Section 6. Interrupts

HIGHLIGHTS

This section of the manual contains the following topics:

6.1 Introduction ................................................................. 6-2
6.2 Non-Maskable Traps .................................................... 6-7
6.3 Interrupt Processing Timing ....................................... 6-12
6.4 Interrupt Control and Status Registers ...................... 6-15
6.5 Interrupt Setup Procedures ....................................... 6-55
6.6 Design Tips ............................................................... 6-60
6.7 Related Application Notes ......................................... 6-61
6.8 Revision History .......................................................... 6-62
6.1 INTRODUCTION

The PIC24H Interrupt Controller module reduces the numerous peripheral interrupt request signals to a single interrupt request signal to the PIC24H CPU. It has these features:

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

6.1.1 Interrupt Vector Table

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

6.1.2 Alternate Vector Table

Figure 6-1 shows the AIVT that is located after the IVT. Access to the AIVT is provided by the Enable Alternate Interrupt Vector Table (ALTIVT) control bit in Interrupt Control Register 2 (INTCON2<15>). If the ALTIVT bit is set, all interrupt and exception 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 needed, the AIVT should be programmed with the same addresses used in the IVT.

6.1.3 Reset Sequence

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

**Note:** Any unimplemented or unused vector locations in the IVT and AIVT should be programmed with the address of a default interrupt handler routine that contains a RESET instruction.
Section 6. Interrupts

Figure 6-1: Interrupt Vector Table

<table>
<thead>
<tr>
<th>Interrupt Vector Table</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>0x000006</td>
</tr>
<tr>
<td>Address Error Trap Vector</td>
<td>0x000008</td>
</tr>
<tr>
<td>Stack Error Trap Vector</td>
<td>0x00000A</td>
</tr>
<tr>
<td>Math Error Trap Vector</td>
<td>0x00000C</td>
</tr>
<tr>
<td>DMAC Error Trap Vector</td>
<td>0x00000E</td>
</tr>
<tr>
<td>Reserved</td>
<td>0x000010</td>
</tr>
<tr>
<td>Reserved</td>
<td>0x000012</td>
</tr>
<tr>
<td>Interrupt Vector 0</td>
<td>0x000014</td>
</tr>
<tr>
<td>Interrupt Vector 1</td>
<td>0x000016</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>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>0x000104</td>
</tr>
<tr>
<td>Oscillator Fail Trap Vector</td>
<td>0x000106</td>
</tr>
<tr>
<td>Address Error Trap Vector</td>
<td>0x000108</td>
</tr>
<tr>
<td>Stack Error Trap Vector</td>
<td>0x00010A</td>
</tr>
<tr>
<td>Math Error Trap Vector</td>
<td>0x00010C</td>
</tr>
<tr>
<td>DMAC Error Trap Vector</td>
<td>0x00010E</td>
</tr>
<tr>
<td>Reserved</td>
<td>0x000110</td>
</tr>
<tr>
<td>Reserved</td>
<td>0x000112</td>
</tr>
<tr>
<td>Interrupt Vector 0</td>
<td>0x000114</td>
</tr>
<tr>
<td>Interrupt Vector 1</td>
<td>0x000116</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>Interrupt Vector 116</td>
<td>0x0001FC</td>
</tr>
<tr>
<td>Interrupt Vector 117</td>
<td>0x0001FE</td>
</tr>
<tr>
<td>Start of Code</td>
<td>0x000200</td>
</tr>
</tbody>
</table>

Refer to Table 6-1 for Interrupt Vector details.
### Table 6-1: Interrupt Vector Details

<table>
<thead>
<tr>
<th>IRQ #</th>
<th>IVT Address</th>
<th>AIVT Address</th>
<th>Interrupt Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0x000004</td>
<td>0x000104</td>
<td>Reserved</td>
</tr>
<tr>
<td>1</td>
<td>0x000006</td>
<td>0x000106</td>
<td>Oscillator Failure</td>
</tr>
<tr>
<td>2</td>
<td>0x000008</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>DMAC Error</td>
</tr>
<tr>
<td>6</td>
<td>0x000010</td>
<td>0x000110</td>
<td>Reserved</td>
</tr>
<tr>
<td>7</td>
<td>0x000012</td>
<td>0x000112</td>
<td>Reserved</td>
</tr>
<tr>
<td>8</td>
<td>0x000014</td>
<td>0x000114</td>
<td>INT0 – External Interrupt 0</td>
</tr>
<tr>
<td>9</td>
<td>0x000016</td>
<td>0x000116</td>
<td>OC1 – Output Compare 1</td>
</tr>
<tr>
<td>10</td>
<td>0x000018</td>
<td>0x000118</td>
<td>T1 – Timer1</td>
</tr>
<tr>
<td>11</td>
<td>0x00001A</td>
<td>0x00011A</td>
<td>DMA0 – DMA Channel 0</td>
</tr>
<tr>
<td>12</td>
<td>0x00001C</td>
<td>0x00011C</td>
<td>IC2 – Input Capture 2</td>
</tr>
<tr>
<td>13</td>
<td>0x000020</td>
<td>0x000120</td>
<td>OC2 – Output Compare 2</td>
</tr>
<tr>
<td>14</td>
<td>0x000022</td>
<td>0x000122</td>
<td>T2 – Timer2</td>
</tr>
<tr>
<td>15</td>
<td>0x000024</td>
<td>0x000124</td>
<td>T3 – Timer3</td>
</tr>
<tr>
<td>16</td>
<td>0x000026</td>
<td>0x000126</td>
<td>SPI1E – SPI 1 Fault</td>
</tr>
<tr>
<td>17</td>
<td>0x000028</td>
<td>0x000128</td>
<td>SPI1 – SPI 1 Transfer Done</td>
</tr>
<tr>
<td>18</td>
<td>0x00002A</td>
<td>0x00012A</td>
<td>U1RX – UART1 Receiver</td>
</tr>
<tr>
<td>19</td>
<td>0x00002C</td>
<td>0x00012C</td>
<td>U1TX – UART1 Transmitter</td>
</tr>
<tr>
<td>20</td>
<td>0x00002E</td>
<td>0x00012E</td>
<td>AD1 – ADC1 Convert Done</td>
</tr>
<tr>
<td>21</td>
<td>0x000030</td>
<td>0x000130</td>
<td>DMA1 – DMA Channel 1</td>
</tr>
<tr>
<td>22</td>
<td>0x000032</td>
<td>0x000132</td>
<td>Reserved</td>
</tr>
<tr>
<td>23</td>
<td>0x000034</td>
<td>0x000134</td>
<td>S1C1 – I2C™ 1 Slave Event</td>
</tr>
<tr>
<td>24</td>
<td>0x000036</td>
<td>0x000136</td>
<td>M1C1 – I2C 1 Master Event</td>
</tr>
<tr>
<td>25</td>
<td>0x000038</td>
<td>0x000138</td>
<td>Reserved</td>
</tr>
<tr>
<td>26</td>
<td>0x00003A</td>
<td>0x00013A</td>
<td>CN – Input Change Interrupt</td>
</tr>
<tr>
<td>27</td>
<td>0x00003C</td>
<td>0x00013C</td>
<td>INT1 – External Interrupt 1</td>
</tr>
<tr>
<td>28</td>
<td>0x00003E</td>
<td>0x00013E</td>
<td>AD2 – ADC2 Convert Done</td>
</tr>
<tr>
<td>29</td>
<td>0x000040</td>
<td>0x000140</td>
<td>IC7 – Input Capture 7</td>
</tr>
<tr>
<td>30</td>
<td>0x000042</td>
<td>0x000142</td>
<td>IC8 – Input Capture 8</td>
</tr>
<tr>
<td>31</td>
<td>0x000044</td>
<td>0x000144</td>
<td>DMA2 – DMA Channel 2</td>
</tr>
<tr>
<td>32</td>
<td>0x000046</td>
<td>0x000146</td>
<td>OC3 – Output Compare 3</td>
</tr>
<tr>
<td>33</td>
<td>0x000048</td>
<td>0x000148</td>
<td>OC4 – Output Compare 4</td>
</tr>
<tr>
<td>34</td>
<td>0x00004A</td>
<td>0x00014A</td>
<td>T4 – Timer4</td>
</tr>
<tr>
<td>35</td>
<td>0x00004C</td>
<td>0x00014C</td>
<td>T5 – Timer5</td>
</tr>
<tr>
<td>36</td>
<td>0x00004E</td>
<td>0x00014E</td>
<td>INT2 – External Interrupt 2</td>
</tr>
<tr>
<td>37</td>
<td>0x000050</td>
<td>0x000150</td>
<td>U2RX – UART2 Receiver</td>
</tr>
<tr>
<td>38</td>
<td>0x000052</td>
<td>0x000152</td>
<td>U2TX – UART2 Transmitter</td>
</tr>
<tr>
<td>39</td>
<td>0x000054</td>
<td>0x000154</td>
<td>SPI2E – SPI2 Fault</td>
</tr>
<tr>
<td>40</td>
<td>0x000056</td>
<td>0x000156</td>
<td>SPI2 – SPI2 Transfer Done</td>
</tr>
<tr>
<td>41</td>
<td>0x000058</td>
<td>0x000158</td>
<td>C1RX – CAN1 RX Data Ready</td>
</tr>
<tr>
<td>42</td>
<td>0x00005A</td>
<td>0x00015A</td>
<td>C1 – CAN1 Event</td>
</tr>
<tr>
<td>43</td>
<td>0x00005C</td>
<td>0x00015C</td>
<td>DMA3 – DMA Channel 3</td>
</tr>
<tr>
<td>44</td>
<td>0x00005E</td>
<td>0x00015E</td>
<td>IC3 – Input Capture 3</td>
</tr>
<tr>
<td>45</td>
<td>0x000060</td>
<td>0x000160</td>
<td>IC4 – Input Capture 4</td>
</tr>
<tr>
<td>46</td>
<td>0x000062</td>
<td>0x000162</td>
<td>IC5 – Input Capture 5</td>
</tr>
<tr>
<td>47</td>
<td>0x000064</td>
<td>0x000164</td>
<td>IC6 – Input Capture 6</td>
</tr>
<tr>
<td>48</td>
<td>0x000066</td>
<td>0x000166</td>
<td>OC5 – Output Compare 5</td>
</tr>
<tr>
<td>49</td>
<td>0x000068</td>
<td>0x000168</td>
<td>OC6 – Output Compare 6</td>
</tr>
<tr>
<td>50</td>
<td>0x00006A</td>
<td>0x00016A</td>
<td>OC7 – Output Compare 7</td>
</tr>
</tbody>
</table>

Highest Natural Order Priority

© 2008 Microchip Technology Inc.
### Table 6-1: Interrupt Vector Details (Continued)

<table>
<thead>
<tr>
<th>IRQ #</th>
<th>IVT Address</th>
<th>AIVT Address</th>
<th>Interrupt Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>52</td>
<td>0x00006C</td>
<td>0x00016C</td>
<td>OC8 – Output Compare 8</td>
</tr>
<tr>
<td>53</td>
<td>0x00006E</td>
<td>0x00016E</td>
<td>Reserved</td>
</tr>
<tr>
<td>54</td>
<td>0x000070</td>
<td>0x000170</td>
<td>DMA4 – DMA Channel 4</td>
</tr>
<tr>
<td>55</td>
<td>0x000072</td>
<td>0x000172</td>
<td>T6 – Timer6</td>
</tr>
<tr>
<td>56</td>
<td>0x000074</td>
<td>0x000174</td>
<td>T7 – Timer7</td>
</tr>
<tr>
<td>57</td>
<td>0x000076</td>
<td>0x000176</td>
<td>I2C2S – I²C™ 2 Slave Event</td>
</tr>
<tr>
<td>58</td>
<td>0x000078</td>
<td>0x000178</td>
<td>I2C2M – I²C 2 Master Event</td>
</tr>
<tr>
<td>59</td>
<td>0x00007A</td>
<td>0x00017A</td>
<td>T8 – Timer8</td>
</tr>
<tr>
<td>60</td>
<td>0x00007C</td>
<td>0x00017C</td>
<td>T9 – Timer9</td>
</tr>
<tr>
<td>61</td>
<td>0x00007E</td>
<td>0x00017E</td>
<td>INT3 – External Interrupt 3</td>
</tr>
<tr>
<td>62</td>
<td>0x000080</td>
<td>0x000180</td>
<td>INT4 – External Interrupt 4</td>
</tr>
<tr>
<td>63</td>
<td>0x000082</td>
<td>0x000182</td>
<td>C2RX – CAN2 RX Data Ready</td>
</tr>
<tr>
<td>64</td>
<td>0x000084</td>
<td>0x000184</td>
<td>C2 – CAN2 Event</td>
</tr>
<tr>
<td>65</td>
<td>0x000086</td>
<td>0x000186</td>
<td>Reserved</td>
</tr>
<tr>
<td>66</td>
<td>0x000088</td>
<td>0x000188</td>
<td>Reserved</td>
</tr>
<tr>
<td>67</td>
<td>0x00008A</td>
<td>0x00018A</td>
<td>Reserved</td>
</tr>
<tr>
<td>68</td>
<td>0x00008C</td>
<td>0x00018C</td>
<td>Reserved</td>
</tr>
<tr>
<td>69</td>
<td>0x00008E</td>
<td>0x00018E</td>
<td>DMA5 – DMA Channel 5</td>
</tr>
<tr>
<td>70</td>
<td>0x000090</td>
<td>0x000190</td>
<td>Reserved</td>
</tr>
<tr>
<td>71</td>
<td>0x000092</td>
<td>0x000192</td>
<td>Reserved</td>
</tr>
<tr>
<td>72</td>
<td>0x000094</td>
<td>0x000194</td>
<td>Reserved</td>
</tr>
<tr>
<td>73</td>
<td>0x000096</td>
<td>0x000196</td>
<td>U1E – UART1 Error Interrupt</td>
</tr>
<tr>
<td>74</td>
<td>0x000098</td>
<td>0x000198</td>
<td>U2E – UART2 Error Interrupt</td>
</tr>
<tr>
<td>75</td>
<td>0x000099</td>
<td>0x000199</td>
<td>Reserved</td>
</tr>
<tr>
<td>76</td>
<td>0x00009C</td>
<td>0x00019C</td>
<td>DMA6 – DMA Channel 6</td>
</tr>
<tr>
<td>77</td>
<td>0x00009E</td>
<td>0x00019E</td>
<td>DMA7 – DMA Channel 7</td>
</tr>
<tr>
<td>78</td>
<td>0x0000A0</td>
<td>0x0001A0</td>
<td>C1TX – CAN1 TX Data Request</td>
</tr>
<tr>
<td>79</td>
<td>0x0000A2</td>
<td>0x0001A2</td>
<td>C2TX – CAN2 TX Data Request</td>
</tr>
<tr>
<td>80</td>
<td>0x0000A4</td>
<td>0x0001A4</td>
<td>Reserved</td>
</tr>
<tr>
<td>81</td>
<td>0x0000A6</td>
<td>0x0001A6</td>
<td>Reserved</td>
</tr>
<tr>
<td>82</td>
<td>0x0000A8</td>
<td>0x0001A8</td>
<td>Reserved</td>
</tr>
<tr>
<td>83-124</td>
<td>0x0000AA-0x0000FC</td>
<td>0x0001AA-0x0001FC</td>
<td>Reserved</td>
</tr>
<tr>
<td>125</td>
<td>0x0000FE</td>
<td>0x0001FE</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

Lowest Natural Order Priority
6.1.4 CPU Priority Status

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

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

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

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

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

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

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

Note: The IPL<2:0> bits become read-only bits when interrupt nesting is disabled. For more information, refer to 6.2.4.2 “Interrupt Nesting”.

6.1.5 Interrupt Priority

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

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

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

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

Note: The peripherals and sources of interrupt available in the IVT vary depending on the specific PIC24H device. The sources of interrupt shown in this document represent a comprehensive listing of all interrupt sources found on PIC24H devices. For further details, refer to the specific device data sheet.
Section 6. Interrupts

6.2 NON-MASKABLE TRAPS

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

The PIC24H consists of the following implemented sources of non-maskable traps:

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

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

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

6.2.1 Soft Traps

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

6.2.1.1 STACK ERROR TRAP (SOFT TRAP, LEVEL 12)

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

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

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

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

6.2.1.2 MATH ERROR TRAP (SOFT TRAP, LEVEL 11)

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

A math error trap can be detected in software by polling the Math Error Status (MATHERR) bit (INTCON1<4>). To avoid re-entering the Trap Service Routine, the MATHERR status flag must be cleared (in software) before the program returns from the trap (with a RETFIE instruction). Before the MATHERR status bit can be cleared, all conditions that caused the trap to occur must also be cleared.
6.2.1.3 DMAC ERROR TRAP (SOFT TRAP, LEVEL 10)

A DMAC error trap occurs with these conditions:

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

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

6.2.2 Hard Traps

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

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

6.2.2.1 TRAP PRIORITY AND HARD TRAP CONFLICTS

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

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

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

6.2.2.2 OSCILLATOR FAILURE TRAP (HARD TRAP, LEVEL 14)

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

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

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

For more information about the Fail-Safe Clock Monitor, refer to Section 7 “Oscillator” (DS70227) and Section 25 “Device Configuration” (DS70231). For the latest documentation, refer to the Microchip web site at www.microchip.com.

6.2.2.3 ADDRESS ERROR TRAP (HARD TRAP, LEVEL 13)

Operating conditions that can generate an address error trap include:

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

When an address error trap occurs, data space writes are inhibited so that data is not destroyed. An address error can be detected in software by polling the ADDRERR status bit (INTCON1<3>). To avoid re-entering the Trap Service Routine, the ADDRERR status flag must be cleared (in software) before the program returns from the trap (with a RETFIE instruction).

### 6.2.3 Disable Interrupts Instruction

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

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

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

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

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

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

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

### 6.2.4 Interrupt Operation

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

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

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

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

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

After this information saves 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.
6.2.4.1 RETURN FROM INTERRUPT

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

6.2.4.2 INTERRUPT NESTING

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

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

6.2.5 Wake-Up from Sleep and Idle

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

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

**Note:** User interrupt sources assigned to CPU priority level 0 cannot wake the CPU from Sleep or Idle mode, because the interrupt source is effectively disabled. To use an interrupt as a wake-up source, the program must assign the CPU priority level for the interrupt to level 1 or greater.
6.2.6 Analog-to-Digital Converter (ADC) External Conversion Request

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

6.2.7 External Interrupt Support

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

6.3.1 Interrupt Latency for One-Cycle Instructions

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

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

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

![Diagram showing the sequence of events during an interrupt in a one-cycle instruction.](image-url)
6.3.2 Interrupt Latency for Two-Cycle Instructions

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

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

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

![Diagram showing interrupt timing during a two-cycle instruction](image)

Figure 6-5: Interrupt Timing, Interrupt Occurs During 1st Cycle of a Two-Cycle Instruction

![Diagram showing interrupt timing when interrupt occurs during the first cycle of a two-cycle instruction](image)
6.3.3 Returning from Interrupt

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

Figure 6-6: Return from Interrupt Timing

6.3.4 Special Conditions for Interrupt Latency

The PIC24H allows the current instruction to complete when a peripheral interrupt source becomes pending. The interrupt latency is the same for both one- and two-cycle instructions. However, certain conditions can increase interrupt latency by one cycle, depending on when the interrupt occurs. If a fixed latency is critical to the application, you should avoid these conditions:

- 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
6.4 INTERRUPT CONTROL AND STATUS REGISTERS

These are associated with the interrupt controller:

- **INTCON1, INTCON2 Registers**
  The following registers control global interrupt functions:
  - INTCON1 contains the Interrupt Nesting Disable (NSTDIS) bit, as well as the control and status flags for the processor trap sources.
  - INTCON2 controls external interrupt request signal behavior and use of the alternate vector table.

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

- **IECx: Interrupt Enable Control Registers**
  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**
  Each user interrupt source can be assigned to one of eight priority levels. The IPC registers set the interrupt priority level for each source of interrupt.

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

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

Each register is described in detail in the following sections.

**Note:** The total number and type of interrupt sources depend on the device variant. For further details, refer to the specific device data sheet.

### 6.4.1 Assignment of Interrupts to Control Registers

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

<table>
<thead>
<tr>
<th>bit 15-8</th>
<th>Unimplemented: Read as ‘0’</th>
</tr>
</thead>
<tbody>
<tr>
<td>bit 15-8</td>
<td></td>
</tr>
</tbody>
</table>

### bit 7-5
**IPL<2:0>: CPU Interrupt Priority Level Status bits**

- `111` = CPU interrupt priority level is 7 (15). User interrupts disabled
- `110` = CPU interrupt priority level is 6 (14)
- `101` = CPU interrupt priority level is 5 (13)
- `100` = CPU interrupt priority level is 4 (12)
- `011` = CPU interrupt priority level is 3 (11)
- `010` = CPU interrupt priority level is 2 (10)
- `001` = CPU interrupt priority level is 1 (9)
- `000` = CPU interrupt priority level is 0 (8)

### bit 4-0
**Not used by the Interrupt Controller**

(Refer to the "dsPIC30F/33F Programmer’s Reference Manual" (DS70157) for description of SR bits.)

### Note
1. The IPL<2:0> bits are concatenated with the IPL<3> bit (CORCON<3>) to form the CPU interrupt priority level. The value in parentheses indicates the IPL if IPL<3> = 1.
2. The IPL<2:0> status bits are read only when NSTDIS = 1 (INTCON1<15>).

### 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 15-8</th>
<th>Unimplemented: Read as ‘0’</th>
</tr>
</thead>
<tbody>
<tr>
<td>bit 7-8</td>
<td></td>
</tr>
<tr>
<td>R/W-0</td>
<td>R/W-0</td>
</tr>
<tr>
<td>IPL&lt;2:0&gt;</td>
<td>RA</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
Register 6-2: CORCON: Core Control Register

<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>
</tbody>
</table>

bit 15 bit 8

<table>
<thead>
<tr>
<th>U-0</th>
<th>U-0</th>
<th>U-0</th>
<th>U-0</th>
<th>R/C-0</th>
<th>R/W-0</th>
<th>U-0</th>
<th>U-0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>IPL3</td>
<td>PSV</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 15-4 Unimplemented: Read as ‘0’
bit 3 IPL3: CPU Interrupt Priority Level Status bit 3(1)
1 = CPU interrupt priority level is greater than 7
0 = CPU interrupt priority level is 7 or less

bit 2 Not used by the Interrupt Controller
(Refer to the “dsPIC30F/33F Programmer’s Reference Manual” (DS70157), for description of CORCON bits.)

bit 1-0 Unimplemented: Read as ‘0’

Note 1: The IPL3 bit is concatenated with the IPL<2:0> bits (SR<7:5>) to form the CPU interrupt priority level.
Register 6-3: INTCON1: Interrupt Control Register 1

<table>
<thead>
<tr>
<th>R/W-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>NSTDIS</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>bit 15</td>
<td>bit 8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<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>U-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>DIV0ERR</td>
<td>DMACERR</td>
<td>MATHERR</td>
<td>ADDRERR</td>
<td>STKERR</td>
<td>OSCFAIL</td>
<td>—</td>
</tr>
<tr>
<td>bit 7</td>
<td>bit 0</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   NSTDIS: Interrupt Nesting Disable bit
1 = Interrupt nesting is disabled
0 = Interrupt nesting is enabled

bit 14-7  Unimplemented: Read as ‘0’

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

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

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

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

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’
### Section 6. Interrupts

#### Register 6-4: INTCON2: Interrupt Control Register 2

<table>
<thead>
<tr>
<th>bit 15</th>
<th>bit 14</th>
<th>bit 13-5</th>
<th>bit 4</th>
<th>bit 3</th>
<th>bit 2</th>
<th>bit 1</th>
<th>bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 = Use alternate vector table</td>
<td>1 = DISI instruction is active</td>
<td>‘0’ = Bit is cleared</td>
<td>1 = Interrupt on negative edge</td>
<td>1 = Interrupt on negative edge</td>
<td>1 = Interrupt on negative edge</td>
<td>1 = Interrupt on negative edge</td>
<td>1 = Interrupt on negative edge</td>
</tr>
<tr>
<td>0 = Use standard (default) vector table</td>
<td>0 = DISI is not active</td>
<td>x = Bit is unknown</td>
<td>0 = Interrupt on positive edge</td>
<td>0 = Interrupt on positive edge</td>
<td>0 = Interrupt on positive edge</td>
<td>0 = Interrupt on positive edge</td>
<td>0 = Interrupt on positive edge</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

---

© 2008 Microchip Technology Inc. Preliminary DS70224C-page 6-19
Register 6-5: IFS0: Interrupt Flag Status 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>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>DMA01IF</td>
<td>T1IF</td>
<td>OC1IF</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'
between

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

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

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

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

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

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

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

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

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

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

bit 4 **DMA01IF**: DMA Channel 0 Data Transfer Complete Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 3 **T1IF**: Timer1 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred
### Register 6-5: IFS0: Interrupt Flag Status Register 0 (Continued)

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td><strong>OC1IF</strong>: Output Compare Channel 1 Interrupt Flag Status bit</td>
<td>Interrupt request has occurred</td>
<td>Interrupt request has not occurred</td>
</tr>
<tr>
<td>1</td>
<td><strong>IC1IF</strong>: Input Capture Channel 1 Interrupt Flag Status bit</td>
<td>Interrupt request has occurred</td>
<td>Interrupt request has not occurred</td>
</tr>
<tr>
<td>0</td>
<td><strong>INT0IF</strong>: External Interrupt 0 Flag Status bit</td>
<td>Interrupt request has occurred</td>
<td>Interrupt request has not occurred</td>
</tr>
</tbody>
</table>
Register 6-6: IFS1: Interrupt Flag Status Register 1

<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>R/W-0</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>
</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>U-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>IC8IF</td>
<td>IC7IF</td>
<td>AD2IF</td>
<td>INT1IF</td>
<td>CNIF</td>
<td>—</td>
<td>MI2C1IF</td>
<td>SI2C1IF</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  
U2TXIF: UART2 Transmitter Interrupt Flag Status bit  
1 = Interrupt request has occurred  
0 = Interrupt request has not occurred

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

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

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

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

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

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

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

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

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

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

bit 4  
INT1IF: External Interrupt 1 Flag Status bit  
1 = Interrupt request has occurred  
0 = Interrupt request has not occurred
Register 6-6: IFS1: Interrupt Flag Status Register 1 (Continued)

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

bit 2       **Unimplemented:** Read as ‘0’

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

bit 0       **SI2C1IF:** I2C1 Slave Events Interrupt Flag Status bit
            1 = Interrupt request has occurred
            0 = Interrupt request has not occurred
Register 6-7: IFS2: Interrupt Flag Status Register 2

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Value at POR</th>
<th>Set</th>
<th>Cleared</th>
<th>Unknown</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>T6IF: Timer6 Interrupt Flag Status bit</td>
<td></td>
<td>1</td>
<td>0</td>
<td>x</td>
</tr>
<tr>
<td>14</td>
<td>DMA4IF: DMA Channel 4 Data Transfer Complete Interrupt Flag Status bit</td>
<td></td>
<td>1</td>
<td>0</td>
<td>x</td>
</tr>
<tr>
<td>13</td>
<td>Unimplemented: Read as '0'</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>OC8IF: Output Compare Channel 8 Interrupt Flag Status bit</td>
<td></td>
<td>1</td>
<td>0</td>
<td>x</td>
</tr>
<tr>
<td>11</td>
<td>OC7IF: Output Compare Channel 7 Interrupt Flag Status bit</td>
<td></td>
<td>1</td>
<td>0</td>
<td>x</td>
</tr>
<tr>
<td>10</td>
<td>OC6IF: Output Compare Channel 6 Interrupt Flag Status bit</td>
<td></td>
<td>1</td>
<td>0</td>
<td>x</td>
</tr>
<tr>
<td>9</td>
<td>OC5IF: Output Compare Channel 5 Interrupt Flag Status bit</td>
<td></td>
<td>1</td>
<td>0</td>
<td>x</td>
</tr>
<tr>
<td>8</td>
<td>IC6IF: Input Capture Channel 6 Interrupt Flag Status bit</td>
<td></td>
<td>1</td>
<td>0</td>
<td>x</td>
</tr>
<tr>
<td>7</td>
<td>IC5IF: Input Capture Channel 5 Interrupt Flag Status bit</td>
<td></td>
<td>1</td>
<td>0</td>
<td>x</td>
</tr>
<tr>
<td>6</td>
<td>IC4IF: Input Capture Channel 4 Interrupt Flag Status bit</td>
<td></td>
<td>1</td>
<td>0</td>
<td>x</td>
</tr>
<tr>
<td>5</td>
<td>IC3IF: Input Capture Channel 3 Interrupt Flag Status bit</td>
<td></td>
<td>1</td>
<td>0</td>
<td>x</td>
</tr>
<tr>
<td>4</td>
<td>DMA3IF: DMA Channel 3 Data Transfer Complete Interrupt Flag Status bit</td>
<td></td>
<td>1</td>
<td>0</td>
<td>x</td>
</tr>
<tr>
<td>3</td>
<td>C1IF: ECAN1 Event Interrupt Flag Status bit</td>
<td></td>
<td>1</td>
<td>0</td>
<td>x</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 6-7: IFS2: Interrupt Flag Status Register 2 (Continued)

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

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

bit 0 **SPI2EIF**: SPI2 Error Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred
### Register 6-8: IFS3: Interrupt Flag Status Register 3

<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>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>C2RXIF</td>
<td>INT4IF</td>
<td>INT3IF</td>
<td>T9IF</td>
<td>T8IF</td>
<td>M12C2IF</td>
<td>SI2C2IF</td>
<td>T7IF</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 15-14 | Unimplemented: Read as ‘0’ |
| bit 13   | DMA5IF: DMA Channel 5 Data Transfer Complete Interrupt Flag Status bit |
|         | 1 = Interrupt request has occurred |
|         | 0 = Interrupt request has not occurred |
| bit 12-9| Unimplemented: Read as ‘0’ |
| bit 8   | C2IF: ECAN2 Event Interrupt Flag Status bit |
|         | 1 = Interrupt request has occurred |
|         | 0 = Interrupt request has not occurred |
| bit 7   | C2RXIF: ECAN2 Receive Data Ready Interrupt Flag Status bit |
|         | 1 = Interrupt request has occurred |
|         | 0 = Interrupt request has not occurred |
| bit 6   | INT4IF: External Interrupt 4 Flag Status bit |
|         | 1 = Interrupt request has occurred |
|         | 0 = Interrupt request has not occurred |
| bit 5   | INT3IF: External Interrupt 3 Flag Status bit |
|         | 1 = Interrupt request has occurred |
|         | 0 = Interrupt request has not occurred |
| bit 4   | T9IF: Timer9 Interrupt Flag Status bit |
|         | 1 = Interrupt request has occurred |
|         | 0 = Interrupt request has not occurred |
| bit 3   | T8IF: Timer8 Interrupt Flag Status bit |
|         | 1 = Interrupt request has occurred |
|         | 0 = Interrupt request has not occurred |
| bit 2   | M12C2IF: I2C2 Master Events Interrupt Flag Status bit |
|         | 1 = Interrupt request has occurred |
|         | 0 = Interrupt request has not occurred |
| bit 1   | SI2C2IF: I2C2 Slave Events Interrupt Flag Status bit |
|         | 1 = Interrupt request has occurred |
|         | 0 = Interrupt request has not occurred |
| bit 0   | T7IF: Timer7 Interrupt Flag Status bit |
|         | 1 = Interrupt request has occurred |
|         | 0 = Interrupt request has not occurred |
### Section 6. Interrupts

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

<table>
<thead>
<tr>
<th>Bit 15-8</th>
<th>Unimplemented: Read as ‘0’</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit 7</td>
<td>C2TXIF: ECAN2 Transmit Data Request Interrupt Flag Status bit</td>
</tr>
<tr>
<td></td>
<td>1 = Interrupt request has occurred</td>
</tr>
<tr>
<td></td>
<td>0 = Interrupt request has not occurred</td>
</tr>
<tr>
<td>Bit 6</td>
<td>C1TXIF: ECAN1 Transmit Data Request Interrupt Flag Status bit</td>
</tr>
<tr>
<td></td>
<td>1 = Interrupt request has occurred</td>
</tr>
<tr>
<td></td>
<td>0 = Interrupt request has not occurred</td>
</tr>
<tr>
<td>Bit 5</td>
<td>DMA7IF: DMA Channel 7 Data Transfer Complete Interrupt Flag Status bit</td>
</tr>
<tr>
<td></td>
<td>1 = Interrupt request has occurred</td>
</tr>
<tr>
<td></td>
<td>0 = Interrupt request has not occurred</td>
</tr>
<tr>
<td>Bit 4</td>
<td>DMA6IF: DMA Channel 6 Data Transfer Complete Interrupt Flag Status bit</td>
</tr>
<tr>
<td></td>
<td>1 = Interrupt request has occurred</td>
</tr>
<tr>
<td></td>
<td>0 = Interrupt request has not occurred</td>
</tr>
<tr>
<td>Bit 3</td>
<td>Unimplemented: Read as ‘0’</td>
</tr>
<tr>
<td>Bit 2</td>
<td>U2EIF: UART2 Error Interrupt Flag Status bit</td>
</tr>
<tr>
<td></td>
<td>1 = Interrupt request has occurred</td>
</tr>
<tr>
<td></td>
<td>0 = Interrupt request has not occurred</td>
</tr>
<tr>
<td>Bit 1</td>
<td>U1EIF: UART1 Error Interrupt Flag Status bit</td>
</tr>
<tr>
<td></td>
<td>1 = Interrupt request has occurred</td>
</tr>
<tr>
<td></td>
<td>0 = Interrupt request has not occurred</td>
</tr>
<tr>
<td>Bit 0</td>
<td>Unimplemented: Read as ‘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

<table>
<thead>
<tr>
<th>Bit 7</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>U-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>U-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>C2TXIF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C1TXIF</td>
<td></td>
<td></td>
<td>DMA7IF</td>
<td></td>
<td>DMA6IF</td>
<td></td>
<td>U2EIF</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>U1EIF</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit 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>
</tr>
</tbody>
</table>
## Register 6-10: IEC0: Interrupt Enable Control Register 0

<table>
<thead>
<tr>
<th>bit 15</th>
<th>DMA1IE: DMA Channel 1 Data Transfer Complete Interrupt Enable bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>bit 14</td>
<td>AD1IE: ADC1 Conversion Complete Interrupt Enable bit</td>
</tr>
<tr>
<td>bit 13</td>
<td>U1TXIE: UART1 Transmitter Interrupt Enable bit</td>
</tr>
<tr>
<td>bit 12</td>
<td>U1RXIE: UART1 Receiver Interrupt Enable bit</td>
</tr>
<tr>
<td>bit 11</td>
<td>SPI1IE: SPI1 Event Interrupt Enable bit</td>
</tr>
<tr>
<td>bit 10</td>
<td>SPI1EIE: SPI1 Error Interrupt Enable bit</td>
</tr>
<tr>
<td>bit 9</td>
<td>T3IE: Timer3 Interrupt Enable bit</td>
</tr>
<tr>
<td>bit 8</td>
<td>T2IE: Timer2 Interrupt Enable bit</td>
</tr>
<tr>
<td>bit 7</td>
<td>OC2IE: Output Compare Channel 2 Interrupt Enable bit</td>
</tr>
<tr>
<td>bit 6</td>
<td>IC2IE: Input Capture Channel 2 Interrupt Enable bit</td>
</tr>
<tr>
<td>bit 5</td>
<td>DMA0IE: DMA Channel 0 Data Transfer Complete Interrupt Enable bit</td>
</tr>
<tr>
<td>bit 4</td>
<td>T1IE: Timer1 Interrupt Enable bit</td>
</tr>
<tr>
<td>bit 3</td>
<td>U = Unimplemented bit, read as '0'</td>
</tr>
<tr>
<td>-n = Value at POR</td>
<td>'1' = Bit is set</td>
</tr>
<tr>
<td></td>
<td>'0' = Bit is cleared</td>
</tr>
<tr>
<td>R = Readable bit</td>
<td>x = Bit is unknown</td>
</tr>
<tr>
<td>W = Writable bit</td>
<td>T3IE: Timer3 Interrupt Enable bit</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 6-10: IEC0: Interrupt Enable Control Register 0 (Continued)

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

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

bit 0  **INT0IE: External Interrupt 0 Enable bit**
1 = Interrupt request enabled
0 = Interrupt request not enabled
### Register 6-11: IEC1: Interrupt Enable Control Register 1

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Field</th>
<th>Field</th>
<th>Field</th>
<th>Field</th>
<th>Field</th>
<th>Field</th>
<th>Field</th>
<th>Field</th>
<th>Field</th>
<th>Field</th>
<th>Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>UART2 Transmitter Interrupt Enable bit</td>
<td>U2TXIE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>UART2 Receiver Interrupt Enable bit</td>
<td>U2RXIE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>External Interrupt 2 Enable bit</td>
<td>INT2IE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Timer5 Interrupt Enable bit</td>
<td>T5IE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Timer4 Interrupt Enable bit</td>
<td>T4IE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Output Compare Channel 4 Interrupt Enable bit</td>
<td>OC4IE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Output Compare Channel 3 Interrupt Enable bit</td>
<td>OC3IE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>DMA Channel 2 Data Transfer Complete Interrupt Enable bit</td>
<td>DMA2IE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Input Capture Channel 8 Interrupt Enable bit</td>
<td>IC8IE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Input Capture Channel 7 Interrupt Enable bit</td>
<td>IC7IE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>ADC2 Conversion Complete Interrupt Enable bit</td>
<td>AD2IE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>External Interrupt 1 Enable bit</td>
<td>INT1IE</td>
<td></td>
<td></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**
  - **U2TXIE**: UART2 Transmitter Interrupt Enable bit
    - 1 = Interrupt request enabled
    - 0 = Interrupt request not enabled

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

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

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

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

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

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

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

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

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

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

- **bit 4**
  - **INT1IE**: External Interrupt 1 Enable bit
    - 1 = Interrupt request enabled
    - 0 = Interrupt request not enabled
### Register 6-11: IEC1: Interrupt Enable Control Register 1 (Continued)

<table>
<thead>
<tr>
<th>Bit</th>
<th>Bit Description</th>
<th>Description</th>
</tr>
</thead>
</table>
| 3   | CNIE: Input Change Notification Interrupt Enable bit | 1 = Interrupt request enabled  
0 = Interrupt request not enabled |
| 2   | Unimplemented: Read as ‘0’ | |
| 1   | MI2C1IE: I2C1 Master Events Interrupt Enable bit | 1 = Interrupt request enabled  
0 = Interrupt request not enabled |
| 0   | SI2C1IE: I2C1 Slave Events Interrupt Enable bit | 1 = Interrupt request enabled  
0 = Interrupt request not enabled |
### Register 6-12: IEC2: Interrupt Enable Control Register 2

<table>
<thead>
<tr>
<th>Bit 15</th>
</tr>
</thead>
<tbody>
<tr>
<td>T6IE</td>
</tr>
<tr>
<td>DMA4IE</td>
</tr>
<tr>
<td>—</td>
</tr>
<tr>
<td>OC8IE</td>
</tr>
<tr>
<td>OC7IE</td>
</tr>
<tr>
<td>OC6IE</td>
</tr>
<tr>
<td>OC5IE</td>
</tr>
<tr>
<td>IC6IE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit 14</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMA3IE</td>
</tr>
<tr>
<td>C1IE</td>
</tr>
<tr>
<td>C1RXIE</td>
</tr>
<tr>
<td>SPI2IE</td>
</tr>
<tr>
<td>SPI2EIE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit 13</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unimplemented</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit 12</th>
</tr>
</thead>
<tbody>
<tr>
<td>OC8IE</td>
</tr>
<tr>
<td>OC7IE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit 11</th>
</tr>
</thead>
<tbody>
<tr>
<td>OC6IE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>OC5IE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit 9</th>
</tr>
</thead>
<tbody>
<tr>
<td>IC6IE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>IC5IE</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**  
  T6IE: Timer6 Interrupt Enable bit  
  1 = Interrupt request enabled  
  0 = Interrupt request not enabled

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

- **bit 13**  
  Unimplemented: Read as ‘0’

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

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

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

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

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

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

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

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

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

- **bit 3**  
  C1IE: ECAN1 Event Interrupt Enable bit  
  1 = Interrupt request enabled  
  0 = Interrupt request not enabled
<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td><strong>C1RXIE</strong>: ECAN1 Receive Data Ready Interrupt Enable bit</td>
<td>1 = Interrupt request enabled 0 = Interrupt request not enabled</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td><strong>SPI2IE</strong>: SPI2 Event Interrupt Enable bit</td>
<td>1 = Interrupt request enabled 0 = Interrupt request not enabled</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td><strong>SPI2EIE</strong>: SPI2 Error Interrupt Enable bit</td>
<td>1 = Interrupt request enabled 0 = Interrupt request not enabled</td>
<td></td>
</tr>
</tbody>
</table>
Register 6-13: **IEC3: Interrupt Enable Control Register 3**

<table>
<thead>
<tr>
<th>bit 15</th>
<th>bit 14</th>
<th>bit 13</th>
<th>bit 12-9</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>
</tr>
</thead>
<tbody>
<tr>
<td>DMA5IE</td>
<td>INT4IE</td>
<td>INT3IE</td>
<td>T9IE</td>
<td>T8IE</td>
<td>MI2C2IE</td>
<td>SI2C2IE</td>
<td>T7IE</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

- **Unimplemented**: Read as ‘0’

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

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

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

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

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

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

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

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

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

- **T7IE**: Timer7 Interrupt Enable bit
  - 1 = Interrupt request enabled
  - 0 = Interrupt request not enabled
Register 6-14: IEC4: Interrupt Enable Control Register 4

<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>
<th>bit 2</th>
<th>bit 1</th>
<th>bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
<td>U-0</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 15-8: Unimplemented: Read as ‘0’
- bit 7: C2TXIE: ECAN2 Transmit Data Request Interrupt Enable bit
  1 = Interrupt request enabled
  0 = Interrupt request not enabled
- bit 6: C1TXIE: ECAN1 Transmit Data Request Interrupt Enable bit
  1 = Interrupt request enabled
  0 = Interrupt request not enabled
- bit 5: DMA7IE: DMA Channel 7 Data Transfer Complete Enable Status bit
  1 = Interrupt request enabled
  0 = Interrupt request not enabled
- bit 4: DMA6IE: DMA Channel 6 Data Transfer Complete Enable Status bit
  1 = Interrupt request enabled
  0 = Interrupt request not enabled
- bit 3: Unimplemented: Read as ‘0’
- bit 2: U2EIE: UART2 Error Interrupt Enable bit
  1 = Interrupt request enabled
  0 = Interrupt request not enabled
- bit 1: U1EIE: UART1 Error Interrupt Enable bit
  1 = Interrupt request enabled
  0 = Interrupt request not enabled
- bit 0: Unimplemented: Read as ‘0’
Register 6-15: IPC0: Interrupt Priority Control Register 0

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

<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>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>—</td>
<td>T2IP&lt;2:0&gt;</td>
<td>—</td>
<td>OC2IP&lt;2:0&gt;</td>
<td>—</td>
<td>IC2IP&lt;2:0&gt;</td>
<td>—</td>
<td>DMA0IP&lt;2:0&gt;</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**  
**T2IP<2:0>:** Timer2 Interrupt Priority bits

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

**bit 11**  
Unimplemented: Read as ‘0’

**bit 10-8**  
**OC2IP<2:0>:** Output Compare Channel 2 Interrupt Priority bits

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

**bit 7**  
Unimplemented: Read as ‘0’

**bit 6-4**  
**IC2IP<2:0>:** Input Capture Channel 2 Interrupt Priority bits

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

**bit 3**  
Unimplemented: Read as ‘0’

**bit 2-0**  
**DMA0IP<2:0>:** DMA Channel 0 Data Transfer Complete Interrupt Priority bits

- 111 = Interrupt is priority 7 (highest priority interrupt)
- 001 = Interrupt is priority 1
- 000 = Interrupt source is disabled
Register 6-17: IPC2: Interrupt Priority Control Register 2

<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

**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

**Unimplemented**: Read as '0'

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

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

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

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

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

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

bit 2-0 **T3IP<2:0>: Timer3 Interrupt Priority bits**
- 111 = Interrupt is priority 7 (highest priority interrupt)
- 001 = Interrupt is priority 1
- 000 = Interrupt source is disabled
## Section 6. Interrupts

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

<table>
<thead>
<tr>
<th>bit 15-11</th>
<th>DMA1IP&lt;2:0&gt;: DMA Channel 1 Data Transfer Complete Interrupt Priority bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>bit 10-8</td>
<td>AD1IP&lt;2:0&gt;: ADC1 Conversion Complete Interrupt Priority bits</td>
</tr>
<tr>
<td>bit 7</td>
<td>U1TXIP&lt;2:0&gt;: UART1 Transmitter Interrupt Priority bits</td>
</tr>
</tbody>
</table>

#### DMA1IP<2:0>:

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

#### AD1IP<2:0>:

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

#### U1TXIP<2:0>:

- **111** = Interrupt is priority 7 (highest priority interrupt)
- •
- •
- **001** = Interrupt is priority 1
- **000** = Interrupt source is disabled
Register 6-19: IPC4: Interrupt Priority Control Register 4

<table>
<thead>
<tr>
<th>bit 15</th>
<th>R/W-0</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>CNIP&lt;2:0&gt;</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 CNIP<2:0>: Change Notification Interrupt Priority bits**
  - 111 = Interrupt is priority 7 (highest priority interrupt)
  - 110
  - 100
  - 001 = Interrupt is priority 1
  - 000 = Interrupt source is disabled

- **bit 11-7 Unimplemented:** Read as '0'
- **bit 6-4 MI2C1IP<2:0>: I2C1 Master Events Interrupt Priority bits**
  - 111 = Interrupt is priority 7 (highest priority interrupt)
  - 110
  - 100
  - 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)
  - 110
  - 100
  - 001 = Interrupt is priority 1
  - 000 = Interrupt source is disabled
### Register 6-20: IPC5: Interrupt Priority Control Register 5

<table>
<thead>
<tr>
<th>bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>Unimplemented: Read as ‘0’</td>
</tr>
<tr>
<td>14-12</td>
<td>IC8IP&lt;2:0&gt;: Input Capture Channel 8 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>•</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>11</td>
<td>Unimplemented: Read as ‘0’</td>
</tr>
<tr>
<td>10-8</td>
<td>IC7IP&lt;2:0&gt;: Input Capture Channel 7 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>•</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>7</td>
<td>Unimplemented: Read as ‘0’</td>
</tr>
<tr>
<td>6-4</td>
<td>AD2IP&lt;2:0&gt;: ADC2 Conversion 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>•</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>3</td>
<td>Unimplemented: Read as ‘0’</td>
</tr>
<tr>
<td>2-0</td>
<td>INT1IP&lt;2:0&gt;: External Interrupt 1 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>•</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 6-21: 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></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T4IP&lt;2:0&gt;</td>
<td></td>
<td>OC4IP&lt;2:0&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></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>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>OC3IP&lt;2:0&gt;</td>
<td></td>
<td>DMA2IP&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 Unimplemented: Read as '0'
bit 14-12 T4IP<2:0>: Timer4 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled
bit 11 Unimplemented: Read as '0'
bit 10-8 OC4IP<2:0>: Output Compare Channel 4 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled
bit 7 Unimplemented: Read as '0'
bit 6-4 OC3IP<2:0>: Output Compare Channel 3 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled
bit 3 Unimplemented: Read as '0'
bit 2-0 DMA2IP<2:0>: DMA Channel 2 Data Transfer Complete Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled
Register 6-22: IPC7: Interrupt Priority Control Register 7

<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></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>
<td></td>
<td></td>
</tr>
<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></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bit 15</td>
<td>bit 8</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  U2TXIP<2:0>: UART2 Transmitter Interrupt Priority bits
  111 = Interrupt is priority 7 (highest priority interrupt)
  .
  .
  001 = Interrupt is priority 1
  000 = Interrupt source is disabled

bit 11  Unimplemented: Read as ‘0’
bit 10-8  U2RXIP<2:0>: UART2 Receiver Interrupt Priority bits
  111 = Interrupt is priority 7 (highest priority interrupt)
  .
  .
  001 = Interrupt is priority 1
  000 = Interrupt source is disabled

bit 7  Unimplemented: Read as ‘0’
bit 6-4  INT2IP<2:0>: External Interrupt 2 Priority bits
  111 = Interrupt is priority 7 (highest priority interrupt)
  .
  .
  001 = Interrupt is priority 1
  000 = Interrupt source is disabled

bit 3  Unimplemented: Read as ‘0’
bit 2-0  T5IP<2:0>: Timer5 Interrupt Priority bits
  111 = Interrupt is priority 7 (highest priority interrupt)
  .
  .
  001 = Interrupt is priority 1
  000 = Interrupt source is disabled
Register 6-23: IPC8: Interrupt Priority Control Register 8

<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>C1IP&lt;2:0&gt;</td>
<td></td>
<td>C1RXIP&lt;2:0&gt;</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’
- **-** = Value at POR ‘1’ = Bit is set ‘0’ = Bit is cleared ‘x’ = Bit is unknown

bit 15  **Unimplemented**: Read as ‘0’

bit 14-12  **C1IP<2:0>**: ECAN1 Event Interrupt Priority bits

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

bit 11  **Unimplemented**: Read as ‘0’

bit 10-8  **C1RXIP<2:0>**: ECAN1 Receive Data Ready Interrupt Priority bits

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

bit 7  **Unimplemented**: Read as ‘0’

bit 6-4  **SPI2IP<2:0>**: SPI2 Event Interrupt Priority bits

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

bit 3  **Unimplemented**: Read as ‘0’

bit 2-0  **SPI2EIP<2:0>**: SPI2 Error Interrupt Priority bits

- 111 = Interrupt is priority 7 (highest priority interrupt)
- **•**
- **•**
- 001 = Interrupt is priority 1
- 000 = Interrupt source is disabled
## Register 6-24: IPC9: Interrupt Priority Control Register 9

<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>bit 15</td>
<td>IC5IP&lt;2:0&gt;</td>
<td></td>
<td></td>
<td>IC4IP&lt;2:0&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<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>bit 7</td>
<td>IC3IP&lt;2:0&gt;</td>
<td></td>
<td></td>
<td>DMA3IP&lt;2:0&gt;</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 Descriptions:

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

- **bit 14-12**  
  **IC5IP<2:0>**: Input Capture Channel 5 Interrupt Priority bits
  - 111 = Interrupt is priority 7 (highest priority interrupt)
  - 110
  - 100
  - 001 = Interrupt is priority 1
  - 000 = Interrupt source is disabled

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

- **bit 10-8**  
  **IC4IP<2:0>**: Input Capture Channel 4 Interrupt Priority bits
  - 111 = Interrupt is priority 7 (highest priority interrupt)
  - 110
  - 100
  - 001 = Interrupt is priority 1
  - 000 = Interrupt source is disabled

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

- **bit 6-4**  
  **IC3IP<2:0>**: Input Capture Channel 3 Interrupt Priority bits
  - 111 = Interrupt is priority 7 (highest priority interrupt)
  - 110
  - 100
  - 001 = Interrupt is priority 1
  - 000 = Interrupt source is disabled

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

- **bit 2-0**  
  **DMA3IP<2:0>**: DMA Channel 3 Data Transfer Complete Interrupt Priority bits
  - 111 = Interrupt is priority 7 (highest priority interrupt)
  - 110
  - 100
  - 001 = Interrupt is priority 1
  - 000 = Interrupt source is disabled
Register 6-25: IPC10: Interrupt Priority Control Register 10

<table>
<thead>
<tr>
<th>Bit 15</th>
<th>R/W-1</th>
<th>R/W-0</th>
<th>OC7IP&lt;2:0&gt;</th>
<th>Bit 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>U-0</td>
<td>-</td>
<td>-</td>
<td>Unimplemented</td>
<td>R/W-0</td>
</tr>
<tr>
<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

<table>
<thead>
<tr>
<th>Bit 14-12</th>
<th>R/W-1</th>
<th>R/W-0</th>
<th>OC6IP&lt;2:0&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>U-0</td>
<td>-</td>
<td>-</td>
<td>Unimplemented</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit 11</th>
<th>R/W-1</th>
<th>R/W-0</th>
<th>OC5IP&lt;2:0&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>U-0</td>
<td>-</td>
<td>-</td>
<td>Unimplemented</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit 10-8</th>
<th>R/W-1</th>
<th>R/W-0</th>
<th>IC6IP&lt;2:0&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>U-0</td>
<td>-</td>
<td>-</td>
<td>Unimplemented</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

**bit 14-12**: OC7IP<2:0>: Output Compare Channel 7 Interrupt Priority bits

- 111 = Interrupt is priority 7 (highest priority interrupt)
- 110 = 
- 101 = 
- 100 = 
- 011 = Interrupt is priority 1
- 010 = 
- 001 = 
- 000 = Interrupt source is disabled

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

**bit 10-8**: OC6IP<2:0>: Output Compare Channel 6 Interrupt Priority bits

- 111 = Interrupt is priority 7 (highest priority interrupt)
- 110 = 
- 101 = 
- 100 = 
- 011 = Interrupt is priority 1
- 010 = 
- 001 = 
- 000 = Interrupt source is disabled

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

**bit 6-4**: OC5IP<2:0>: Output Compare Channel 5 Interrupt Priority bits

- 111 = Interrupt is priority 7 (highest priority interrupt)
- 110 = 
- 101 = 
- 100 = 
- 011 = Interrupt is priority 1
- 010 = 
- 001 = 
- 000 = Interrupt source is disabled

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

**bit 2-0**: IC6IP<2:0>: Input Capture Channel 6 Interrupt Priority bits

- 111 = Interrupt is priority 7 (highest priority interrupt)
- 110 = 
- 101 = 
- 100 = 
- 011 = Interrupt is priority 1
- 010 = 
- 001 = 
- 000 = Interrupt source is disabled
### Register 6-26: IPC11: Interrupt Priority Control Register 11

<table>
<thead>
<tr>
<th>Bit 15</th>
<th>Bit 14-12</th>
<th>Bit 11</th>
<th>Bit 10-8</th>
<th>Bit 7-3</th>
<th>Bit 2-0</th>
</tr>
</thead>
<tbody>
<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>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>—</td>
<td>T6IP&lt;2:0&gt;</td>
<td></td>
<td>DMA4IP&lt;2:0&gt;</td>
<td></td>
<td>OC8IP&lt;2:0&gt;</td>
</tr>
<tr>
<td>bit 15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>bit 8</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** T6IP<2:0>: Timer6 Interrupt Priority bits
  - 111 = Interrupt is priority 7 (highest priority interrupt)
  - 001 = Interrupt is priority 1
  - 000 = Interrupt source is disabled
- **bit 11** Unimplemented: Read as ‘0’
- **bit 10-8** DMA4IP<2:0>: DMA Channel 4 Data Transfer Complete Interrupt Priority bits
  - 111 = Interrupt is priority 7 (highest priority interrupt)
  - 001 = Interrupt is priority 1
  - 000 = Interrupt source is disabled
- **bit 7-3** Unimplemented: Read as ‘0’
- **bit 2-0** OC8IP<2:0>: Output Compare Channel 8 Interrupt Priority bits
  - 111 = Interrupt is priority 7 (highest priority interrupt)
  - 001 = Interrupt is priority 1
  - 000 = Interrupt source is disabled
Register 6-27: IPC12: Interrupt Priority Control Register 12

<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>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>—</td>
<td>T8IP&lt;2:0&gt;</td>
<td>—</td>
<td>MI2C2IP&lt;2:0&gt;</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 T8IP<2:0>: Timer8 Interrupt Priority bits

111 = Interrupt is priority 7 (highest priority interrupt)

001 = Interrupt is priority 1

000 = Interrupt source is disabled

bit 11 Unimplemented: Read as ‘0’

bit 10-8 MI2C2IP<2:0>: I2C2 Master Events Interrupt Priority bits

111 = Interrupt is priority 7 (highest priority interrupt)

001 = Interrupt is priority 1

000 = Interrupt source is disabled

bit 7 Unimplemented: Read as ‘0’

bit 6-4 SI2C2IP<2:0>: I2C2 Slave Events Interrupt Priority bits

111 = Interrupt is priority 7 (highest priority interrupt)

001 = Interrupt is priority 1

000 = Interrupt source is disabled

bit 3 Unimplemented: Read as ‘0’

bit 2-0 T7IP<2:0>: Timer7 Interrupt Priority bits

111 = Interrupt is priority 7 (highest priority interrupt)

001 = Interrupt is priority 1

000 = Interrupt source is disabled
## Section 6. Interrupts

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

<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>bit 15</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**: C2RXIP<2:0>: ECAN2 Receive Data Ready Interrupt Priority bits
  - 111 = Interrupt is priority 7 (highest priority interrupt)
  - 110
  - 101
  - 010
  - 001 = Interrupt is priority 1
  - 000 = Interrupt source is disabled
- **bit 11**: Unimplemented: Read as ‘0’
- **bit 10-8**: INT4IP<2:0>: External Interrupt 4 Priority bits
  - 111 = Interrupt is priority 7 (highest priority interrupt)
  - 110
  - 101
  - 010
  - 001 = Interrupt is priority 1
  - 000 = Interrupt source is disabled
- **bit 7**: Unimplemented: Read as ‘0’
- **bit 6-4**: INT3IP<2:0>: External Interrupt 3 Priority bits
  - 111 = Interrupt is priority 7 (highest priority interrupt)
  - 110
  - 101
  - 010
  - 001 = Interrupt is priority 1
  - 000 = Interrupt source is disabled
- **bit 3**: Unimplemented: Read as ‘0’
- **bit 2-0**: T9IP<2:0>: Timer9 Interrupt Priority bits
  - 111 = Interrupt is priority 7 (highest priority interrupt)
  - 110
  - 101
  - 010
  - 001 = Interrupt is priority 1
  - 000 = Interrupt source is disabled
Register 6-29: IPC14: Interrupt Priority Control Register 14

<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>
<th>bit 2</th>
<th>bit 1</th>
<th>bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
</tr>
<tr>
<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>
<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 15-3: **Unimplemented**: Read as '0'

bit 2-0: **C2IP<2:0>**: ECAN2 Event Interrupt Priority bits
- 111 = Interrupt is priority 7 (highest priority interrupt)
- *
- *
- *
- 001 = Interrupt is priority 1
- 000 = Interrupt source is disabled
### Register 6-30: IPC15: Interrupt Priority Control Register 15

<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>
</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 15-7 | 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 3-0 | Unimplemented: Read as ‘0’ |
Register 6-31: IPC16: Interrupt Priority Control Register 16

<table>
<thead>
<tr>
<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>
</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></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 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'
Register 6-32: IPC17: Interrupt Priority Control Register 17

<table>
<thead>
<tr>
<th>bit 15</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>C2TXIP&lt;2:0&gt;</td>
<td>—</td>
<td>—</td>
<td>C1TXIP&lt;2:0&gt;</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 C2TXIP<2:0>: ECAN2 Transmit Data Request Interrupt Priority bits

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

bit 11 Unimplemented: Read as ‘0’

bit 10-8 C1TXIP<2:0>: ECAN1 Transmit Data Request Interrupt Priority bits

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

bit 7 Unimplemented: Read as ‘0’

bit 6-4 DMA7IP<2:0>: DMA Channel 7 Data Transfer Complete Interrupt Priority bits

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

bit 3 Unimplemented: Read as ‘0’

bit 2-0 DMA6IP<2:0>: DMA Channel 6 Data Transfer Complete Interrupt Priority bits

- 111 = Interrupt is priority 7 (highest priority interrupt)
- 011 = Interrupt is priority 1
- 000 = Interrupt source is disabled
Register 6-33: INTTREG: Interrupt Control and Status Register

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

- **ILR<3:0>**
  - bit 15
  - bit 8

- **VECNUM<6:0>**
  - bit 7
  - bit 0

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

- **Unimplemented**: Read as '0'
- **ILR<3:0>**: New CPU Interrupt Priority Level bits
  - 1111 = CPU Interrupt Priority Level is 15
  - 1110
  - 1101
  - 1100
  - 1011
  - 1010
  - 1001
  - 1000
  - 0111 = CPU Interrupt Priority Level is 1
  - 0110
  - 0101
  - 0100
  - 0011
  - 0010
  - 0001
  - 0000

- **Unimplemented**: Read as '0'

- **VECNUM<6:0>**: Vector Number of Pending Interrupt bits
  - 1111111 = Interrupt vector pending is number 135
  - 1111110
  - 1111101
  - 1111100
  - 1111011
  - 1111010
  - 1111001
  - 1111000
  - 1110111
  - 1110110
  - 1110101
  - 1110100
  - 1110011
  - 1110010
  - 1110001
  - 1110000
  - 1101111
  - 1101110
  - 1101101
  - 1101100
  - 1101011
  - 1101010
  - 1101001
  - 1101000
  - 1100111
  - 1100110
  - 1100101
  - 1100100
  - 1100011
  - 1100010
  - 1100001
  - 1100000
  - 1011111
  - 1011110
  - 1011101
  - 1011100
  - 1011011
  - 1011010
  - 1011001
  - 1011000
  - 1010111
  - 1010110
  - 1010101
  - 1010100
  - 1010011
  - 1010010
  - 1010001
  - 1010000
  - 1001111
  - 1001110
  - 1001101
  - 1001100
  - 1001011
  - 1001010
  - 1001001
  - 1001000
  - 1000111
  - 1000110
  - 1000101
  - 1000100
  - 1000011
  - 1000010
  - 1000001
  - 1000000
  - 0111111
  - 0111110
  - 0111101
  - 0111100
  - 0111011
  - 0111010
  - 0111001
  - 0111000
  - 0110111
  - 0110110
  - 0110101
  - 0110100
  - 0110011
  - 0110010
  - 0110001
  - 0110000
  - 0101111
  - 0101110
  - 0101101
  - 0101100
  - 0101011
  - 0101010
  - 0101001
  - 0101000
  - 0100111
  - 0100110
  - 0100101
  - 0100100
  - 0100011
  - 0100010
  - 0100001
  - 0100000
  - 0011111
  - 0011110
  - 0011101
  - 0011100
  - 0011011
  - 0011010
  - 0011001
  - 0011000
  - 0010111
  - 0010110
  - 0010101
  - 0010100
  - 0010011
  - 0010010
  - 0010001
  - 0010000
  - 0001111
  - 0001110
  - 0001101
  - 0001100
  - 0001011
  - 0001010
  - 0001001
  - 0001000
  - 0000111
  - 0000110
  - 0000101
  - 0000100
  - 0000011
  - 0000010
  - 0000001
  - 0000000

- **Interrupt vector pending is number 9**
- **Interrupt vector pending is number 8**
6.5 INTERRUPT SETUP PROCEDURES

6.5.1 Initialization

To configure an interrupt source:

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

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

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

6.5.2 Interrupt Service Routine

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

6.5.3 Trap Service Routine

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

6.5.4 Interrupt Disable

To disable interrupts:

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

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

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

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

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

Example 6-1: Interrupt Setup 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 Timer4 interrupt priority to 3 */
    IPC6bits.T4IP = 3;

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

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

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

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

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

    /* Enable CN interrupts */
    IEC1bits.CNIE = 1;
}
```
Example 6-1: Interrupt Setup Code Example (Continued)

```c
/* Enable Timer1 interrupt */
IEC0bits.T1IE = 1;
/* Enable Timer2 interrupt (PWM time base) */
IEC0bits.T2IE = 1;
/* Enable Timer3 interrupt */
IEC0bits.T3IE = 1;
/* Enable Timer4 interrupt (replacement for Timer 2 */
IEC1bits.T4IE = 1;
/* Reset change notice interrupt flag */
IFS1bits.CNIF = 0;

return;
}

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

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

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

void __attribute__((__interrupt__)) _T4Interrupt(void)
{
    /* Insert ISR Code Here*/
    /* Clear Timer4 interrupt */
    IFS1bits.T4IF = 0;
}

void __attribute__((__interrupt__)) _CNInterrupt(void)
{
    /* Insert ISR Code Here*/
    /* Clear CN interrupt */
    IFS1bits.CNIF = 0;
}
```
### Table 6-2: Interrupt Controller Register Map

| File Name | Addr | Bit 15 | Bit 14 | Bit 13 | Bit 12 | Bit 11 | Bit 10 | Bit 9 | Bit 8 | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | All Resets |
|-----------|------|--------|--------|--------|--------|--------|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| INTCON1   | 0080 | —      | —      | —      | —      | —      | —      | —     | —     | —     | —     | DIVERR | DMACERR | MATERR | ADDRERR | STKERR | OSCFAIL | —      | 0000    |
| INTCON2   | 0082 | ALTIIT | DISI   | —      | —      | —      | —      | —     | —     | —     | —     | —      | —      | —      | —      | —      | —      | —      | 0000    |
| IFS0      | 0084 | —      | —      | —      | —      | —      | —      | —     | —     | —     | —     | —      | —      | —      | —      | —      | —      | —      | —      | 0000    |
| IFS1      | 0086 | U2TXIF | U2RXIF | T5IF   | T4IF   | OC4IF  | OC3IF  | DM2IF | IC8IF | IC7IF | AD2IF | INT1IF | CNIF   | —      | —      | —      | —      | —      | 0000    |
| IFS2      | 0088 | T6IF   | DM4IF  | —      | OC5IF  | OC7IF  | OC6IF  | OC5IF | IC6IF | IC5IF | IC4IF | IC3IF | DMA3IF | C1IF   | C1RXIF | SPI2IF | SPI2EIF | —      | 0000    |
| IFS3      | 008A | —      | —      | —      | —      | —      | —      | —     | —     | —     | —     | —      | —      | —      | —      | —      | —      | —      | —      | 0000    |
| IFS4      | 008C | —      | —      | —      | —      | —      | —      | —     | —     | —     | —     | —      | —      | —      | —      | —      | —      | —      | —      | 0000    |
| IFS5      | 008E | —      | —      | —      | —      | —      | —      | —     | —     | —     | —     | —      | —      | —      | —      | —      | —      | —      | —      | 0000    |
| IFS6      | 0090 | —      | —      | —      | —      | —      | —      | —     | —     | —     | —     | —      | —      | —      | —      | —      | —      | —      | —      | 0000    |
| IFS7      | 0092 | —      | —      | —      | —      | —      | —      | —     | —     | —     | —     | —      | —      | —      | —      | —      | —      | —      | —      | 0000    |
| IEC0      | 0094 | —      | —      | —      | —      | —      | —      | —     | —     | —     | —     | —      | —      | —      | —      | —      | —      | —      | —      | 0000    |
| IEC1      | 0096 | U2TXIE | U2RXIE | INT2IE | T5IE   | T4IE   | OC4IE  | OC3IE | DMA3IE | IC8IE | IC7IE | AD2IE | INT1IE | CNIE   | —      | —      | —      | —      | 0000    |
| IEC2      | 0098 | T6IE   | DMA4IE | —      | OC8IE  | OC7IE  | OC6IE  | OC5IE | IC6IE | IC5IE | IC4IE | IC3IE | DMA3IE | C1IE   | C1RXIE | SPI2IE | SPI2EIF | —      | 0000    |
| IEC3      | 009A | —      | —      | —      | —      | —      | —      | —     | —     | —     | —     | —      | —      | —      | —      | —      | —      | —      | —      | 0000    |
| IEC4      | 009C | —      | —      | —      | —      | —      | —      | —     | —     | —     | —     | —      | —      | —      | —      | —      | —      | —      | —      | 0000    |
| IEC5      | 009E | —      | —      | —      | —      | —      | —      | —     | —     | —     | —     | —      | —      | —      | —      | —      | —      | —      | —      | 0000    |
| IEC6      | 00A0 | —      | —      | —      | —      | —      | —      | —     | —     | —     | —     | —      | —      | —      | —      | —      | —      | —      | —      | 0000    |
| IEC7      | 00A2 | —      | —      | —      | —      | —      | —      | —     | —     | —     | —     | —      | —      | —      | —      | —      | —      | —      | —      | 0000    |
| IPC0      | 00A4 | —      | —      | —      | —      | —      | —      | —     | —     | —     | —     | —      | —      | —      | —      | —      | —      | —      | —      | 0000    |
| IPC1      | 00A6 | —      | —      | —      | —      | —      | —      | —     | —     | —     | —     | —      | —      | —      | —      | —      | —      | —      | —      | 0000    |
| IPC2      | 00A8 | —      | —      | —      | —      | —      | —      | —     | —     | —     | —     | —      | —      | —      | —      | —      | —      | —      | —      | 0000    |
| IPC3      | 00AA | —      | —      | —      | —      | —      | —      | —     | —     | —     | —     | —      | —      | —      | —      | —      | —      | —      | —      | 0000    |
| IPC4      | 00AC | —      | —      | —      | —      | —      | —      | —     | —     | —     | —     | —      | —      | —      | —      | —      | —      | —      | —      | 0000    |
| IPC5      | 00AE | —      | —      | —      | —      | —      | —      | —     | —     | —     | —     | —      | —      | —      | —      | —      | —      | —      | —      | 0000    |
| IPC6      | 00B0 | —      | —      | —      | —      | —      | —      | —     | —     | —     | —     | —      | —      | —      | —      | —      | —      | —      | —      | 0000    |
| IPC7      | 00B2 | —      | —      | —      | —      | —      | —      | —     | —     | —     | —     | —      | —      | —      | —      | —      | —      | —      | —      | 0000    |
| IPC8      | 00B4 | —      | —      | —      | —      | —      | —      | —     | —     | —     | —     | —      | —      | —      | —      | —      | —      | —      | —      | 0000    |
| IPC9      | 00B6 | —      | —      | —      | —      | —      | —      | —     | —     | —     | —     | —      | —      | —      | —      | —      | —      | —      | —      | 0000    |
| IPC10     | 00B8 | —      | —      | —      | —      | —      | —      | —     | —     | —     | —     | —      | —      | —      | —      | —      | —      | —      | —      | 0000    |
| IPC11     | 00BA | —      | —      | —      | —      | —      | —      | —     | —     | —     | —     | —      | —      | —      | —      | —      | —      | —      | —      | 0000    |
| IPC12     | 00BC | —      | —      | —      | —      | —      | —      | —     | —     | —     | —     | —      | —      | —      | —      | —      | —      | —      | —      | 0000    |

**Legend:**

- = unimplemented, read as '0'. Reset values are shown in hexadecimal.
### Table 6-2: Interrupt Controller Register Map (Continued)

<table>
<thead>
<tr>
<th>File Name</th>
<th>Addr</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>IPC13</td>
<td>00BE</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>—</td>
<td>—</td>
<td>4444</td>
</tr>
<tr>
<td>IPC14</td>
<td>00C0</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>—</td>
<td>—</td>
<td>4444</td>
</tr>
<tr>
<td>IPC15</td>
<td>00C2</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>—</td>
<td>—</td>
<td>4444</td>
</tr>
<tr>
<td>IPC16</td>
<td>00C4</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>—</td>
<td>—</td>
<td>4444</td>
</tr>
<tr>
<td>IPC17</td>
<td>00C6</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>—</td>
<td>—</td>
<td>4444</td>
</tr>
<tr>
<td>IPC18</td>
<td>00C8</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>—</td>
<td>—</td>
<td>4444</td>
</tr>
<tr>
<td>IPC19</td>
<td>00CA</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>—</td>
<td>—</td>
<td>4444</td>
</tr>
<tr>
<td>IPC20</td>
<td>00CC</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>—</td>
<td>—</td>
<td>4444</td>
</tr>
<tr>
<td>IPC21</td>
<td>00CE</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>—</td>
<td>—</td>
<td>4444</td>
</tr>
<tr>
<td>IPC22</td>
<td>00D0</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>—</td>
<td>—</td>
<td>4444</td>
</tr>
<tr>
<td>IPC23</td>
<td>00D2</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>—</td>
<td>—</td>
<td>4444</td>
</tr>
<tr>
<td>IPC24</td>
<td>00D4</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>—</td>
<td>—</td>
<td>4444</td>
</tr>
<tr>
<td>IPC25</td>
<td>00D6</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>—</td>
<td>—</td>
<td>4444</td>
</tr>
<tr>
<td>IPC26</td>
<td>00D8</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>—</td>
<td>—</td>
<td>4444</td>
</tr>
<tr>
<td>IPC27</td>
<td>00DA</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>—</td>
<td>—</td>
<td>4444</td>
</tr>
<tr>
<td>IPC28</td>
<td>00DC</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>—</td>
<td>—</td>
<td>4444</td>
</tr>
<tr>
<td>IPC29</td>
<td>00DE</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>—</td>
<td>—</td>
<td>4444</td>
</tr>
<tr>
<td>INTTREG</td>
<td>00E0</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>—</td>
<td>—</td>
<td>0000</td>
</tr>
</tbody>
</table>

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

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

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

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

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

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

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

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

<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:** For additional Application Notes and code examples for the PIC24H device family, visit the Microchip web site (www.microchip.com).
6.8 REVISION HISTORY

Revision A (February 2007)
This is the initial release of this document.

Revision B (May 2007)
Minor updates were made to this document.

Revision C (July 2008)
This revision incorporates the following updates:

- **Examples:**
  - The term “Programmable Low-Voltage Detect (PLVD)” in the example, in 6.1.5 “Interrupt Priority” has been corrected as “UART1 Rx Interrupt”.

- **Headings:**
  - Math Error Trap (Soft Trap, Level 11) has been updated as 6.2.1.2 “Math Error Trap (Soft Trap, Level 11)”.

- **Registers:**
  - The bit description for bit 6 and bit 4 in the INTCON1: Interrupt Control Register 1 (see Register 6-3) has been corrected.
  - The bit descriptions for bit 5, bit 6, bit 7, and bit 8 in the IEC1: Interrupt Enable Control Register 1 have been corrected (see Register 6-11).
  - The bit description for bit 2, bit 3, bit 4, and bit 14 in the IEC2: Interrupt Enable Control Register 2 have been corrected (see Register 6-12).
  - The bit descriptions for all the bits in the IEC3: Interrupt Enable Control Register 3 have been corrected (see Register 6-13).
  - The bit descriptions for all the bits in the IEC4: Interrupt Enable Control Register 4 have been corrected (see Register 6-14).
  - Added a new register “INTTREG: Interrupt Control and Status Register” (see Register 6-33).

- **Notes:**
  - Added a note after the first paragraph in 6.1.5 “Interrupt Priority”, which provides information on changing the interrupt priority levels “on-the-fly”.

- **Tables:**
  - Updated the IVT Address and AIVT Address for the IRQ numbers 83-124, in Table 6-1.

- Additional minor corrections such as language and formatting updates are incorporated throughout the document.