The dsPIC33CK256MP508 family devices that you have received conform functionally to the current Device Data Sheet (DS70005349H), except for the anomalies described in this document.

The silicon issues discussed in the following pages are for silicon revisions with the Device and Revision IDs listed in Table 1. The silicon issues are summarized in Table 2.

The errata described in this document will be addressed in future revisions of the dsPIC33CK256MP508 silicon. For example, to identify the silicon revision level using MPLAB IDE in conjunction with a hardware debugger:

1. Using the appropriate interface, connect the device to the hardware debugger.
2. Open an MPLAB IDE project.
3. Configure the MPLAB IDE project for the appropriate device and hardware debugger.
4. Based on the version of MPLAB IDE you are using, do one of the following:
   a) For MPLAB IDE 8, select Programmer > Reconnect.
   b) For MPLAB X IDE, select Window > Dashboard and click the Refresh Debug Tool Status icon ( ).
5. Depending on the development tool used, the part number and Device Revision ID value appear in the Output window.

The DEVREV values for the various dsPIC33CK256MP508 silicon revisions are shown in Table 1.

### TABLE 1: SILICON DEVREV VALUES

<table>
<thead>
<tr>
<th>Part Number</th>
<th>Device ID(1)</th>
<th>Revision ID for Silicon Revision(2)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A1</td>
<td>B2</td>
</tr>
<tr>
<td>dsPIC33CK256MP508 Family With CAN FD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dsPIC33CK256MP508</td>
<td>0x7C74</td>
<td></td>
</tr>
<tr>
<td>dsPIC33CK256MP506</td>
<td>0x7C73</td>
<td></td>
</tr>
<tr>
<td>dsPIC33CK256MP505</td>
<td>0x7C72</td>
<td></td>
</tr>
<tr>
<td>dsPIC33CK256MP503</td>
<td>0x7C71</td>
<td></td>
</tr>
<tr>
<td>dsPIC33CK256MP502</td>
<td>0x7C70</td>
<td></td>
</tr>
<tr>
<td>dsPIC33CK128MP508</td>
<td>0x7C64</td>
<td>0x0001</td>
</tr>
<tr>
<td>dsPIC33CK128MP506</td>
<td>0x7C63</td>
<td>0x0004</td>
</tr>
<tr>
<td>dsPIC33CK128MP505</td>
<td>0x7C62</td>
<td></td>
</tr>
<tr>
<td>dsPIC33CK128MP503</td>
<td>0x7C61</td>
<td></td>
</tr>
<tr>
<td>dsPIC33CK128MP502</td>
<td>0x7C60</td>
<td></td>
</tr>
</tbody>
</table>

Note 1: The Device IDs (DEVID and DEVREV) are located at the last two implemented addresses of configuration memory space. They are shown in hexadecimal in the format “DEVID DEVREV”.

Note 2: Refer to the “dsPIC33CK256MP508 Family Flash Programming Specification” (DS70005300) for detailed information on Device and Revision IDs for your specific device.
## TABLE 1: SILICON DEVREV VALUES (CONTINUED)

<table>
<thead>
<tr>
<th>Part Number</th>
<th>Device ID&lt;sup&gt;(1)&lt;/sup&gt;</th>
<th>Revision ID for Silicon Revision&lt;sup&gt;(2)&lt;/sup&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>A1</td>
</tr>
<tr>
<td><strong>dsPIC33CK256MP508 Family With CAN FD (Continued)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dsPIC33CK64MP508</td>
<td>0x7C54</td>
<td>0x0001</td>
</tr>
<tr>
<td>dsPIC33CK64MP506</td>
<td>0x7C53</td>
<td></td>
</tr>
<tr>
<td>dsPIC33CK64MP505</td>
<td>0x7C52</td>
<td></td>
</tr>
<tr>
<td>dsPIC33CK64MP503</td>
<td>0x7C51</td>
<td></td>
</tr>
<tr>
<td>dsPIC33CK64MP502</td>
<td>0x7C50</td>
<td></td>
</tr>
<tr>
<td>dsPIC33CK32MP506</td>
<td>0x7C43</td>
<td></td>
</tr>
<tr>
<td>dsPIC33CK32MP505</td>
<td>0x7C42</td>
<td></td>
</tr>
<tr>
<td>dsPIC33CK32MP503</td>
<td>0x7C41</td>
<td></td>
</tr>
<tr>
<td>dsPIC33CK32MP502</td>
<td>0x7C40</td>
<td></td>
</tr>
<tr>
<td><strong>dsPIC33CK256MP508 Family Without CAN FD</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dsPIC33CK256MP208</td>
<td>0x7C34</td>
<td></td>
</tr>
<tr>
<td>dsPIC33CK256MP206</td>
<td>0x7C33</td>
<td></td>
</tr>
<tr>
<td>dsPIC33CK256MP205</td>
<td>0x7C32</td>
<td></td>
</tr>
<tr>
<td>dsPIC33CK256MP203</td>
<td>0x7C31</td>
<td></td>
</tr>
<tr>
<td>dsPIC33CK256MP202</td>
<td>0x7C30</td>
<td></td>
</tr>
<tr>
<td>dsPIC33CK128MP208</td>
<td>0x7C24</td>
<td></td>
</tr>
<tr>
<td>dsPIC33CK128MP206</td>
<td>0x7C23</td>
<td></td>
</tr>
<tr>
<td>dsPIC33CK128MP205</td>
<td>0x7C22</td>
<td></td>
</tr>
<tr>
<td>dsPIC33CK128MP203</td>
<td>0x7C21</td>
<td></td>
</tr>
<tr>
<td>dsPIC33CK128MP202</td>
<td>0x7C20</td>
<td></td>
</tr>
<tr>
<td>dsPIC33CK64MP208</td>
<td>0x7C14</td>
<td></td>
</tr>
<tr>
<td>dsPIC33CK64MP206</td>
<td>0x7C13</td>
<td></td>
</tr>
<tr>
<td>dsPIC33CK64MP205</td>
<td>0x7C12</td>
<td></td>
</tr>
<tr>
<td>dsPIC33CK64MP203</td>
<td>0x7C11</td>
<td></td>
</tr>
<tr>
<td>dsPIC33CK64MP202</td>
<td>0x7C10</td>
<td></td>
</tr>
<tr>
<td>dsPIC33CK32MP206</td>
<td>0x7C03</td>
<td></td>
</tr>
<tr>
<td>dsPIC33CK32MP205</td>
<td>0x7C02</td>
<td></td>
</tr>
<tr>
<td>dsPIC33CK32MP203</td>
<td>0x7C01</td>
<td></td>
</tr>
<tr>
<td>dsPIC33CK32MP202</td>
<td>0x7C00</td>
<td></td>
</tr>
</tbody>
</table>

**Note 1:** The Device IDs (DEVID and DEVREV) are located at the last two implemented addresses of configuration memory space. They are shown in hexadecimal in the format “DEVID DEVREV”.

**Note 2:** Refer to the “dsPIC33CK256MP508 Family Flash Programming Specification” (DS70005300) for detailed information on Device and Revision IDs for your specific device.
## TABLE 2: SILICON ISSUE SUMMARY

<table>
<thead>
<tr>
<th>Module</th>
<th>Feature</th>
<th>Item Number</th>
<th>Issue Summary</th>
<th>Affected Revisions</th>
</tr>
</thead>
<tbody>
<tr>
<td>I^2C</td>
<td>Interrupt</td>
<td>1.</td>
<td>In Slave mode, incorrect interrupt is generated when DHEN = 1.</td>
<td>X</td>
</tr>
<tr>
<td>I^2C</td>
<td>Error</td>
<td>2.</td>
<td>Bus collision error cannot be cleared.</td>
<td>X</td>
</tr>
<tr>
<td>I^2C</td>
<td>Error</td>
<td>3.</td>
<td>False bus collision error generated.</td>
<td>X</td>
</tr>
<tr>
<td>I^2C</td>
<td>Idle</td>
<td>4.</td>
<td>Address cannot be received in Idle mode.</td>
<td>X</td>
</tr>
<tr>
<td>Oscillator</td>
<td>PLL</td>
<td>5.</td>
<td>FRCDIVN drives the PLL instead of the FRC.</td>
<td>X</td>
</tr>
<tr>
<td>Oscillator</td>
<td>HS, XT</td>
<td>6.</td>
<td>Removed.</td>
<td></td>
</tr>
<tr>
<td>PWM</td>
<td>Dead Time</td>
<td>7.</td>
<td>When feed-forward PCI is used for dead-time compensation (DTMPSEL = 1), the PWMx outputs are overridden.</td>
<td>X</td>
</tr>
<tr>
<td>UART</td>
<td>OERR</td>
<td>8.</td>
<td>The OERR bit cannot be cleared by software.</td>
<td>X</td>
</tr>
<tr>
<td>UART</td>
<td>FERR</td>
<td>9.</td>
<td>The FERR bit will not get set if one Stop bit is received.</td>
<td>X</td>
</tr>
<tr>
<td>UART</td>
<td>OERR</td>
<td>10.</td>
<td>The 9th byte received will not be available to be read.</td>
<td>X</td>
</tr>
<tr>
<td>UART</td>
<td>TRMT</td>
<td>11.</td>
<td>The TRMT bit takes time to set on the last transmit completion.</td>
<td>X</td>
</tr>
<tr>
<td>UART</td>
<td>TRMT</td>
<td>12.</td>
<td>The TRMT bit is unreliable when there is back-to-back Break character transmission.</td>
<td>X</td>
</tr>
<tr>
<td>UART</td>
<td>Idle</td>
<td>13.</td>
<td>The RIDLE bit takes one instruction cycle to get cleared after ABAUD is set.</td>
<td>X</td>
</tr>
<tr>
<td>UART</td>
<td>TXWRE</td>
<td>14.</td>
<td>The TXWRE bit (UxSTAH[7]) cannot be cleared once it gets set.</td>
<td>X</td>
</tr>
<tr>
<td>UART</td>
<td>Address Detect</td>
<td>15.</td>
<td>When writing to UxP1 with UTXBRK = 1, the content of P1 will not get transmitted.</td>
<td>X</td>
</tr>
<tr>
<td>UART</td>
<td>Address Detect</td>
<td>16.</td>
<td>In Address Detect mode, the content of P1 is not transmitted on writing to P1 with UTXBRK = 1.</td>
<td>X</td>
</tr>
<tr>
<td>UART</td>
<td>Sleep</td>
<td>17.</td>
<td>When waking from Sleep with a UART reception, SLIPN needs to be set in addition to WAKE = 1.</td>
<td>X</td>
</tr>
<tr>
<td>UART</td>
<td>Smart Card</td>
<td>18.</td>
<td>The Wait Time Counter Interrupt Flag (WTCIF) is set when the last character transmitted has the bit, LAST = 0.</td>
<td>X</td>
</tr>
<tr>
<td>UART</td>
<td>XOFF</td>
<td>19.</td>
<td>XOFF is transmitted when one empty space remains in the RX buffer.</td>
<td>X</td>
</tr>
<tr>
<td>MBIST</td>
<td>MBISTDONE</td>
<td>20.</td>
<td>After executing a Reset, the MBISTDONE bit will always be set.</td>
<td>X</td>
</tr>
<tr>
<td>CPU</td>
<td>FLIM Instruction</td>
<td>21.</td>
<td>When the operands are of different signs, the FLIM instruction may not force the correct data limit.</td>
<td>X</td>
</tr>
<tr>
<td>SCCP/ MCCP</td>
<td>Clock Source</td>
<td>22.</td>
<td>Using Fosc as the clock source may cause synchronization issues.</td>
<td>X</td>
</tr>
<tr>
<td>I^2C</td>
<td>SMBus 3.0</td>
<td>23.</td>
<td>When Configuration bit, SMBEN (FDEVOPT[10]) = 1, the SMBus 3.0 VIH minimum specification may not be met.</td>
<td>X</td>
</tr>
<tr>
<td>I^2C</td>
<td>POR</td>
<td>24.</td>
<td>Spike on I/O at POR.</td>
<td></td>
</tr>
<tr>
<td>CPU</td>
<td>DIV.SD</td>
<td>25.</td>
<td>Overflow bit is not getting set when an overflow occurs.</td>
<td>X</td>
</tr>
<tr>
<td>CPU</td>
<td>MAXAB/MINAB/ MINZAB Instructions</td>
<td>26.</td>
<td>MAXAB, MINAB and MINZAB do not work for different sign operands.</td>
<td>X</td>
</tr>
<tr>
<td>DMA</td>
<td>ADC Triggers</td>
<td>27.</td>
<td>DMA is triggered continuously from ADC.</td>
<td>X</td>
</tr>
<tr>
<td>PWM</td>
<td>Time Base Capture</td>
<td>28.</td>
<td>PWM Capture Status (CAP) flag will not set again under certain conditions.</td>
<td>X</td>
</tr>
<tr>
<td>I^2C</td>
<td>I^2C</td>
<td>29.</td>
<td>All instances of I^2C may exhibit errors and should not be used.</td>
<td>X</td>
</tr>
<tr>
<td>Oscillator</td>
<td>VCO and AVCO Dividers</td>
<td>30.</td>
<td>Main and auxiliary PLL external VCO dividers can fail to output the clock signal.</td>
<td>X</td>
</tr>
</tbody>
</table>
Silicon Errata Issues

Note: This document summarizes all silicon errata issues from all revisions of silicon, previous as well as current. Only the issues indicated by the shaded column in the following tables apply to the current silicon revision (B2).

1. Module: I²C
   In Slave mode with DHEN = 1 (Data Hold Enable), if software sends a NACK, a Slave interrupt is asserted at the 9th falling edge of the clock.
   **Work around**
   Software should ignore the Slave interrupt that is asserted after sending a NACK.

   **Affected Silicon Revisions**
<table>
<thead>
<tr>
<th>A1</th>
<th>B2</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

2. Module: I²C
   In Slave mode, the Bus Collision Detect bit (BCL) cannot be cleared when bus collision detection is enabled (SBCDE = 1).
   **Work around**
   Disable the I²C module and then re-enable the module.

   **Affected Silicon Revisions**
<table>
<thead>
<tr>
<th>A1</th>
<th>B2</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

3. Module: I²C
   In Slave mode, false bus collision triggers are generated when the bus collision is enabled (SBCDE = 1) and a Stop bit is received.
   **Work around**
   Ignore the bus collision. Disable the I²C module and then re-enable the module.

   **Affected Silicon Revisions**
<table>
<thead>
<tr>
<th>A1</th>
<th>B2</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

4. Module: I²C
   In Slave mode, an address cannot be received when the device is in Idle and the module is set for discontinue in Idle (I2CSIDL = 1).
   **Work around**
   None.

   **Affected Silicon Revisions**
<table>
<thead>
<tr>
<th>A1</th>
<th>B2</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

5. Module: Oscillator
   When using the 8 MHz internal FRC Oscillator with Primary PLL as either a system clock or a peripheral source, FRCDIVN drives the PLL instead of the FRC.
   This means that the PLL FRC input selection is subject to the FRCDIV[2:0] bits and could lead to a condition where the minimum PLL input requirement of 8 MHz is not maintained.
   **Work around**
   Ensure FRCDIV[2:0] bits are maintained as zero when using FRCPLL as either a system clock or a peripheral source.

   **Affected Silicon Revisions**
<table>
<thead>
<tr>
<th>A1</th>
<th>B2</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

6. Module: Oscillator
   This errata is no longer applicable to any silicon revisions of this product. See Section 2.5 External Oscillator Pins in the current device data sheet (DS70005349H) for guidance on oscillator design to avoid start-up related issues.

7. Module: PWM
   When feed-forward PCI is used for dead-time compensation (DTCMPSEL = 1), the PWMx outputs are overridden.
   **Work around**
   Use Sync PCI (DTCMPSEL = 0) for dead-time compensation.

   **Affected Silicon Revisions**
<table>
<thead>
<tr>
<th>A1</th>
<th>B2</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
8. Module: UART

Once the UART receive buffer overflows and the OERR bit (UxSTA[1]) is set, the OERR bit cannot be cleared by software.

**Work around**
1. Make sure that the receive buffer never overflows. Do not let the OERR bit get set by reading the received data byte on each byte reception.
2. Disable and enable UART before clearing the OERR bit.

**Affected Silicon Revisions**

<table>
<thead>
<tr>
<th>A1</th>
<th>B2</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

9. Module: UART

When the UART is operating with STSEL[1:0] = 2 (two Stop bits sent, two checked at receive), the FERR bit will not get set if one Stop bit is received.

**Work around**
3. Use STSELx = 3 instead of STSELx = 2. When operating with STSELx = 3 mode, the UART will be configured to send two Stop bits, but check one at receive.

**Affected Silicon Revisions**

<table>
<thead>
<tr>
<th>A1</th>
<th>B2</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

10. Module: UART

When the receive buffer overflows, the 9th byte received will get lost and cannot be read.

**Work around**
Do not allow the OERR bit to get set by reading the received data byte on each byte reception.

**Affected Silicon Revisions**

<table>
<thead>
<tr>
<th>A1</th>
<th>B2</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

11. Module: UART

At low BRG value, the TRMT bit takes time to set on the last transmit completion, which may result in the transmitted data getting lost.

**Work around**
1. Use the UTXBE bit to monitor for the next transmit.
2. Provide a delay to stabilize the POSC.

**Affected Silicon Revisions**

<table>
<thead>
<tr>
<th>A1</th>
<th>B2</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

12. Module: UART

The Transmit Shifter Empty (TRMT) bit is unreliable when there is back-to-back Break character transmission.

**Work around**
Poll the UART Transmit Break bit, UTXBRK (UxMODE[8]), to be cleared instead of the TRMT bit.

**Affected Silicon Revisions**

<table>
<thead>
<tr>
<th>A1</th>
<th>B2</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

13. Module: UART

During the UART Auto-Baud Detection sequence, the RIDLE bit takes one instruction cycle to get cleared after ABAUD is set.

**Work around**
Ignore the RIDLE bit until the Auto-Baud Detection sequence is complete.

**Affected Silicon Revisions**

<table>
<thead>
<tr>
<th>A1</th>
<th>B2</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
14. Module: UART

Once the TX Write Transmit Error Status bit, TXWRE (UxSTAH[7]), gets set, the TXWRE cannot be cleared by a single clear instruction.

**Work around**

Use multiple clear instructions in a loop until the TXWRE bit gets cleared.

**Affected Silicon Revisions**

| A1 | B2 | X | X |

15. Module: UART

In UART Address Detect mode, writing to UxP1 with UTXBRK = 1 should cause a Break to be transmitted, followed by the content in P1, but the content of P1 will not get transmitted.

**Work around**

After writing to P1, wait for UTXBRK to get clear and then rewrite to P1.

**Affected Silicon Revisions**

| A1 | B2 | X | X |

16. Module: UART

In Address Detect mode, the content of P1 is not transmitted on writing to P1 with UTXBRK = 1.

**Work around**

Write P1 a second time after waiting for the Break transmission to start.

**Affected Silicon Revisions**

| A1 | B2 | X | X |

17. Module: UART

When waking from Sleep with a UART reception, SLPEN needs to be set in addition to WAKE = 1.

**Work around**

Set the SPLEN bit in addition to WAKE before entering Sleep.

**Affected Silicon Revisions**

| A1 | B2 | X | X |

18. Module: UART

In Smart Card T = 1 mode, the Wait Time Counter Interrupt Flag (WTClF) is set when the last character transmitted has the bit, LAST = 0.

**Work around**

Ignore WTC interrupt events on non-last bytes.

**Affected Silicon Revisions**

| A1 | B2 | X | X |

19. Module: UART

In Software Flow Control mode, XOFF is transmitted when one empty space remains in the RX buffer. XOFF transmission can get further delayed if the transmitter has already been loaded, resulting in XOFF transmission on a receive buffer full event.

**Work around 1**

Give a minimum one-byte delay before each byte transmission.

**Work around 2**

Use the UART RX interrupt with URXISEL[2:0] set to at least two empty slots. This allows the RX buffer to be read in time to prevent RX buffer overflow.

**Affected Silicon Revisions**

| A1 | B2 | X | X |

20. Module: MBIST

After a Reset, the MBISTDONE status bit will be set regardless of a BIST test being executed. If a BIST is requested and executed, the MBISTDONE bit will be set as expected.

**Work around**

None.

**Affected Silicon Revisions**

| A1 | B2 | X | X |
21. Module: CPU

The FLIM instruction may incorrectly limit the data range when operating on signed operands of different sign values. If the operands are either all negative or all positive, the limit is correct.

Work around
None.

Affected Silicon Revisions

| A1 | B2 | X | X |

22. Module: SCCP/MCCP

When Fosc is selected as the clock source using the CLKSEL[2:0] bits (CCPxCON1L[10:8]), unexpected operation may occur. For proper SCCP/MCCP input clock synchronization, do not use Fosc as the system clock source.

Work around
Use any of the other available clock sources in CLKSEL[2:0].

Affected Silicon Revisions

| A1 | B2 | X | X |

23. Module: I²C

When selecting SMBus 3.0 operation using Configuration bit, SMBEN (FDEVOPT[10]), the Voltage Input High (Vih) of the SMBus 3.0 specification minimum may not be met.

Work around
None.

Affected Silicon Revisions

| A1 | B2 | X | X |

24. Module: I/O

During a fast device power-up when the Vdd ramp is less than 4 mS, the I/O pins may drive up to 100 µA current for a duration of up to 10 µS (Figure 1-1).

FIGURE 1-1: I/O RAMP

Work around
1. Slow down the Vdd ramp time (greater than 4 mS for Vdd to ramp 0V to 3.3V).
2. Ensure the circuitry that is connected to the pins can endure this pulse.

Example applications affected may include complementary power switches, where a transient current shoot-through might occur. High-voltage applications with complementary switches should power the high-voltage 200 µSec later than powering the dsPIC® device to avoid the current shoot-through. This behavior is specific to each device and not affected by aging.

Affected Silicon Revisions

| A1 | B2 | X | X |

25. Module: CPU

When using the Signed 32-by-16-bit Division instruction, DIV.SD, the Overflow bit may not always get set when an overflow occurs.

Work around
Test for and handle overflow conditions outside of the div.sd instruction.

Affected Silicon Revisions

| A1 | B2 | X | X |
26. Module: CPU

When operating on signed operands of different sign values, the output for MAXAB, MINAB and MINZAB instructions may be incorrect. If the operands are either all negative or all positive, the output is correct.

Work around
None.

Affected Silicon Revisions

<table>
<thead>
<tr>
<th>A1</th>
<th>B2</th>
<th>X</th>
</tr>
</thead>
</table>

27. Module: DMA

The DMA receives multiple continuous triggers from the ADC until the trigger event from the ADC is cleared. The OVRUNIF flag (DMAINTn[3]) will be set. When the OVRUNIF bit changes state from '0' to '1', a DMA interrupt is generated.

Work around
Ignore the OVRUNIF bit and the first DMA interrupt. Clear the ADC trigger source (ANxRDY) with a DMA read of the ADC buffer, ADCBUFx, for the corresponding ADC channel.

Affected Silicon Revisions

<table>
<thead>
<tr>
<th>A1</th>
<th>B2</th>
<th>X</th>
</tr>
</thead>
</table>

28. Module: PWM

When using a PWM Control Input (PCI) to trigger a time base capture, the Capture Status flag, CAP (PGxSTAT[5]), may not set again under certain conditions. When a subsequent PWM capture event occurs while, or just after, reading the current capture value from the PGxCAP register, the Capture Status Flag, CAP, will not set again.

Work around
Read the PWM Generator Capture (PGxCAP, x = 1 to 8) register at a known time to avoid the condition. The timing of the PGxCAP read operation can be scheduled by using PWM Generator x (1-8) interrupt or any of the six PWM Event (A-F) interrupts corresponding to the PCI event which triggered the time base capture. Read the PGxCAP value after the CAP bit has set within the interrupt.

Affected Silicon Revisions

<table>
<thead>
<tr>
<th>A1</th>
<th>B2</th>
<th>X</th>
</tr>
</thead>
</table>

29. Module: I2C

All instances of I2C/SMBus may exhibit errors and should not be used. When operating I2C/SMBus in a noisy environment, the I2C module may exhibit various errors. These errors may include, but are not limited to, corrupted data, unintended interrupts or the I2C bus getting hung up due to injected noise. Examples of system noise include, but are not limited to, PWM outputs or other pins toggled at high speed adjacent to the I2C pins. Both Master and Slave I2C/SMBus modes may exhibit this issue.

Work around
If I2C is required, use a software I2C implementation. An example I2C software library is available from Microchip:
www.microchip.com/dsPIC33C_I2C_SoftwareLibrary

Affected Silicon Revisions

<table>
<thead>
<tr>
<th>A1</th>
<th>B2</th>
<th>X</th>
</tr>
</thead>
</table>

30. Module: Oscillator

At PLL start-up, the main and auxiliary PLL VCO dividers may occasionally halt and not provide a clock output. The VCO and AVCO dividers can be selected as clock sources for different peripheral modules, including the ADC, PWM, DAC, CAN FD, UART, etc. VCO and AVCO divider outputs, FVCO/2, FVCO/3, FVCO/4, FVCO/5, AFVCO/2, AFVCO/3, AFVCO/4 and AFVCO/5 outputs are affected.

Work around
1. Use another clock source, such as the FOSC, PLL or APPL output (FPLLLO and AFPOLLO) instead of the VCO or AVCO dividers.
2. If the application requires the VCO/AVCO divider, test the clock source before using the peripheral in the end application. System resources, including a timer, I/O pin state or interrupts, can be used to detect and verify peripheral activity for presence of the VCO divider clock output. Any type of Reset may recover the VCO divider clock (Software Reset, WDT, MCLR or POR).

Affected Silicon Revisions

<table>
<thead>
<tr>
<th>A1</th>
<th>B2</th>
<th>X</th>
</tr>
</thead>
</table>
Data Sheet Clarifications

The following typographic corrections and clarifications are to be noted for the latest version of the device data sheet (DS70005349H):

<table>
<thead>
<tr>
<th>Note</th>
<th>Corrections are shown in <strong>bold</strong>. Where possible, the original bold text formatting has been removed for clarity.</th>
</tr>
</thead>
</table>

None.
APPENDIX A: DOCUMENT 
REVISION HISTORY 

Rev A Document (5/2018) 
Initial release of this document; issued for revision A1. 

Rev B Document (9/2018) 
Added silicon issues 21 (CPU), 22 (SCCP/MCCP) and 23 (I²C). 

Rev C Document (12/2018) 
Added silicon issues 24 (I/O), 25 (CPU), 26 (CPU) and 27 (CPU). 

Added silicon issues 28 (DMA) and 29 (PWM). 
Updated reference to current Device Data Sheet revision (DS70005349F). 

Rev E Document (10/2019) 
Updated silicon issue 24 (I/O) and silicon issue 28 (PWM). 
Removed silicon issue 27 (CPU) which stated that the upper byte of the destination register may not be persistent. 
Updated reference to current Device Data Sheet revision (DS70005349G). 

Rev F Document (12/2019) 
Added silicon issue 29 (I²C). 

Rev G Document (6/2020) 
Added silicon issue 30 (Oscillator). 
Added data sheet clarification (Document Revision History). 
Removes silicon issue 6 (Oscillator) since it is no longer applicable. 

Rev H Document (7/2020) 
Added silicon revision B2. 
Updated the wording in silicon issue 29 (I²C). 
Removed data sheet clarification (Electrical Characteristics) since it was addressed in the latest device data sheet.
Note the following details of the code protection feature on Microchip devices:

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

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

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

Trademarks

The Microchip name and logo, the Microchip logo, Adaptec, AnyRate, AVR, AVR logo, AVR Freaks, BesTime, BitCloud, chipKIT, chipKIT logo, CryptoMemory, CryptoRF, dsPIC, FlashFlex, flexPWR, HELDO, IGLOO, JukeBox, KeeLoq, Kleer, LANcheck, LinkMD, maXStylus, maXTouch, MediaLB, megaAVR, Microsemi, Microsemi logo, MOST, MOST logo, MPLAB, OptoLyzer, PackeTime, PIC, picoPower, PICSTART, PIC32 logo, PolarFire, Prochip Designer, QTough, SAM-Ba, SaGenusity, SpyNIC, SST, SST Logo, SuperFlash, Symmecromic, SyncServer, Tachyon, TempTrackr, TimeSource, tinyAVR, UNI/O, Vextron, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

APT, ClockWorks, The Embedded Control Solutions Company, EtherSynch, FlashTec, Hyper Speed Control, HyperLight Load, IntelilMOS, Libero, motorBench, mTouch, Powermate 3, Precision Edge, ProASIC, ProASIC Plus, ProASIC Plus logo, Quiet-Wire, SmartFusion, SyncWorld, Temux, TimeCesium, TimeHub, TimePictra, TimeProvider, Vite, WinPath, and ZL are registered trademarks of Microchip Technology Incorporated in the U.S.A.


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

The Adaptec logo, Frequency on Demand, Silicon Storage Technology, and Symmcom are registered trademarks of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

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

© 2018-2020, Microchip Technology Incorporated, All Rights Reserved.

ISBN: 978-1-5224-6346-7

For information regarding Microchip’s Quality Management Systems, please visit www.microchip.com/quality.
## World Wide Sales and Service

### AMERICAS

**Corporate Office**
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277

**Technical Support:**
http://www.microchip.com/support

**Web Address:**
www.microchip.com

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

- **Austin, TX**
  - Tel: 512-257-3370

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

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

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

- **Detroit**
  - Novi, MI
  - Tel: 248-848-4000

- **Houston, TX**
  - Tel: 281-894-5983

- **Indianapolis**
  - Noblesville, IN
  - Tel: 317-773-8323
  - Fax: 317-773-5453
  - Tel: 317-536-2380

- **Los Angeles**
  - Mission Viejo, CA
  - Tel: 949-462-9523
  - Fax: 949-462-9608
  - Tel: 951-273-7800

- **Raleigh, NC**
  - Tel: 919-844-7510

- **New York, NY**
  - Tel: 631-435-6000

- **San Jose, CA**
  - Tel: 408-735-9110
  - Tel: 408-436-4270

- **Canada - Toronto**
  - Tel: 905-695-1980
  - Fax: 905-695-2078

### ASIA/PACIFIC

- **Australia - Sydney**
  - Tel: 61-2-9888-6733

- **China - Beijing**
  - Tel: 86-10-8569-7000

- **China - Chengdu**
  - Tel: 86-28-8665-5511

- **China - Chongqing**
  - Tel: 86-23-8980-9588

- **China - Dongguan**
  - Tel: 86-769-8702-9880

- **China - Guangzhou**
  - Tel: 86-20-8755-8029

- **China - Hangzhou**
  - Tel: 86-571-8792-8115

- **China - Hong Kong SAR**
  - Tel: 852-2943-5100

- **China - Nanjing**
  - Tel: 86-25-8755-8029

- **China - Qingdao**
  - Tel: 86-532-8502-7355

- **China - Shanghai**
  - Tel: 86-21-3326-8000

- **China - Shenyang**
  - Tel: 86-24-2334-2829

- **China - Shenzhen**
  - Tel: 86-755-8864-2200

- **China - Suzhou**
  - Tel: 86-186-6233-1526

- **China - Wuhan**
  - Tel: 86-27-5980-5300

- **China - Xiamen**
  - Tel: 86-3-8833-7252

- **China - Zhuhai**
  - Tel: 86-755-8864-2200

### ASIA/PACIFIC

- **India - Bangalore**
  - Tel: 91-80-3090-4444

- **India - New Delhi**
  - Tel: 91-11-4160-8631

- **India - Pune**
  - Tel: 91-20-4121-0141

- **Japan - Osaka**
  - Tel: 81-6-6152-7160

- **Japan - Tokyo**
  - Tel: 81-3-6880-3770

- **Korea - Daegu**
  - Tel: 82-53-744-4301

- **Korea - Seoul**
  - Tel: 82-2-554-7200

- **Malaysia - Kuala Lumpur**
  - Tel: 60-3-7651-7906

- **Malaysia - Penang**
  - Tel: 60-4-227-8870

- **Philippines - Manila**
  - Tel: 63-2-634-9065

- **Singapore**
  - Tel: 65-6334-8870

- **Taiwan - Hsin Chu**
  - Tel: 886-3-577-8366

- **Taiwan - Kaohsiung**
  - Tel: 886-7-213-7830

- **Taiwan - Taipei**
  - Tel: 886-2-2508-8600

- **Thailand - Bangkok**
  - Tel: 66-2-694-1351

- **Vietnam - Ho Chi Minh**
  - Tel: 84-8-5446-2100

### EUROPE

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

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

- **Finland - Espoo**
  - Tel: 589-9-4520-820

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

- **Germany - Garching**
  - Tel: 94-8931-9700

- **Germany - Haan**
  - Tel: 49-2129-3766400

- **Germany - Heilbronn**
  - Tel: 49-7131-72400

- **Germany - Karlsruhe**
  - Tel: 49-71-625370

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

- **Germany - Rosenheim**
  - Tel: 49-8031-354-560

- **Israel - R’a’anana**
  - Tel: 972-9-744-7705

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

- **Italy - Padova**
  - Tel: 39-049-7625286

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

- **Norway - Trondheim**
  - Tel: 47-7288-4388

- **Poland - Warsaw**
  - Tel: 48-22-3325737

- **Romania - Bucharest**
  - Tel: 40-21-407-87-50

- **Spain - Madrid**
  - Tel: 34-91-708-09-90
  - Fax: 34-91-708-09-91

- **Sweden - Gothenberg**
  - Tel: 46-31-704-60-40

- **Sweden - Stockholm**
  - Tel: 46-8-5090-4654

- **UK - Wokingham**
  - Tel: 44-118-921-5800
  - Fax: 44-118-921-5820