zero. The processor then starts program execution at location 0x000000. The user application 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 32-1: IVT

<table>
<thead>
<tr>
<th>Interrupt Vector</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reset – GOTO Instruction</td>
<td>0x000000</td>
</tr>
<tr>
<td>Reset – GOTO Address</td>
<td>0x000002</td>
</tr>
<tr>
<td>Reserved</td>
<td>0x000004</td>
</tr>
<tr>
<td>Oscillator Fail Trap Vector</td>
<td></td>
</tr>
<tr>
<td>Address Error Trap Vector</td>
<td></td>
</tr>
<tr>
<td>Stack Error Trap Vector</td>
<td></td>
</tr>
<tr>
<td>Math Error Trap Vector</td>
<td></td>
</tr>
<tr>
<td>DMA Error Trap Vector</td>
<td></td>
</tr>
<tr>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>Interrupt Vector 0</td>
<td>0x000014</td>
</tr>
<tr>
<td>~</td>
<td></td>
</tr>
<tr>
<td>~</td>
<td></td>
</tr>
<tr>
<td>Interrupt Vector 52</td>
<td>0x00007C</td>
</tr>
<tr>
<td>Interrupt Vector 53</td>
<td>0x00007E</td>
</tr>
<tr>
<td>Interrupt Vector 54</td>
<td>0x000080</td>
</tr>
<tr>
<td>~</td>
<td></td>
</tr>
<tr>
<td>~</td>
<td></td>
</tr>
<tr>
<td>Interrupt Vector 116</td>
<td>0x0000FC</td>
</tr>
<tr>
<td>Interrupt Vector 117</td>
<td>0x0000FE</td>
</tr>
<tr>
<td>Reserved</td>
<td>0x000100</td>
</tr>
<tr>
<td>Reserved</td>
<td>0x000102</td>
</tr>
<tr>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>Oscillator Fail Trap Vector</td>
<td></td>
</tr>
<tr>
<td>Address Error Trap Vector</td>
<td></td>
</tr>
<tr>
<td>Stack Error Trap Vector</td>
<td></td>
</tr>
<tr>
<td>Math Error Trap Vector</td>
<td></td>
</tr>
<tr>
<td>DMA Error Trap Vector</td>
<td></td>
</tr>
<tr>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>Interrupt Vector 0</td>
<td>0x000114</td>
</tr>
<tr>
<td>~</td>
<td></td>
</tr>
<tr>
<td>~</td>
<td></td>
</tr>
<tr>
<td>Interrupt Vector 52</td>
<td>0x00017C</td>
</tr>
<tr>
<td>Interrupt Vector 53</td>
<td>0x00017E</td>
</tr>
<tr>
<td>Interrupt Vector 54</td>
<td>0x000180</td>
</tr>
<tr>
<td>~</td>
<td></td>
</tr>
<tr>
<td>~</td>
<td></td>
</tr>
<tr>
<td>Interrupt Vector 116</td>
<td>0x0001FE</td>
</tr>
<tr>
<td>Interrupt Vector 117</td>
<td>0x000200</td>
</tr>
</tbody>
</table>

Decreasing Natural Order Priority

See Table 32-1 for Interrupt Vector details.
### Table 32-1: Interrupt Vectors

<table>
<thead>
<tr>
<th>Vector Number</th>
<th>IVT Address</th>
<th>AIVT Address</th>
<th>Interrupt Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0x0000004</td>
<td>0x000104</td>
<td>Reserved</td>
</tr>
<tr>
<td>1</td>
<td>0x0000006</td>
<td>0x000106</td>
<td>Oscillator Failure</td>
</tr>
<tr>
<td>2</td>
<td>0x0000008</td>
<td>0x000108</td>
<td>Address Error</td>
</tr>
<tr>
<td>3</td>
<td>0x00000A</td>
<td>0x00010A</td>
<td>Stack Error</td>
</tr>
<tr>
<td>4</td>
<td>0x00000C</td>
<td>0x00010C</td>
<td>Math Error</td>
</tr>
<tr>
<td>5</td>
<td>0x00000E</td>
<td>0x00010E</td>
<td>Direct Memory Access (DMA) Error</td>
</tr>
<tr>
<td>6-7</td>
<td>0x000010-0x000012</td>
<td>0x000110-0x000112</td>
<td>Reserved</td>
</tr>
<tr>
<td>8</td>
<td>0x000014</td>
<td>0x000114</td>
<td>External Interrupt 0 (INT0)</td>
</tr>
<tr>
<td>9</td>
<td>0x000016</td>
<td>0x000116</td>
<td>Input Capture 1 (IC1)</td>
</tr>
<tr>
<td>10</td>
<td>0x000018</td>
<td>0x000118</td>
<td>Output Compare 1 (OC1)</td>
</tr>
<tr>
<td>11</td>
<td>0x00001A</td>
<td>0x00011A</td>
<td>Timer1 (T1)</td>
</tr>
<tr>
<td>12</td>
<td>0x00001C</td>
<td>0x00011C</td>
<td>DMA Channel 0 (DMA0)</td>
</tr>
<tr>
<td>13</td>
<td>0x00001E</td>
<td>0x00011E</td>
<td>Input Capture 2 (IC2)</td>
</tr>
<tr>
<td>14</td>
<td>0x000020</td>
<td>0x000120</td>
<td>Output Compare 2 (OC2)</td>
</tr>
<tr>
<td>15</td>
<td>0x000022</td>
<td>0x000122</td>
<td>Timer2 (T2)</td>
</tr>
<tr>
<td>16</td>
<td>0x000024</td>
<td>0x000124</td>
<td>Timer3 (T3)</td>
</tr>
<tr>
<td>17</td>
<td>0x000026</td>
<td>0x000126</td>
<td>SPI1 Error (SPI1E)</td>
</tr>
<tr>
<td>18</td>
<td>0x000028</td>
<td>0x000128</td>
<td>SPI1 Transfer Done (SPI1)</td>
</tr>
<tr>
<td>19</td>
<td>0x00002A</td>
<td>0x00012A</td>
<td>UART1 Receiver (U1RX)</td>
</tr>
<tr>
<td>20</td>
<td>0x00002C</td>
<td>0x00012C</td>
<td>UART1 Transmitter (U1TX)</td>
</tr>
<tr>
<td>21</td>
<td>0x00002E</td>
<td>0x00012E</td>
<td>ADC1 Convert Done (AD1)</td>
</tr>
<tr>
<td>22</td>
<td>0x000030</td>
<td>0x000130</td>
<td>DMA Channel 1 (DMA1)</td>
</tr>
<tr>
<td>23</td>
<td>0x000032</td>
<td>0x000132</td>
<td>Reserved</td>
</tr>
<tr>
<td>24</td>
<td>0x000034</td>
<td>0x000134</td>
<td>I2C1 Slave Events (SI2C1)</td>
</tr>
<tr>
<td>25</td>
<td>0x000036</td>
<td>0x000136</td>
<td>I2C1 Master Events (MI2C1)</td>
</tr>
<tr>
<td>26</td>
<td>0x000038</td>
<td>0x000138</td>
<td>Comparator Interrupt (CMP)</td>
</tr>
<tr>
<td>27</td>
<td>0x00003A</td>
<td>0x00013A</td>
<td>Change Notification Interrupt (CN)</td>
</tr>
<tr>
<td>28</td>
<td>0x00003C</td>
<td>0x00013C</td>
<td>External Interrupt 1 (INT1)</td>
</tr>
<tr>
<td>29</td>
<td>0x00003E</td>
<td>0x00013E</td>
<td>Reserved</td>
</tr>
<tr>
<td>30</td>
<td>0x000040</td>
<td>0x000140</td>
<td>Input Capture 7 (IC7)</td>
</tr>
<tr>
<td>31</td>
<td>0x000042</td>
<td>0x000142</td>
<td>Input Capture 8 (IC8)</td>
</tr>
<tr>
<td>32</td>
<td>0x000044</td>
<td>0x000144</td>
<td>DMA Channel 2 (DMA2)</td>
</tr>
<tr>
<td>33</td>
<td>0x000046</td>
<td>0x000146</td>
<td>Output Compare 3 (OC3)</td>
</tr>
<tr>
<td>34</td>
<td>0x000048</td>
<td>0x000148</td>
<td>Output Compare 4 (OC4)</td>
</tr>
<tr>
<td>35</td>
<td>0x00004A</td>
<td>0x00014A</td>
<td>Timer4 (T4)</td>
</tr>
<tr>
<td>36</td>
<td>0x00004C</td>
<td>0x00014C</td>
<td>Timer5 (T5)</td>
</tr>
<tr>
<td>37</td>
<td>0x00004E</td>
<td>0x00014E</td>
<td>External Interrupt 2 (INT2)</td>
</tr>
<tr>
<td>38</td>
<td>0x000050</td>
<td>0x000150</td>
<td>UART2 Receiver (U2RX)</td>
</tr>
<tr>
<td>39</td>
<td>0x000052</td>
<td>0x000152</td>
<td>UART2 Transmitter (U2TX)</td>
</tr>
<tr>
<td>40</td>
<td>0x000054</td>
<td>0x000154</td>
<td>SPI2 Error (SPI2E)</td>
</tr>
<tr>
<td>41</td>
<td>0x000056</td>
<td>0x000156</td>
<td>SPI2 Transfer Done (SPI2)</td>
</tr>
<tr>
<td>42</td>
<td>0x000058</td>
<td>0x000158</td>
<td>ECAN1 RX Data Ready (C1RX)</td>
</tr>
<tr>
<td>43</td>
<td>0x00005A</td>
<td>0x00015A</td>
<td>ECAN1 Event (C1)</td>
</tr>
<tr>
<td>44</td>
<td>0x00005C</td>
<td>0x00015C</td>
<td>DMA Channel 3 (DMA3)</td>
</tr>
<tr>
<td>45-52</td>
<td>0x00005E-0x00006C</td>
<td>0x00015E-0x00016C</td>
<td>Reserved</td>
</tr>
<tr>
<td>53</td>
<td>0x000066</td>
<td>0x000166</td>
<td>Parallel Master Port (PMP)</td>
</tr>
<tr>
<td>54</td>
<td>0x000070</td>
<td>0x000170</td>
<td>DMA Channel 4 (DMA4)</td>
</tr>
<tr>
<td>55-64</td>
<td>0x000072-0x000084</td>
<td>0x000172-0x000184</td>
<td>Reserved</td>
</tr>
</tbody>
</table>
Section 32. Interrupts (Part III)

### 32.1.4 CPU Priority Status

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

A trap is a non-maskable interrupt source intended to detect hardware and software errors (see 32.2 “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 Status bits (IPL<2:0>) in the CPU Status register (SR<7:5>)
- CPU Interrupt Priority Level 3 Status bit (IPL3) in the Core Control register (CORCON<3>)

The IPL<2:0> status bits (SR<7:5>) are readable and writable so that the user application can modify these bits to disable all sources of interrupts below a given priority level. For example, if IPL<2:0> = 3, 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 status bit (CORCON<3>) is set, a trap event is in progress. The IPL3 status bit (CORCON<3>) can be cleared, but not set, by the user application. In some applications, the IPL3 status bit (CORCON<3>) will need to be cleared when a trap has occurred and branch to an instruction other than the original instruction that caused the trap to occur. All user interrupt sources can be disabled by setting the IPL<2:0> = 111.

**Note:** The IPL<2:0> status bits (SR<7:5>) become read-only bits when interrupt nesting is disabled. For more information, refer to 32.2.4.2 “Interrupt Nesting”.

---

### Table 32-1: Interrupt Vectors (Continued)

<table>
<thead>
<tr>
<th>Vector Number</th>
<th>IVT Address</th>
<th>AIVT Address</th>
<th>Interrupt Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>65</td>
<td>0x000086</td>
<td>0x000086</td>
<td>PWM1 Period Match (PWM1)</td>
</tr>
<tr>
<td>66</td>
<td>0x000088</td>
<td>0x000088</td>
<td>Position Counter Compare (QEI1)</td>
</tr>
<tr>
<td>67</td>
<td>0x00008A</td>
<td>0x00008A</td>
<td>DCI Error (DCIE)</td>
</tr>
<tr>
<td>68</td>
<td>0x00008C</td>
<td>0x00008C</td>
<td>DCI Transfer Done (DCI)</td>
</tr>
<tr>
<td>69</td>
<td>0x00008E</td>
<td>0x00008E</td>
<td>DMA Channel 5 (DMA5)</td>
</tr>
<tr>
<td>70</td>
<td>0x000090</td>
<td>0x000090</td>
<td>Real Time Clock (RTCC)</td>
</tr>
<tr>
<td>71</td>
<td>0x000092</td>
<td>0x000092</td>
<td>PWM1 Fault A (FLTA1)</td>
</tr>
<tr>
<td>72</td>
<td>0x000094</td>
<td>0x000094</td>
<td>Reserved</td>
</tr>
<tr>
<td>73</td>
<td>0x000096</td>
<td>0x000096</td>
<td>UART1 Error (U1E)</td>
</tr>
<tr>
<td>74</td>
<td>0x000098</td>
<td>0x000098</td>
<td>UART2 Error (U2E)</td>
</tr>
<tr>
<td>75</td>
<td>0x000099</td>
<td>0x000099</td>
<td>CRC Generator Interrupt (CRC)</td>
</tr>
<tr>
<td>76</td>
<td>0x00009C</td>
<td>0x00009C</td>
<td>DMA Channel 6 (DMA6)</td>
</tr>
<tr>
<td>77</td>
<td>0x00009E</td>
<td>0x00009E</td>
<td>DMA Channel 7 (DMA7)</td>
</tr>
<tr>
<td>78</td>
<td>0x0000A0</td>
<td>0x0000A0</td>
<td>ECAN1 TX Data Request (C1TX)</td>
</tr>
<tr>
<td>79-80</td>
<td>0x0000A2-0x0000A4</td>
<td>0x0001A2-0x0001A4</td>
<td>Reserved</td>
</tr>
<tr>
<td>81</td>
<td>0x0000A6</td>
<td>0x0000A6</td>
<td>PWM2 Period Match (PWM2)</td>
</tr>
<tr>
<td>82</td>
<td>0x0000A8</td>
<td>0x0000A8</td>
<td>PWM2 Fault A (FLTA2)</td>
</tr>
<tr>
<td>83</td>
<td>0x0000AA</td>
<td>0x0000AA</td>
<td>Position Counter Compare (QEI2)</td>
</tr>
<tr>
<td>84-85</td>
<td>0x0000AC-0x0000AE</td>
<td>0x0001AC-0x0001AE</td>
<td>Reserved</td>
</tr>
<tr>
<td>86</td>
<td>0x0000B0</td>
<td>0x0000B0</td>
<td>Right Data Request (DAC1R – DAC1)</td>
</tr>
<tr>
<td>87</td>
<td>0x0000B2</td>
<td>0x0000B2</td>
<td>Left Data Request (DAC1L – DAC1)</td>
</tr>
<tr>
<td>88-125</td>
<td>0x0000B4-0x0000FE</td>
<td>0x0001B4-0x0001FE</td>
<td>Reserved</td>
</tr>
</tbody>
</table>
32.1.5 Interrupt Priority

Each peripheral interrupt source can be assigned to one of seven priority levels. The user application-assignable interrupt priority control bits for each individual interrupt are located in the Least Significant 3 bits (LSbs) of each nibble within the Interrupt Priority Control Registers (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.

More than one interrupt request source can be assigned to a specific priority level. To resolve priority conflicts within a given user application-assignable level, each source of interrupt has a natural order priority based on its location in the IVT. Table 32-1 provides 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 determined first by the user application-assignable priority of that source in the IPCx register, and 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 application-assignable 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 application-assignable priority. Interrupts with the same user application-assignable priority, but a higher natural order priority that becomes pending during the exception process, remain pending until the current exception process completes.

Each interrupt source can be assigned to one of seven priority levels. This enables the user application to assign a low natural order priority and a very high overall priority level to an interrupt. For example, Timer2 can be given a priority of 7, and the External Interrupt 0 (INT0) can be assigned to priority level 1, giving it a very low effective priority.

Note: The peripherals and sources of interrupt available in the IVT vary depending on the specific dsPIC33F/PIC24H 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 “Interrupt Controller” chapter in the specific device data sheet for more information.
32.2 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 dsPIC33F/PIC24H devices consists of five implemented sources of non-maskable traps:
- Oscillator failure
- Stack error
- Address error
- Math error
- DMA error

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 error trap has the lowest priority (see Figure 32-1). In addition, trap sources are classified into two distinct categories: soft traps and hard traps.

32.2.1 Soft Traps

The DMA 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 interrupt sources 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.

32.2.1.1 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 (EAs) 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 Status bit (STKERR) in the Interrupt Control Register 1 (INTCON1<2>). To avoid re-entry into the Trap Service Routine (TSR), the STKERR status flag must be cleared in software with a Return From Interrupt (RETFIE) instruction before the program returns from the trap.

32.2.1.2 MATH ERROR TRAP (SOFT TRAP, LEVEL 11)

Any of the following events can 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
The following three bits in the INTCON1 register enable the following types of accumulator overflow traps:

- The Accumulator A Overflow Trap Enable bit, OVATE (INTCON1<10>) enables traps for an Accumulator A overflow event
- The Accumulator B Overflow Trap Enable bit, OVBTE (INTCON1<9>) enables traps for an Accumulator B overflow event
- The Catastrophic Overflow Trap Enable Enable bit, COVTE (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 bit, OVAERR
  - Accumulator B Overflow Trap Flag bit, OVBERR
  - Accumulator A Catastrophic Overflow Trap Flag bit, COVAERR
  - Accumulator B Catastrophic Overflow Trap Flag bit, 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 Divide-by-zero Error Status bit, DIV0ERR (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 bit, SFTACERR (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 bit, MATHERR (INTCON1<4>). To avoid re-entering the TSR, the MATHERR status flag must be cleared in software with a RETFIE instruction before the program returns from the trap. Before clearing the MATHERR status bit (INTCON1<4>), all conditions that caused the trap to occur must be cleared. If the trap was due to an accumulator overflow, the Accumulator Overflow Status bits, OA and OB (SR<15:14>) must be cleared. The OA and OB status bits (SR<15:14>) are read-only, so the user application 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 (SR<15:14>).

32.2.1.3 DMA ERROR TRAP (SOFT TRAP, LEVEL 10)

A DMA 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 the CPU and DMA channels simultaneously attempt to write to a target address, the CPU is given priority and the DMA write is ignored. In this case, a DMA error trap is generated and the DMA Controller Error Status bit, DMACERR (INTCON1<5>) is set.
32.2.2 Hard Traps

Hard traps include exceptions of priority levels from 13 to 15. 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 is completed. Normal program execution flow does not resume until the trap has been acknowledged and processed.

32.2.2.1 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, and then 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 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 Status bit (TRAPR) 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. For more information, refer to Section 8. “Reset” (DS70192).

32.2.2.2 OSCILLATOR FAILURE TRAP (HARD TRAP, LEVEL 14)

An oscillator failure trap event is generated for any of the following 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 the software by polling the Oscillator Failure Trap Status bit, OSCFAIL (INTCON1<1>) or the Clock Fail Status bit, CF (OSCCON<3> in the Oscillator module). To avoid re-entering the TSR, the OSCFAIL status flag must be cleared in software with a RETFIE instruction before the program returns from the trap.

32.2.2.3 ADDRESS ERROR TRAP (HARD TRAP, LEVEL 13)

Operating conditions that can generate an address error trap include the following:

- A misaligned data word fetch is attempted. This condition occurs when an instruction performs a word access with the LSb of the EA set to ‘1’. The dsPIC33F/PIC24H CPU requires all word accesses to be aligned to an even address boundary.
- A bit manipulation instruction uses Indirect Addressing mode with the LSb of the EA 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 PC has been modified to point unimplemented program memory addresses. The PC 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 overwritten.

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

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.
32.2.3 Disable Interrupts (DISI) Instruction

The DISI 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 to 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 to 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 to 6 interrupts are re-enabled. The value specified in the DISI instruction includes all cycles due to Program Space Visibility (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 Status bit, DISI (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.

32.2.4 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 Interrupt Flag Status (IFSx) register. The IRQ causes an interrupt, if the corresponding bit in the Interrupt Enable Control (IECx) registers is set. For the rest of the instruction cycle in which the IRQ is sampled, the priorities of all pending IRQs 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 ISR is executed.

If there is a pending IRQ with a user application-assigned priority level greater than the current processor priority level that is 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 PC address value, MCU status bits, and the 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 32-2 illustrates the stack operation for interrupt event.

**Figure 32-2: Stack Operation for Interrupt Event**

This stack stores the IPL3 Status bit (CORCON<3>).
32.2.4.1 RETURN FROM INTERRUPT (RETFIE)

The RETFIE instruction unstacks the PC return address, IPL3 status bit (CORCON<3>) and SRL register to return the processor to the state and priority level that existed before the interrupt sequence.

32.2.4.2 INTERRUPT NESTING

Interrupts are nestable by default. Any ISR in progress can be interrupted by another source of interrupt with a higher user application-assigned priority level. Interrupt nesting can be disabled by setting the Interrupt Nesting Disable control bit, NSTDIS (INTCON1<15>). When the NSTDIS control bit (INTCON1<15>) 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 application-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 software from setting IPL<2:0> to a lower value, which would effectively re-enable interrupt nesting.

32.2.5 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 IECx registers, a wake-up signal is sent to the dsPIC33F/PIC24H CPU. When the device wakes from Sleep or Idle mode, one of the following 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 application-assigned interrupt priority level for the source is less 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 user application must assign the CPU priority level for the interrupt to level 1 or greater.

32.2.6 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 as an external conversion request feature.

32.2.7 External Interrupt Support

The dsPIC33F/PIC24H supports up to three external interrupt pin sources (INT0 to INT2). Each external interrupt pin has edge detection circuitry to detect the interrupt event. The INTCON2 register has three control bits (INT0EP, INT1EP and INT2EP) 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 32-4).
## 32.3  INTERRUPT PROCESSING TIMING

### 32.3.1  Interrupt Latency for One-cycle Instructions

Figure 32-3 illustrates 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 Figure 32-3 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 SRL registers are saved into a temporary buffer register. The second cycle of the interrupt process is executed as a `NOP` instruction to maintain consistency with the sequence taken during a two-cycle instruction (see 32.3.2 “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` instruction while the first instruction in the ISR is fetched.

**Figure 32-3: Interrupt Timing During a One-cycle Instruction**

<table>
<thead>
<tr>
<th>Cycle</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Peripheral interrupt event occurs at or before midpoint of this cycle</td>
</tr>
<tr>
<td>2</td>
<td>Save PC in temporary buffer</td>
</tr>
<tr>
<td>3</td>
<td>PUSH SRL and High 8 bits of PC (from temporary buffer)</td>
</tr>
<tr>
<td>4</td>
<td>PUSH Low 16 bits of PC (from temporary buffer)</td>
</tr>
</tbody>
</table>

![Diagram](image-url)
32.3.2 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 32-4 illustrates the peripheral interrupt event occurring in the instruction cycle prior to execution of the two-cycle instruction.

Figure 32-5 illustrates 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 32.3.1 “Interrupt Latency for One-cycle Instructions”).
32.3.3 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 PC. This cycle executes as a `NOP` instruction. On the fourth cycle, program execution resumes at the point where the interrupt occurred.

Figure 32-6: Return from Interrupt Timing

32.3.4 Special Conditions for Interrupt Latency

The dsPIC33F/PIC24H allows the current instruction to complete when a peripheral interrupt source becomes pending. The interrupt latency is the same for one-cycle 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 that 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
32.4  INTERRUPT CONTROL AND STATUS REGISTERS

The following registers are associated with the interrupt controller:

- **INTCON1: Interrupt Control Register 1**
  This register controls the global interrupt functions and contains the NSTDIS bit (INTCON1<15>), as well as the control and status flags for the processor trap sources.

- **INTCON2: Interrupt Control Register 2**
  This register controls the global interrupt functions and also controls external interrupt request signal behavior and use of the alternate vector table.

- **IFSx: Interrupt Flag Status Registers (see Register 32-5 through Register 32-9)**
  All interrupt request flags are maintained in the IFSx registers, where ‘x’ denotes the register number. Each source of interrupt has a status bit, that is set by the respective peripherals or external signal and cleared by the software.

- **IECx: Interrupt Enable Control Registers (see Register 32-10 through Register 32-14)**
  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 (see Register 32-15 through Register 32-31)**
  Each user interrupt source can be assigned to one of the eight priority levels. The IPCx registers set the interrupt priority level for each source of interrupt.

- **SR: CPU Status Register**
  This register is not a specific part of the interrupt controller hardware; however, it contains the IPL<2:0> status bits (SR<7:5>), which 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 a specific part of the interrupt controller hardware; however, it contains the IPL3 status bit (CORCON<3>), which indicates the current CPU priority level. IPL3 is a read-only bit so that trap events cannot be masked by the user application.

- **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 Vector Number (VECNUM<6:0>) and Interrupt level (ILR<3:0>) bit fields in the INTTREG register. The new interrupt priority level is the priority of the pending interrupt.

Each register is described in detail in the following sections.

**Note:** The total number and type of interrupt sources depend on the device variant. Refer to the “Interrupt Controller” chapter in the specific device data sheet for more information.

### 32.4.1 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 32-1. For example, the External Interrupt 0 (INT0) source has vector number and natural order priority 0. Thus, the INTOIF status bit is found in IFS0<0>. The INTO interrupt uses bit 0 of the IEC0 register as its Enable bit. The IPC0<2:0> bits assign the interrupt priority level for the INTO interrupt.
Register 32-1: SR: CPU Status Register

<table>
<thead>
<tr>
<th>bit 15</th>
<th>bit 8</th>
<th>bit 7</th>
<th>bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>R-0</td>
<td>R-0</td>
<td>R/C-0</td>
<td>R/C-0</td>
</tr>
<tr>
<td>OA</td>
<td>OB</td>
<td>SA</td>
<td>SB</td>
</tr>
<tr>
<td>bit 15-8</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R-0</td>
</tr>
<tr>
<td>bit 7-5</td>
<td>IPL&lt;2:0&gt;</td>
<td>RA</td>
<td>N</td>
</tr>
<tr>
<td>bit 7</td>
<td>bit 0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>bit 15-8</th>
<th>Not used by the Interrupt Controller</th>
</tr>
</thead>
<tbody>
<tr>
<td>For description of the SR bits, refer to the &quot;dsPIC30F/33F Programmer’s Reference Manual&quot; (DS70157).</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>bit 7-5</th>
<th>IPL&lt;2:0&gt;: CPU Interrupt Priority Level Status bits 2^{1,2}</th>
</tr>
</thead>
<tbody>
<tr>
<td>111    = CPU Interrupt Priority Level is 7 (15); user interrupts disabled</td>
<td></td>
</tr>
<tr>
<td>110    = CPU Interrupt Priority Level is 6 (14)</td>
<td></td>
</tr>
<tr>
<td>101    = CPU Interrupt Priority Level is 5 (13)</td>
<td></td>
</tr>
<tr>
<td>100    = CPU Interrupt Priority Level is 4 (12)</td>
<td></td>
</tr>
<tr>
<td>011    = CPU Interrupt Priority Level is 3 (11)</td>
<td></td>
</tr>
<tr>
<td>010    = CPU Interrupt Priority Level is 2 (10)</td>
<td></td>
</tr>
<tr>
<td>001    = CPU Interrupt Priority Level is 1 (9)</td>
<td></td>
</tr>
<tr>
<td>000    = CPU Interrupt Priority Level is 0 (8)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>bit 4-0</th>
<th>Not used by the Interrupt Controller</th>
</tr>
</thead>
<tbody>
<tr>
<td>For description of the SR bits, refer to the &quot;dsPIC30F/33F Programmer’s Reference Manual&quot; (DS70157).</td>
<td></td>
</tr>
</tbody>
</table>

Note 1: The IPL<2:0> status 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. User interrupts are disabled when IPL<3> = 1.

2: The IPL<2:0> status bits are read-only when the NSTDIS bit (INTCON1<15>) = 1.
### Section 32. Interrupts (Part III)

#### Register 32-2: CORCON: Core Control Register

<table>
<thead>
<tr>
<th>bit 15-4</th>
<th>bit 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>R/W-0</td>
<td>R/W-0</td>
</tr>
<tr>
<td>SATA</td>
<td>SATB</td>
</tr>
<tr>
<td>SATDW</td>
<td>ACCSAT</td>
</tr>
<tr>
<td>IPL3&lt;1:0&gt;</td>
<td>PSV</td>
</tr>
<tr>
<td>IF</td>
<td>RND</td>
</tr>
</tbody>
</table>

- **Legend:**
  - C = Clear only bit
  - 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

- **Not used by the Interrupt Controller**
  - For description of the CORCON bits, refer to the “dsPIC30F/33F Programmer’s Reference Manual” (DS70157).

#### bit 3

- **IPL3:** CPU Interrupt Priority Level Status bit 3<sup>(1)</sup>
  - 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**
  - For description of the CORCON bits, refer to the “dsPIC30F/33F Programmer’s Reference Manual” (DS70157).

#### Note 1:

The IPL3 bit is concatenated with the IPL<2:0> status bits (SR<7:5>) to form the CPU interrupt priority level.
### Register 32-3: INTCON1: Interrupt Control Register 1

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Value at POR</th>
<th>Set</th>
<th>Clear</th>
<th>Unknown</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>NSTDIS: Interrupt Nesting Disable bit</td>
<td>1 = Interrupt nesting is disabled</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>OVAERR: Accumulator A Overflow Trap Flag bit</td>
<td>1 = Trap was caused by overflow of Accumulator A</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>OVBERR: Accumulator B Overflow Trap Flag bit</td>
<td>1 = Trap was caused by overflow of Accumulator B</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>COVAERR: Accumulator A Catastrophic Overflow Trap Flag bit</td>
<td>1 = Trap was caused by catastrophic overflow of Accumulator A</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>COVBERR: Accumulator B Catastrophic Overflow Trap Flag bit</td>
<td>1 = Trap was caused by catastrophic overflow of Accumulator B</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>OVATE: Accumulator A Overflow Trap Enable bit</td>
<td>1 = Trap overflow of Accumulator A</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>OVBTE: Accumulator B Overflow Trap Enable bit</td>
<td>1 = Trap overflow of Accumulator B</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>COVTE: Catastrophic Overflow Trap Enable bit</td>
<td>1 = Trap on catastrophic overflow of Accumulator A or B enabled</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>SFTACERR: Shift Accumulator Error Status bit</td>
<td>1 = Math error trap was caused by an invalid accumulator shift</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>DIV0ERR: Divide-by-Zero Error Status bit</td>
<td>1 = Divide-by-Zero error trap was caused by a divide by zero</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>DMACERR: DMA Controller Error Status bit</td>
<td>1 = DMA Controller error trap has occurred</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>MATHERR: Math Error Status bit</td>
<td>1 = Math error trap has occurred</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**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
Register 32-3: INTCON1: Interrupt Control Register 1 (Continued)

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

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 32-4: INTCON2: Interrupt Control Register 2

<table>
<thead>
<tr>
<th></th>
<th>R/W-0</th>
<th>R-0</th>
<th>U-0</th>
<th>U-0</th>
<th>U-0</th>
<th>U-0</th>
<th>U-0</th>
<th>U-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALTIVT</td>
<td>DISI</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>bit 15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>U-0</th>
<th>U-0</th>
<th>U-0</th>
<th>U-0</th>
<th>U-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>INT2EP</td>
<td>INT1EP</td>
<td>INT0EP</td>
</tr>
<tr>
<td>bit 7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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: Disable Interrupts (DISI) Instruction Status bit
          1 = DISI instruction is active
          0 = DISI instruction is not active
bit 13-3  Unimplemented: Read as ‘0’
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
### Section 32. Interrupts (Part III)

**Register 32-5: IFS0: Interrupt Flag Status Register 0**

<table>
<thead>
<tr>
<th></th>
<th>U-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>bit 15</td>
<td>DMA1IF</td>
<td>AD1IF</td>
<td>U1TXIF</td>
<td>U1RXIF</td>
<td>SPI1IF</td>
<td>SPI1EIF</td>
<td>T3IF</td>
<td></td>
</tr>
<tr>
<td>bit 14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bit 13</td>
<td>T2IF</td>
<td>OC2IF</td>
<td>IC2IF</td>
<td>DMA0IF</td>
<td>T1IF</td>
<td>OC1IF</td>
<td>IC1IF</td>
<td>INT0IF</td>
</tr>
<tr>
<td>bit 7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**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 Error 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 32-5: IFS0: Interrupt Flag Status Register 0 (Continued)

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Status</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>OC1IF: Output Compare Channel 1 Interrupt Flag Status bit</td>
<td></td>
<td>1 = Interrupt request has occurred, 0 = Interrupt request has not occurred</td>
</tr>
<tr>
<td>1</td>
<td>IC1IF: Input Capture Channel 1 Interrupt Flag Status bit</td>
<td></td>
<td>1 = Interrupt request has occurred, 0 = Interrupt request has not occurred</td>
</tr>
<tr>
<td>0</td>
<td>INTOIF: External Interrupt 0 Flag Status bit</td>
<td></td>
<td>1 = Interrupt request has occurred, 0 = Interrupt request has not occurred</td>
</tr>
</tbody>
</table>
### Section 32. Interrupts (Part III)

#### Register 32-6: IFS1: Interrupt Flag Status Register 1

<table>
<thead>
<tr>
<th>Bit 15</th>
<th>Bit 14</th>
<th>Bit 13</th>
<th>Bit 12</th>
<th>Bit 11</th>
<th>Bit 10</th>
<th>Bit 9</th>
<th>Bit 8</th>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>U2TXIF</td>
<td>U2RXIF</td>
<td>INT2IF</td>
<td>T5IF</td>
<td>T4IF</td>
<td>OC4IF</td>
<td>OC3IF</td>
<td>DMA2IF</td>
<td>IC8IF</td>
<td>IC7IF</td>
<td>—</td>
<td>INT1IF</td>
<td>CNIF</td>
</tr>
</tbody>
</table>

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**
  - **Unimplemented**: Read as ‘0’

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

- **bit 3**
  - **CNIF**: Input Change Notification Interrupt Flag Status bit
  - 1 = Interrupt request has occurred
  - 0 = Interrupt request has not occurred
Register 32-6: IFS1: Interrupt Flag Status Register 1 (Continued)

bit 2  **CMIF**: Comparator Interrupt Flag Status bit
       1 = Interrupt request has occurred
       0 = Interrupt request has not occurred

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 32-7: IFS2: Interrupt Flag Status Register 2

<table>
<thead>
<tr>
<th>bit</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>Unimplemented</td>
<td>Read as ‘0’</td>
</tr>
<tr>
<td>14</td>
<td>DMA4IF:</td>
<td>DMA Channel 4 Data Transfer Complete Interrupt Flag Status bit</td>
</tr>
<tr>
<td></td>
<td>1 = Interrupt request has occurred</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 = Interrupt request has not occurred</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>PMPIF:</td>
<td>Parallel Master Port Interrupt Flag Status bit</td>
</tr>
<tr>
<td></td>
<td>1 = Interrupt request has occurred</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 = Interrupt request has not occurred</td>
<td></td>
</tr>
<tr>
<td>12-5</td>
<td>Unimplemented</td>
<td>Read as ‘0’</td>
</tr>
<tr>
<td>4</td>
<td>DMA3IF:</td>
<td>DMA Channel 3 Data Transfer Complete Interrupt Flag Status bit</td>
</tr>
<tr>
<td></td>
<td>1 = Interrupt request has occurred</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 = Interrupt request has not occurred</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>C1IF:</td>
<td>ECAN1 Event Interrupt Flag Status bit</td>
</tr>
<tr>
<td></td>
<td>1 = Interrupt request has occurred</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 = Interrupt request has not occurred</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>C1RXIF:</td>
<td>ECAN1 Receive Data Ready Interrupt Flag Status bit</td>
</tr>
<tr>
<td></td>
<td>1 = Interrupt request has occurred</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 = Interrupt request has not occurred</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>SPI2IF:</td>
<td>SPI2 Event Interrupt Flag Status bit</td>
</tr>
<tr>
<td></td>
<td>1 = Interrupt request has occurred</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 = Interrupt request has not occurred</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>SPI2EIF:</td>
<td>SPI2 Error Interrupt Flag Status bit</td>
</tr>
<tr>
<td></td>
<td>1 = Interrupt request has occurred</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 = Interrupt request has not occurred</td>
<td></td>
</tr>
</tbody>
</table>

#### 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
Register 32-8:  IFS3: Interrupt Flag Status Register 3

<table>
<thead>
<tr>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>U-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLTA1IF</td>
<td>RTCIF</td>
<td>DMA5IF</td>
<td>DCIIIF</td>
<td>DCIEIF</td>
<td>QEIIIF</td>
<td>PWM1IF</td>
<td>—</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>bit 15</th>
<th>bit 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>U-0</td>
<td>U-0</td>
</tr>
<tr>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>bit 7</th>
<th>bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>U-0</td>
<td>U-0</td>
</tr>
<tr>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

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 FLTA1IF: PWM1 Fault A Interrupt Flag Status bit
  - 1 = Interrupt request has occurred
  - 0 = Interrupt request has not occurred

- bit 14 RTCIF: Real-time Clock and Calendar Interrupt Flag Status bit
  - 1 = Interrupt request has occurred
  - 0 = Interrupt request has not occurred

- 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 DCIIIF: DCI Event 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 QEIIIF: QEII Event Interrupt Flag Status bit
  - 1 = Interrupt request has occurred
  - 0 = Interrupt request has not occurred

- bit 9 PWM1IF: PWM1 Interrupt Flag Status bit
  - 1 = Interrupt request has occurred
  - 0 = Interrupt request has not occurred

- bit 8-0 Unimplemented: Read as '0'
Interrupts (Part III)

Register 32-9: IFS4: Interrupt Flag Status Register 4

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Readable</th>
<th>Writable</th>
<th>Unimplemented</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>DAC1LIF: DAC Left Channel Interrupt Status bit</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 = Interrupt request has occurred</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 = Interrupt request has not occurred</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>DAC1RIF: DAC Right Channel Interrupt Status bit</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 = Interrupt request has occurred</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 = Interrupt request has not occurred</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13-12</td>
<td>Unimplemented: Read as '0'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>QE2IF: QEI2 Event Interrupt Status bit</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 = Interrupt request has occurred</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 = Interrupt request has not occurred</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>FLTA2IF: PWM2 Fault A Interrupt Status bit</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 = Interrupt request has occurred</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 = Interrupt request has not occurred</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>PWM2IF: PWM2 Interrupt Enable bit</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 = Interrupt request has occurred</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 = Interrupt request has not occurred</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8-7</td>
<td>Unimplemented: Read as '0'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>C1TXIF: ECAN1 Transmit Data Ready Interrupt Status bit</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 = Interrupt request has occurred</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 = Interrupt request has not occurred</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>DMA7IF: DMA Channel 7 Data Transfer Complete Interrupt Status bit</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 = Interrupt request has occurred</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 = Interrupt request has not occurred</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>DMA6IF: DMA Channel 6 Data Transfer Complete Interrupt Status bit</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 = Interrupt request has occurred</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 = Interrupt request has not occurred</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>CRCIF: CRC Generator Interrupt Status bit</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 = Interrupt request has occurred</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 = Interrupt request has not occurred</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>U2EIF: UART2 Error Interrupt Status bit</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 = Interrupt request has occurred</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 = Interrupt request has not occurred</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>U1EIF: UART1 Error Interrupt Status bit</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 = Interrupt request has occurred</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 = Interrupt request has not occurred</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>Unimplemented: Read as '0'</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## dsPIC33F/PIC24H Family Reference Manual

### Register 32-10: IEC0: Interrupt Enable Control Register 0

<table>
<thead>
<tr>
<th>Bit 15</th>
<th>Bit 14</th>
<th>Bit 13</th>
<th>Bit 12</th>
<th>Bit 11</th>
<th>Bit 10</th>
<th>Bit 9</th>
<th>Bit 8</th>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMA1IE</td>
<td>AD1IE</td>
<td>U1TXIE</td>
<td>U1RXIE</td>
<td>SPI1IE</td>
<td>SPI1EIE</td>
<td>T3IE</td>
<td>DMA0IE</td>
<td>T1IE</td>
<td>OC1IE</td>
<td>IC1IE</td>
<td>INTOIE</td>
<td>OC2IE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Legend:</th>
</tr>
</thead>
<tbody>
<tr>
<td>R = Readable bit</td>
</tr>
<tr>
<td>W = Writable bit</td>
</tr>
<tr>
<td>U = Unimplemented bit, read as ‘0’</td>
</tr>
<tr>
<td>-n = Value at POR</td>
</tr>
<tr>
<td>‘1’ = Bit is set</td>
</tr>
<tr>
<td>‘0’ = Bit is cleared</td>
</tr>
<tr>
<td>x = Bit is unknown</td>
</tr>
</tbody>
</table>

- **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 32-10: IEC0: Interrupt Enable Control Register 0 (Continued)

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>1 = Interrupt request enabled</th>
<th>0 = Interrupt request not enabled</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td><strong>OC1IE</strong>: Output Compare Channel 1 Interrupt Enable bit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td><strong>IC1IE</strong>: Input Capture Channel 1 Interrupt Enable bit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td><strong>INT0IE</strong>: External Interrupt 0 Enable bit</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**OC1IE**: Output Compare Channel 1 Interrupt Enable bit

**IC1IE**: Input Capture Channel 1 Interrupt Enable bit

**INT0IE**: External Interrupt 0 Enable bit
Register 32-11:  IEC1: Interrupt Enable Control Register 1

<table>
<thead>
<tr>
<th></th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>U2TXIE</td>
<td>U2RXIE</td>
<td>INT2IE</td>
<td>T5IE</td>
<td>T4IE</td>
<td>OC4IE</td>
<td>OC3IE</td>
<td>DMA2IE</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>bit 15</th>
<th>U2TXIE: UART2 Transmitter Interrupt Enable bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Interrupt request enabled</td>
</tr>
<tr>
<td>0</td>
<td>Interrupt request not enabled</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>bit 14</th>
<th>U2RXIE: UART2 Receiver Interrupt Enable bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Interrupt request enabled</td>
</tr>
<tr>
<td>0</td>
<td>Interrupt request not enabled</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>bit 13</th>
<th>INT2IE: External Interrupt 2 Enable bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Interrupt request enabled</td>
</tr>
<tr>
<td>0</td>
<td>Interrupt request not enabled</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>bit 12</th>
<th>T5IE: Timer5 Interrupt Enable bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Interrupt request enabled</td>
</tr>
<tr>
<td>0</td>
<td>Interrupt request not enabled</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>bit 11</th>
<th>T4IE: Timer4 Interrupt Enable bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Interrupt request enabled</td>
</tr>
<tr>
<td>0</td>
<td>Interrupt request not enabled</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>bit 10</th>
<th>OC4IE: Output Compare Channel 4 Interrupt Enable bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Interrupt request enabled</td>
</tr>
<tr>
<td>0</td>
<td>Interrupt request not enabled</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>bit 9</th>
<th>OC3IE: Output Compare Channel 3 Interrupt Enable bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Interrupt request enabled</td>
</tr>
<tr>
<td>0</td>
<td>Interrupt request not enabled</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>bit 8</th>
<th>DMA2IE: DMA Channel 2 Data Transfer Complete Interrupt Enable bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Interrupt request enabled</td>
</tr>
<tr>
<td>0</td>
<td>Interrupt request not enabled</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>bit 7</th>
<th>IC8IE: Input Capture Channel 8 Interrupt Enable bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Interrupt request enabled</td>
</tr>
<tr>
<td>0</td>
<td>Interrupt request not enabled</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>bit 6</th>
<th>IC7IE: Input Capture Channel 7 Interrupt Enable bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Interrupt request enabled</td>
</tr>
<tr>
<td>0</td>
<td>Interrupt request not enabled</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>bit 5</th>
<th>Unimplemented: Read as ‘0’</th>
</tr>
</thead>
<tbody>
<tr>
<td>bit 4</td>
<td>INT1IE: External Interrupt 1 Enable bit</td>
</tr>
<tr>
<td></td>
<td>1 = Interrupt request enabled</td>
</tr>
<tr>
<td></td>
<td>0 = Interrupt request not enabled</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>bit 3</th>
<th>CNIE: Input Change Notification Interrupt Enable bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Interrupt request enabled</td>
</tr>
<tr>
<td>0</td>
<td>Interrupt request not enabled</td>
</tr>
</tbody>
</table>
Register 32-11: IEC1: Interrupt Enable Control Register 1 (Continued)

bit 2  CMIE: Comparator Interrupt Enable bit
       1 = Interrupt request enabled
       0 = Interrupt request not enabled

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 32-12: IEC2: Interrupt Enable Control Register 2

<table>
<thead>
<tr>
<th>bit 15</th>
<th>DMA4IE</th>
<th>bit 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>U-0</td>
<td>R/W-0</td>
<td>U-0</td>
</tr>
<tr>
<td>---</td>
<td>DMA4IE</td>
<td>---</td>
</tr>
<tr>
<td>bit 14</td>
<td>PMPIE</td>
<td>bit 7</td>
</tr>
<tr>
<td>U-0</td>
<td>U-0</td>
<td>R/W-0</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>bit 13</td>
<td>SPI2IE</td>
<td>bit 0</td>
</tr>
<tr>
<td>U-0</td>
<td>U-0</td>
<td>R/W-0</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
</tbody>
</table>

**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 | DMA4IE: DMA Channel 4 Data Transfer Complete Interrupt Enable bit |
| 1 = Interrupt request enabled  
0 = Interrupt request not enabled |
| bit 13 | PMPIE: Parallel Master Port Interrupt Enable bit |
| 1 = Interrupt request enabled  
0 = Interrupt request not enabled |
| bit 12-5 | Unimplemented: Read as ’0’ |
| 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 |
| 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 32-13: IEC3: Interrupt Enable Control Register 3

<table>
<thead>
<tr>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>U-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLTA1IE</td>
<td>RTCIE</td>
<td>DMA5IE</td>
<td>DCIIE</td>
<td>DCIEIE</td>
<td>QE11IE</td>
<td>PWM1IE</td>
<td>—</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>bit 15</th>
<th>bit 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>U-0</td>
<td>U-0</td>
</tr>
<tr>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>bit 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>U-0</td>
</tr>
</tbody>
</table>

**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: FLTA1IE: PWM1 Fault A Interrupt Enable bit
- 1 = Interrupt request enabled
- 0 = Interrupt request not enabled

#### bit 14: RTCIE: Real-time Clock and Calendar Interrupt Enable bit
- 1 = Interrupt request enabled
- 0 = Interrupt request not enabled

#### 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: QE11IE: QE1 Event Interrupt Enable bit
- 1 = Interrupt request enabled
- 0 = Interrupt request not enabled

#### bit 9: PWM1IE: PWM1 Interrupt Enable bit
- 1 = Interrupt request enabled
- 0 = Interrupt request not enabled

#### bit 8-0: Unimplemented: Read as ‘0’
Register 32-14: IEC4: Interrupt Enable Control Register 4

<table>
<thead>
<tr>
<th>R/W-0</th>
<th>R/W-0</th>
<th>U-0</th>
<th>U-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>U-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAC1LIE</td>
<td>DAC1RIE</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>QEI2IE</td>
<td>FLTA2IE</td>
<td>PWM2IE</td>
</tr>
</tbody>
</table>

bit 15

<table>
<thead>
<tr>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>U-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>C1TXIE</td>
<td>DMA7IE</td>
<td>DMA6IE</td>
<td>CRCIE</td>
<td>U2IE</td>
<td>U1IE</td>
<td>—</td>
</tr>
</tbody>
</table>

bit 7

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
DAC1LIE: DAC Left Channel Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 14
DAC1RIE: DAC Right Channel Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 13-12
Unimplemented: Read as ‘0’

bit 11
QEI2IE: QEI2 Event Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 10
FLTA2IE: PWM2 Fault A Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 9
PWM2IE: PWM2 Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 8-7
Unimplemented: Read as ‘0’

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

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

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

bit 3
CRCIE: CRC Generator Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 2
U2IE: UART2 Error Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 1
U1IE: UART1 Error Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 0
Unimplemented: Read as ‘0’
### Register 32-15: IPC0: Interrupt Priority Control Register 0

<table>
<thead>
<tr>
<th>Bit</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>Unimplemented</td>
<td>Read as '0'</td>
</tr>
<tr>
<td>14-12</td>
<td>T1IP&lt;2:0&gt;</td>
<td>Timer1 Interrupt Priority bits</td>
</tr>
<tr>
<td></td>
<td></td>
<td>111 = Interrupt is priority 7 (highest priority interrupt)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>001 = Interrupt is priority 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>000 = Interrupt source is disabled</td>
</tr>
<tr>
<td>11</td>
<td>Unimplemented</td>
<td>Read as '0'</td>
</tr>
<tr>
<td>10-8</td>
<td>OC1IP&lt;2:0&gt;</td>
<td>Output Compare Channel 1 Interrupt Priority bits</td>
</tr>
<tr>
<td></td>
<td></td>
<td>111 = Interrupt is priority 7 (highest priority interrupt)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>001 = Interrupt is priority 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>000 = Interrupt source is disabled</td>
</tr>
<tr>
<td>7</td>
<td>Unimplemented</td>
<td>Read as '0'</td>
</tr>
<tr>
<td>6-4</td>
<td>IC1IP&lt;2:0&gt;</td>
<td>Input Capture Channel 1 Interrupt Priority bits</td>
</tr>
<tr>
<td></td>
<td></td>
<td>111 = Interrupt is priority 7 (highest priority interrupt)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>001 = Interrupt is priority 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>000 = Interrupt source is disabled</td>
</tr>
<tr>
<td>3</td>
<td>Unimplemented</td>
<td>Read as ‘0’</td>
</tr>
<tr>
<td>2-0</td>
<td>INT0IP&lt;2:0&gt;</td>
<td>External Interrupt 0 Priority bits</td>
</tr>
<tr>
<td></td>
<td></td>
<td>111 = Interrupt is priority 7 (highest priority interrupt)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>001 = Interrupt is priority 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>000 = Interrupt source is disabled</td>
</tr>
</tbody>
</table>
Register 32-16: IPC1: Interrupt Priority Control Register 1

<table>
<thead>
<tr>
<th>bit 15 R/W-1</th>
<th>bit 14-12 R/W-0</th>
<th>bit 11-8 R/W-0</th>
<th>bit 7-4 R/W-0</th>
<th>bit 3-0 R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>—</td>
<td>OC2IP&lt;2:0&gt;</td>
<td>—</td>
<td>DMA0IP&lt;2:0&gt;</td>
</tr>
<tr>
<td>bit 15 Unimplemented: Read as '0'</td>
<td>bit 14-12 T2IP&lt;2:0&gt;: Timer2 Interrupt Priority bits</td>
<td>111 = Interrupt is priority 7 (highest priority interrupt)</td>
<td>000 = Interrupt source is disabled</td>
<td></td>
</tr>
<tr>
<td>bit 10-8 IC2IP&lt;2:0&gt;: Output Compare Channel 2 Interrupt Priority bits</td>
<td>111 = Interrupt is priority 7 (highest priority interrupt)</td>
<td>000 = Interrupt source is disabled</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bit 7 Unimplemented: Read as '0'</td>
<td>bit 7 Unimplemented: Read as '0'</td>
<td>bit 6-4 IC2IP&lt;2:0&gt;: Input Capture Channel 2 Interrupt Priority bits</td>
<td>111 = Interrupt is priority 7 (highest priority interrupt)</td>
<td>000 = Interrupt source is disabled</td>
</tr>
<tr>
<td>bit 2-0 DMA0IP&lt;2:0&gt;: DMA Channel 0 Data Transfer complete Interrupt Priority bits</td>
<td>111 = Interrupt is priority 7 (highest priority interrupt)</td>
<td>000 = Interrupt source is disabled</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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
Section 32. Interrupts (Part III)

Register 32-17: 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

--- SPI1EIP<2:0> --- T3IP<2:0>

bit 7

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’
b14-12 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’
b10-8 SPI1EIP<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’
b6-4 SPI1IP<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’
b2-0 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 32-18: IPC3: Interrupt Priority Control Register 3

<table>
<thead>
<tr>
<th>Bit 15-11</th>
<th>Bit 10-8</th>
<th>Bit 7</th>
<th>Bit 6-4</th>
<th>Bit 3</th>
<th>Bit 2-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>U-0</td>
<td>R/W-1</td>
<td>U-0</td>
<td>R/W-0</td>
<td>U-0</td>
<td>R/W-0</td>
</tr>
<tr>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>—</td>
<td></td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>U-0</td>
<td>R/W-1</td>
<td>U-0</td>
<td>R/W-0</td>
<td>U-0</td>
<td>R/W-0</td>
</tr>
<tr>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>—</td>
<td></td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

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)
- 110 =
- 101 =
- 100 =
- 011 = Interrupt is priority 1
- 010 =
- 001 =
- 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)
- 110 =
- 101 =
- 100 =
- 011 = Interrupt is priority 1
- 010 =
- 001 =
- 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)
- 110 =
- 101 =
- 100 =
- 011 = Interrupt is priority 1
- 010 =
- 001 =
- 000 = Interrupt source is disabled

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
Register 32-19: IPC4: Interrupt Priority Control Register 4

<table>
<thead>
<tr>
<th>U-0</th>
<th>R/W-1</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>U-0</th>
<th>R/W-1</th>
<th>R/W-0</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

Unimplemented: Read as ‘0’

bit 10-8

CMIP<2:0>: Comparator 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

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

Legend:

R = Readable bit
W = Writable bit
U = Unimplemented bit, read as ‘0’

- = Value at POR
‘1’ = Bit is set
‘0’ = Bit is cleared
x = Bit is unknown
Register 32-20: IPC5: Interrupt Priority Control Register 5

<table>
<thead>
<tr>
<th>U-0</th>
<th>R/W-1</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>U-0</th>
<th>R/W-1</th>
<th>R/W-0</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>——</td>
<td></td>
<td>IC8IP&lt;2:0&gt;</td>
<td>——</td>
<td>IC7IP&lt;2:0&gt;</td>
<td>——</td>
<td></td>
<td>——</td>
</tr>
</tbody>
</table>

bit 15

<table>
<thead>
<tr>
<th>U-0</th>
<th>U-0</th>
<th>U-0</th>
<th>U-0</th>
<th>U-0</th>
<th>R/W-1</th>
<th>R/W-0</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>——</td>
<td>——</td>
<td>——</td>
<td>——</td>
<td>——</td>
<td>INT1IP&lt;2:0&gt;</td>
<td>——</td>
<td>——</td>
</tr>
</tbody>
</table>

bit 7

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 7-3  **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 32-21: IPC6: Interrupt Priority Control Register 6

<table>
<thead>
<tr>
<th>U-0</th>
<th>R/W-1</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>U-0</th>
<th>R/W-1</th>
<th>R/W-0</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>T4IP&lt;2:0&gt;</td>
<td>—</td>
<td>OC4IP&lt;2:0&gt;</td>
<td>bit 15</td>
<td>bit 8</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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 32-22:  IPC7: Interrupt Priority Control Register 7

<table>
<thead>
<tr>
<th>U-0</th>
<th>R/W-1</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>U-0</th>
<th>R/W-1</th>
<th>R/W-0</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U2TXIP&lt;2:0&gt;</td>
<td></td>
<td></td>
<td></td>
<td>U2RXIP&lt;2:0&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

bit 15  bit 8

<table>
<thead>
<tr>
<th>U-0</th>
<th>R/W-1</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>U-0</th>
<th>R/W-1</th>
<th>R/W-0</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INT2IP&lt;2:0&gt;</td>
<td></td>
<td></td>
<td></td>
<td>T5IP&lt;2:0&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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 32-23: IPC8: Interrupt Priority Control Register 8

<table>
<thead>
<tr>
<th>U-0</th>
<th>R/W-1</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>U-0</th>
<th>R/W-1</th>
<th>R/W-0</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>C1IP&lt;2:0&gt;</td>
<td>—</td>
<td>C1RXIP&lt;2:0&gt;</td>
<td>—</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

bit 15

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)

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)

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)

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)
### Register 32-24: IPC9: Interrupt Priority Control Register 9

<table>
<thead>
<tr>
<th>bit 15</th>
<th>bit 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>U-0</td>
<td>U-0</td>
</tr>
<tr>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

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

**bit 15-3**: **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)
- **110**
- **101**
- **100**
- **011** = Interrupt is priority 1
- **010**
- **000** = Interrupt source is disabled
Register 32-25: IPC11: Interrupt Priority Control Register 11

<table>
<thead>
<tr>
<th>U-0</th>
<th>U-0</th>
<th>U-0</th>
<th>U-0</th>
<th>U-0</th>
<th>R/W-1</th>
<th>R/W-0</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>DMA4IP&lt;2:0&gt;</td>
<td>—</td>
<td>bit 8</td>
</tr>
</tbody>
</table>

bit 15

<table>
<thead>
<tr>
<th>U-0</th>
<th>R/W-1</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>U-0</th>
<th>U-0</th>
<th>U-0</th>
<th>U-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>—</td>
<td>PMPIP&lt;2:0&gt;</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

bit 7

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  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  **Unimplemented**: Read as ‘0’

bit 6-4  PMPIP<2:0>: Parallel Master Port Interrupt Priority bits

111 = Interrupt is priority 7 (highest priority interrupt)
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 3-0  **Unimplemented**: Read as ‘0’
Register 32-26: IPC14: Interrupt Priority Control Register 14

<table>
<thead>
<tr>
<th>U-0</th>
<th>R/W-1</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>U-0</th>
<th>R/W-1</th>
<th>R/W-0</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>DCIEIP&lt;2:0&gt;</td>
<td></td>
<td></td>
<td></td>
<td>QEIEIP&lt;2:0&gt;</td>
</tr>
</tbody>
</table>

bit 15

<table>
<thead>
<tr>
<th>U-0</th>
<th>R/W-1</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>U-0</th>
<th>U-0</th>
<th>U-0</th>
<th>U-0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>PWM1IP&lt;2:0&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

bit 7

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)
- 110 = Interrupt is priority 6
- 101 = Interrupt is priority 5
- 100 = Interrupt is priority 4
- 001 = Interrupt is priority 1
- 000 = Interrupt source is disabled

bit 11
Unimplemented: Read as ‘0’

bit 10-8
QEIEIP<2:0>: QEII Interrupt Priority bits
- 111 = Interrupt is priority 7 (highest priority interrupt)
- 110 = Interrupt is priority 6
- 101 = Interrupt is priority 5
- 100 = Interrupt is priority 4
- 001 = Interrupt is priority 1
- 000 = Interrupt source is disabled

bit 7
Unimplemented: Read as ‘0’

bit 6-4
PWM1IP<2:0>: PWM1 Interrupt Priority bits
- 111 = Interrupt is priority 7 (highest priority interrupt)
- 110 = Interrupt is priority 6
- 101 = Interrupt is priority 5
- 100 = Interrupt is priority 4
- 001 = Interrupt is priority 1
- 000 = Interrupt source is disabled

bit 3-0
Unimplemented: Read as ‘0’
Register 32-27: IPC15: Interrupt Priority Control Register 15

<table>
<thead>
<tr>
<th>bit 15</th>
<th>bit 14-12</th>
<th>bit 11</th>
<th>bit 10-8</th>
<th>bit 7</th>
<th>bit 6-4</th>
<th>bit 3</th>
<th>bit 2-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>Unimplemented: Read as ‘0’</td>
<td>Unimplemented: Read as ‘0’</td>
</tr>
<tr>
<td>U-0</td>
<td>R/W-1</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>U-0</td>
<td>R/W-1</td>
<td>R/W-0</td>
<td>R/W-0</td>
</tr>
<tr>
<td>FLTA1IP&lt;2:0&gt;</td>
<td>—</td>
<td>RTCIP&lt;2:0&gt;</td>
<td>—</td>
<td>DMA5IP&lt;2:0&gt;</td>
<td>—</td>
<td>DCIIIP&lt;2:0&gt;</td>
<td>—</td>
</tr>
</tbody>
</table>

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 FLTA1IP<2:0>: PWM1 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 Unimplemented: Read as ‘0’
bit 10-8 RTCIP<2:0>: Real-time Clock and Calendar 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 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 DCIIIP<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 32-28: IPC16: Interrupt Priority Control Register 16

<table>
<thead>
<tr>
<th>bit 15</th>
<th>bit 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRCIP&lt;2:0&gt;</td>
<td>U2EIP&lt;2:0&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>bit 7</th>
<th>bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>U1EIP&lt;2:0&gt;</td>
<td></td>
</tr>
</tbody>
</table>

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  CRCIP<2:0>: CRC Generator 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  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 3-0  Unimplemented: Read as '0'
### Section 32. Interrupts (Part III)

**Register 32-29: IPC17: Interrupt Priority Control Register 17**

<table>
<thead>
<tr>
<th>Bit 15-11</th>
<th><strong>Unimplemented:</strong> Read as '0'</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit 10-8</td>
<td><strong>C1TXIP&lt;2:0&gt;</strong>: ECAN1 Transmit Data Request Interrupt Priority bits</td>
</tr>
<tr>
<td></td>
<td>111 = Interrupt is priority 7 (highest priority interrupt)</td>
</tr>
<tr>
<td></td>
<td>•</td>
</tr>
<tr>
<td></td>
<td>•</td>
</tr>
<tr>
<td></td>
<td>001 = Interrupt is priority 1</td>
</tr>
<tr>
<td></td>
<td>000 = Interrupt source is disabled</td>
</tr>
<tr>
<td>Bit 7</td>
<td><strong>Unimplemented:</strong> Read as '0'</td>
</tr>
<tr>
<td>Bit 6-4</td>
<td><strong>DMA7IP&lt;2:0&gt;</strong>: DMA Channel 7 Data Transfer Complete Interrupt Priority bits</td>
</tr>
<tr>
<td></td>
<td>111 = Interrupt is priority 7 (highest priority interrupt)</td>
</tr>
<tr>
<td></td>
<td>•</td>
</tr>
<tr>
<td></td>
<td>•</td>
</tr>
<tr>
<td></td>
<td>001 = Interrupt is priority 1</td>
</tr>
<tr>
<td></td>
<td>000 = Interrupt source is disabled</td>
</tr>
<tr>
<td>Bit 3</td>
<td><strong>Unimplemented:</strong> Read as '0'</td>
</tr>
<tr>
<td>Bit 2-0</td>
<td><strong>DMA6IP&lt;2:0&gt;</strong>: DMA Channel 6 Data Transfer Complete Interrupt Priority bits</td>
</tr>
<tr>
<td></td>
<td>111 = Interrupt is priority 7 (highest priority interrupt)</td>
</tr>
<tr>
<td></td>
<td>•</td>
</tr>
<tr>
<td></td>
<td>•</td>
</tr>
<tr>
<td></td>
<td>001 = Interrupt is priority 1</td>
</tr>
<tr>
<td></td>
<td>000 = Interrupt source is disabled</td>
</tr>
</tbody>
</table>

**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
Register 32-30: IPC18: Interrupt Priority Control Register 18

<table>
<thead>
<tr>
<th>bit 15</th>
<th>bit 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>U-0</td>
<td>R/W-1</td>
</tr>
<tr>
<td>R/W-0</td>
<td>R/W-0</td>
</tr>
<tr>
<td>R/W-0</td>
<td>U-0</td>
</tr>
<tr>
<td>R/W-1</td>
<td>R/W-0</td>
</tr>
<tr>
<td>U-0</td>
<td>U-0</td>
</tr>
<tr>
<td>R/W-1</td>
<td>R/W-0</td>
</tr>
<tr>
<td>FLTA2IP&lt;2:0&gt;</td>
<td>—</td>
</tr>
<tr>
<td>QEI2IP&lt;2:0&gt;</td>
<td>—</td>
</tr>
</tbody>
</table>

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 **QEI2IP<2:0>**: QEI2 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 **FLTA2IP<2:0>**: PWM2 Fault A 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 **PWM2IP<2:0>**: PWM2 Interrupt Priority bits
  - 111 = Interrupt is priority 7 (highest priority interrupt)
  - 001 = Interrupt is priority 1
  - 000 = Interrupt source is disabled
- bit 3-0 **Unimplemented**: Read as '0'
**Register 32-31: IPC19: Interrupt Priority Control Register 19**

<table>
<thead>
<tr>
<th>U-0</th>
<th>R/W-1</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>U-0</th>
<th>R/W-1</th>
<th>R/W-0</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>—</td>
<td>DAC1LIP&lt;2:0&gt;</td>
<td>—</td>
<td>DAC1RIP&lt;2:0&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

bit 15

<table>
<thead>
<tr>
<th>U-0</th>
<th>U-0</th>
<th>U-0</th>
<th>U-0</th>
<th>U-0</th>
<th>U-0</th>
<th>U-0</th>
<th>U-0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bit 7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**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**
  - DAC1LIP<2:0>: DAC Left Channel Interrupt Flag Status 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**
  - DAC1RIP<2:0>: DAC Right Channel Interrupt Flag Status bits
    - 111 = Interrupt is priority 7 (highest priority interrupt)
    - .
    - .
    - 001 = Interrupt is priority 1
    - 000 = Interrupt source is disabled

- **bit 7-0**
  - Unimplemented: Read as ‘0’
### dsPIC33F/PIC24H Family Reference Manual

Register 32-32: INTTREG: Interrupt Control and Status Register

<table>
<thead>
<tr>
<th>U-0</th>
<th>U-0</th>
<th>U-0</th>
<th>U-0</th>
<th>R-0</th>
<th>R-0</th>
<th>R-0</th>
<th>R-0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ILR&lt;3:0&gt;</td>
</tr>
</tbody>
</table>

*bit 15*

<table>
<thead>
<tr>
<th>U-0</th>
<th>R-0</th>
<th>R-0</th>
<th>R-0</th>
<th>R-0</th>
<th>R-0</th>
<th>R-0</th>
<th>R-0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>VECNUM&lt;6:0&gt;</td>
</tr>
</tbody>
</table>

*bit 7*

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
- 1110
- 1101
- 1100
- 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
- 0111111 = Interrupt Vector pending is number 135
- 0111110
- 0111101
- 0111100
- 0000001 = Interrupt Vector pending is number 9
- 0000000 = Interrupt Vector pending is number 8
32.5 INTERRUPT SETUP PROCEDURES

32.5.1 Initialization

To configure an interrupt source, do the following:

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

   **Note:** At a device Reset, the IPCx 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 register.
4. Enable the interrupt source by setting the interrupt enable control bit associated with the source in the appropriate IECx register.

32.5.2 Interrupt Service Routine (ISR)

The method used to declare an ISR and initialize the 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 the ISR is coded in Assembler language, it must be terminated using a `RETFIE` instruction to unstack the saved PC value, SRL value and old CPU priority level.

32.5.3 Trap Service Routine (TSR)

A 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.

32.5.4 Interrupt Disable

To disable the interrupts, do the following:

1. Push the current SR register 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, the user 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 to level 15) cannot be disabled.

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

Example 32-1 provides a sample code that enables the nested interrupts, sets up Timer1, Timer2, Timer3, and changes the notice peripherals to priority levels 2, 5, 6 and 4, respectively. It also illustrates how interrupts can be enabled and disabled using the Status register. Sample ISRs illustrate interrupt clearing.

Example 32-1: Nested Interrupt Code Example

```c
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 Timer1 interrupt priority to 2 */
    IPC0bits.T1IP = 2;

    /* Reset Timer1 interrupt flag */
    IFS0bits.T1IF = 0;

    /* Reset Timer2 interrupt flag */
    IFS0bits.T2IF = 0;

    /* Reset Timer3 interrupt flag */
    IFS0bits.T3IF = 0;

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

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

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

    /* Enable Timer3 interrupt */
    IEC0bits.T3IE = 1;
}```
Example 32-1: Nested Interrupt Code Example (Continued)

```c
/* Reset change notice interrupt flag */
IFS1bits.CNIF = 0;
return;
}

void __attribute__((interrupt, no_auto_psv)) _T1Interrupt(void)
{
    /* Insert ISR Code Here*/
    /* Clear Timer1 interrupt */
    IFS0bits.T1IF = 0;
}

void __attribute__((interrupt, no_auto_psv)) _T2Interrupt(void)
{
    /* Insert ISR Code Here*/
    /* Clear Timer2 interrupt */
    IFS0bits.T2IF = 0;
}

void __attribute__((interrupt, no_auto_psv)) _T3Interrupt(void)
{
    /* Insert ISR Code Here*/
    /* Clear Timer3 interrupt */
    IFS0bits.T3IF = 0;
}

void __attribute__((interrupt, no_auto_psv)) _CNInterrupt(void)
{
    /* Insert ISR Code Here*/
    /* Clear CN interrupt */
    IFS1bits.CNIF = 0;
}
```
### Table 32-2: Interrupt Controller Register Map

<table>
<thead>
<tr>
<th>File Name</th>
<th>Bit 15</th>
<th>Bit 14</th>
<th>Bit 13</th>
<th>Bit 12</th>
<th>Bit 11</th>
<th>Bit 10</th>
<th>Bit 9</th>
<th>Bit 8</th>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
<th>All Resets</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTCON1</td>
<td>NSTDIS</td>
<td>OVAERR</td>
<td>OVBERR</td>
<td>COVAERR</td>
<td>COVBERR</td>
<td>OVATE</td>
<td>OVTIE</td>
<td>SFTACERR</td>
<td>DIV0ERR</td>
<td>DMACERR</td>
<td>MATERR</td>
<td>ADDRERR</td>
<td>STKERR</td>
<td>OSCFAIL</td>
<td>—</td>
<td>0000</td>
<td></td>
</tr>
<tr>
<td>INTCON2</td>
<td>ALTVT</td>
<td>DISI</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>0000</td>
</tr>
<tr>
<td>IFS0</td>
<td>—</td>
<td>DMA1IF</td>
<td>AD1IF</td>
<td>U1TXIF</td>
<td>U1RXIF</td>
<td>SPI1IF</td>
<td>SPI1EIF</td>
<td>T3IF</td>
<td>T2IF</td>
<td>OC2IF</td>
<td>IC2IF</td>
<td>DMA0IF</td>
<td>T1IF</td>
<td>OC1IF</td>
<td>IC1IF</td>
<td>INT0IF</td>
<td></td>
</tr>
<tr>
<td>IFS1</td>
<td>U2TXIF</td>
<td>U2RXIF</td>
<td>INT2IF</td>
<td>T5IF</td>
<td>T4IF</td>
<td>OC4IF</td>
<td>OC3IF</td>
<td>DMA2IF</td>
<td>IC8IF</td>
<td>IC7IF</td>
<td>—</td>
<td>INT1IF</td>
<td>CNIF</td>
<td>CMIF</td>
<td>M2C1IF</td>
<td>S1ZC1IF</td>
<td></td>
</tr>
<tr>
<td>IFS2</td>
<td>—</td>
<td>DMA4IF</td>
<td>PMPIF</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>0000</td>
</tr>
<tr>
<td>IFS3</td>
<td>FLTA1IF</td>
<td>RTCIF</td>
<td>DMA5IF</td>
<td>DCIF</td>
<td>DCIEIF</td>
<td>QEI1IF</td>
<td>PWM1IF</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>DMA3IF</td>
<td>C1IF</td>
<td>C1RXIF</td>
<td>SPI2IF</td>
<td>SPI2EIF</td>
<td></td>
</tr>
<tr>
<td>IFS4</td>
<td>DAC1IF</td>
<td>DAC1RIF</td>
<td>—</td>
<td>QEI2IF</td>
<td>FLTA2IF</td>
<td>PWM2IF</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>0000</td>
<td></td>
</tr>
<tr>
<td>IEC0</td>
<td>—</td>
<td>DMA1IE</td>
<td>AD1IE</td>
<td>U1TXIE</td>
<td>U1RXIE</td>
<td>SPI1IE</td>
<td>SPI1IE</td>
<td>T3IE</td>
<td>T2IE</td>
<td>OC2IE</td>
<td>IC2IE</td>
<td>DMA0IE</td>
<td>T1IE</td>
<td>OC1IE</td>
<td>IC1IE</td>
<td>INT0IE</td>
<td></td>
</tr>
<tr>
<td>IEC1</td>
<td>U2TXIE</td>
<td>U2RXIE</td>
<td>INT2IE</td>
<td>T5IE</td>
<td>T4IE</td>
<td>OC4IE</td>
<td>OC3IE</td>
<td>DMA2IE</td>
<td>IC8IE</td>
<td>IC7IE</td>
<td>—</td>
<td>INT1IE</td>
<td>CNIE</td>
<td>CMIE</td>
<td>M2C1IE</td>
<td>S1ZC1IE</td>
<td></td>
</tr>
<tr>
<td>IEC2</td>
<td>—</td>
<td>DMA4IE</td>
<td>PMPIE</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>0000</td>
</tr>
<tr>
<td>IEC3</td>
<td>FLTA1IE</td>
<td>RTCIE</td>
<td>DMA5IE</td>
<td>DCIE</td>
<td>DCIEIE</td>
<td>QEI1IE</td>
<td>PWM1IE</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>DMA3IE</td>
<td>C1IE</td>
<td>C1RXIE</td>
<td>SPI2IE</td>
<td>SPI2IE</td>
<td></td>
</tr>
<tr>
<td>IEC4</td>
<td>DAC1IE</td>
<td>DAC1RIE</td>
<td>—</td>
<td>QEI2IE</td>
<td>FLTA2IE</td>
<td>PWM2IE</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>0000</td>
<td></td>
</tr>
<tr>
<td>IPC0</td>
<td>—</td>
<td>T1IP&lt;2:0&gt;</td>
<td>—</td>
<td>—</td>
<td>OC1IP&lt;2:0&gt;</td>
<td>—</td>
<td>IC1IP&lt;2:0&gt;</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>INT0IP&lt;2:0&gt;</td>
<td>—</td>
<td>4444</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IPC1</td>
<td>—</td>
<td>T2IP&lt;2:0&gt;</td>
<td>—</td>
<td>—</td>
<td>OC2IP&lt;2:0&gt;</td>
<td>—</td>
<td>IC2IP&lt;2:0&gt;</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>DMA0IP&lt;2:0&gt;</td>
<td>—</td>
<td>4444</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IPC2</td>
<td>—</td>
<td>U1RXIP&lt;2:0&gt;</td>
<td>—</td>
<td>—</td>
<td>SPI1IP&lt;2:0&gt;</td>
<td>—</td>
<td>SP1IP&lt;2:0&gt;</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>T3IP&lt;2:0&gt;</td>
<td>—</td>
<td>4444</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IPC3</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>DMA1IP&lt;2:0&gt;</td>
<td>—</td>
<td>AD1IP&lt;2:0&gt;</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>U1TXIP&lt;2:0&gt;</td>
<td>—</td>
<td>0444</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IPC4</td>
<td>—</td>
<td>CNIP&lt;2:0&gt;</td>
<td>—</td>
<td>—</td>
<td>CMIP&lt;2:0&gt;</td>
<td>—</td>
<td>Mi2C1IP&lt;2:0&gt;</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>Si2C1IP&lt;2:0&gt;</td>
<td>—</td>
<td>4444</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IPC5</td>
<td>—</td>
<td>IC8IP&lt;2:0&gt;</td>
<td>—</td>
<td>—</td>
<td>IC7IP&lt;2:0&gt;</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>INT1IP&lt;2:0&gt;</td>
<td>—</td>
<td>4444</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IPC6</td>
<td>—</td>
<td>T4IP&lt;2:0&gt;</td>
<td>—</td>
<td>—</td>
<td>OC4IP&lt;2:0&gt;</td>
<td>—</td>
<td>OC3IP&lt;2:0&gt;</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>DMA2IP&lt;2:0&gt;</td>
<td>—</td>
<td>4444</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IPC7</td>
<td>—</td>
<td>U2TXIP&lt;2:0&gt;</td>
<td>—</td>
<td>—</td>
<td>U2RXIP&lt;2:0&gt;</td>
<td>—</td>
<td>T2IP&lt;2:0&gt;</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>DMA2IP&lt;2:0&gt;</td>
<td>—</td>
<td>4444</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IPC8</td>
<td>—</td>
<td>C1IP&lt;2:0&gt;</td>
<td>—</td>
<td>—</td>
<td>C1RXIP&lt;2:0&gt;</td>
<td>—</td>
<td>SP2IP&lt;2:0&gt;</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>SPI2IP&lt;2:0&gt;</td>
<td>—</td>
<td>4444</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IPC9</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>DMA4IP&lt;2:0&gt;</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>DMA3IP&lt;2:0&gt;</td>
<td>—</td>
<td>0004</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IPC10</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>DMA5IP&lt;2:0&gt;</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>PCIIP&lt;2:0&gt;</td>
<td>—</td>
<td>0000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IPC11</td>
<td>—</td>
<td>DCIEIP&lt;2:0&gt;</td>
<td>—</td>
<td>—</td>
<td>QEI2IP&lt;2:0&gt;</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>PCIIP&lt;2:0&gt;</td>
<td>—</td>
<td>0000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IPC12</td>
<td>—</td>
<td>FLTA1IP&lt;2:0&gt;</td>
<td>—</td>
<td>—</td>
<td>RTcip&lt;2:0&gt;</td>
<td>—</td>
<td>DMA3IP&lt;2:0&gt;</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>OC2IP&lt;2:0&gt;</td>
<td>—</td>
<td>4444</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IPC13</td>
<td>—</td>
<td>CRCIP&lt;2:0&gt;</td>
<td>—</td>
<td>—</td>
<td>U2EIP&lt;2:0&gt;</td>
<td>—</td>
<td>U1EIP&lt;2:0&gt;</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>4440</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IPC14</td>
<td>—</td>
<td>C1TXIP&lt;2:0&gt;</td>
<td>—</td>
<td>—</td>
<td>DMA7IP&lt;2:0&gt;</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>DMAM6IP&lt;2:0&gt;</td>
<td>—</td>
<td>0444</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IPC15</td>
<td>—</td>
<td>QEI2IP&lt;2:0&gt;</td>
<td>—</td>
<td>—</td>
<td>FLTA2IP&lt;2:0&gt;</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>4440</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IPC16</td>
<td>—</td>
<td>DAC1IP&lt;2:0&gt;</td>
<td>—</td>
<td>—</td>
<td>DAC1RIP&lt;2:0&gt;</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>4400</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IPC17</td>
<td>—</td>
<td>INTREG</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>4444</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Legend:  
*x* = unknown value on Reset, — = unimplemented, read as ‘0’. Reset values are shown in hexadecimal.

Note:  
Not all bits are available on all devices. Refer to the specific device data sheet for availability.
32.7 DESIGN TIPS

Question 1: What happens when two sources of interrupt become pending at the same time and have the same user application-assigned priority level?
Answer: The interrupt source with the highest natural order priority will take precedence. The natural order priority is determined by the 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 application.
32.8  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 (Part III) module are:

<table>
<thead>
<tr>
<th>Title</th>
<th>Application Note #</th>
</tr>
</thead>
<tbody>
<tr>
<td>No related application notes at this time.</td>
<td></td>
</tr>
</tbody>
</table>

**Note:** Please visit the Microchip web site ([www.microchip.com](http://www.microchip.com)) for additional Application Notes and code examples for the dsPIC33F/PIC24H family of devices.
32.9 REVISION HISTORY

Revision A (October 2007)

This is the initial released version of the document

Revision B (February 2009)

This revision includes the following updates:

- Note:
  - Removed incorrect note reference on Software modification of the DISICNT register in 32.2.3 “Disable Interrupts (DISI) Instruction"
- Registers:
  - The bit descriptions for bit 6, bit 11 and bit 12 in the INTCON1: Interrupt Control Register 1 have been corrected (see Register 32-3)
  - Minor updates to the text and formatting were incorporated throughout the document

Revision C (May 2012)

This revision incorporates the following updates:

- Registers:
  - Updated the bit 7-5 name IPL2<2:0> to IPL<2:0> in Register 32-1
  - Updated the bit 4 name “Arithmetic Error Status bit” to “Math Error Status bit” in Register 32-3
  - Updated the bit 9 name in Register 32-8, Register 32-9, Register 32-13 and Register 32-14
  - Updated all dsPIC33F references to dsPIC33F/PIC24H in the entire document
  - Minor updates to text and formatting were incorporated throughout the document
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.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer’s risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

Trademarks

The Microchip name and logo, the Microchip logo, dsPIC, KEELOG, KEELOG logo, MPLAB, PIC, PICmicro, PICSTART, PIC32 logo, rPIC and Uni/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

FilterLab, Hampshire, HI-TECH C, Linear Active Thermistor, MXDEV, MXMLAB, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, HI-TIDE, In-Circuit Serial Programming, ICSP, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, mTouch, Omniscent Code Generation, PICC, PICC-18, PICDEM, PICDEM.net, PICkit, PICtail, REAL ICE, rflAB, Select Mode, Total Endurance, TSHARC, UniWinDriver, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

© 2007-2012, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

Printed on recycled paper.

## Worldwide Sales and Service

<table>
<thead>
<tr>
<th>AMERICAS</th>
<th>ASIA/PACIFIC</th>
<th>ASIA/PACIFIC</th>
<th>EUROPE</th>
</tr>
</thead>
</table>
| 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: 317-773-9323  
Fax: 317-773-5453  
Los Angeles  
Mission Viejo, CA  
Tel: 949-462-9523  
Fax: 949-462-9608  
Santa Clara  
Santa Clara, CA  
Tel: 408-861-6444  
Fax: 408-961-6445  
Toronto  
Mississauga, Ontario, Canada  
Tel: 905-673-0699  
Fax: 905-673-6509  | 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: 86-10-8590-7000  
Fax: 86-10-8528-2104  
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 - Shenzhen  
Tel: 86-755-8203-2660  
Fax: 86-755-8203-1760  
China - Wuhan  
Tel: 86-27-5980-5300  
Fax: 86-27-5980-5118  
China - Xiamen  
Tel: 86-592-2388138  
Fax: 86-592-2388130  
China - Zhuhai  
Tel: 86-756-3210040  
Fax: 86-756-3210049  | 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 - Osaka  
Tel: 81-66-152-7160  
Fax: 81-66-152-9310  
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  | 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  |

© 2007-2012 Microchip Technology Inc.