dsPIC33EPXXXGS70X/80X Family
Silicon Errata and Data Sheet Clarification

The dsPIC33EPXXXGS70X/80X family devices that you have received conform functionally to the current Device Data Sheet (DS70005258**C**), 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 dsPIC33EPXXXGS70X/80X silicon.

> **Note:** This document summarizes all silicon errata issues from all revisions of silicon, previous as well as current. Only the issues indicated in the last column of Table 2 apply to the current silicon revision (**B0**).

Data Sheet clarifications and corrections start on page 15, following the discussion of silicon issues.

The silicon revision level can be identified using the current version of MPLAB® IDE and Microchip's programmers, debuggers, and emulation tools, which are available at the Microchip corporate website (www.microchip.com).

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.

> **Note:** If you are unable to extract the silicon revision level, please contact your local Microchip sales office for assistance.

The DEVREV values for the various dsPIC33EPXXXGS70X/80X silicon revisions are shown in Table 1.

TABLE 1: SILICON DEVREV VALUES

Part Number	Device ID[(1)]	Revision ID for Silicon Revision[(2)]		Part Number	Device ID[(1)]	Revision ID for Silicon Revision[(2)]	
		A2	B0			A2	B0
dsPIC33EP64GS708	0x6C03	0x4001	0x4002	dsPIC33EP128GS705	0x6C30	0x4001	0x4002
dsPIC33EP64GS804	0x6C40			dsPIC33EP128GS706	0x6C12		
dsPIC33EP64GS805	0x6C60			dsPIC33EP128GS708	0x6C13		
dsPIC33EP64GS806	0x6C42			dsPIC33EP128GS804	0x6C50		
dsPIC33EP64GS808	0x6C43			dsPIC33EP128GS805	0x6C70		
dsPIC33EP128GS702	0x6C11			dsPIC33EP128GS806	0x6C52		
dsPIC33EP128GS704	0x6C10			dsPIC33EP128GS808	0x6C53		

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".

2: Refer to the *"dsPIC33EPXXXGS70X/80X Family Flash Programming Specification"* (DS70005256) for detailed information on Device and Revision IDs for your specific device.

TABLE 2: SILICON ISSUE SUMMARY

Module	Feature	Item Number	Issue Summary	Affected Revisions(1)	
				A2	B0
CPU	`div.sd`	1.	When using the signed 32-by-16-bit division instruction, `div.sd`, the Overflow bit is not getting set when an overflow occurs.	X	X
CPU	Variable Interrupt Latency	2.	When Variable Interrupt Latency is selected (VAR = `1`), an address error trap or incorrect application behavior may occur.	X	X
CPU	`DO` Loop	3.	PSV access, including Table Reads or Writes in the last instruction of a `DO` loop, is not allowed.	X	X
ADC	ADC Sampling	4.	Under specific conditions, multi-core ADC sampling crosstalk noise might be present.	X	X
ADC	Oversampling Filter	5.	Parallel operation of the two oversampling filters results in missing data writes to the lower priority Filter Data Result register.	X	X
ADC	DNL	6.	DNL is out of specification at the mid-code boundary in Single-Ended mode.	X	X
UART	Break Character Generation	7.	The Transmit Shift Register Empty (TRMT) bit is unreliable when there are back-to-back Break character transmissions.	X	X
I^2C	Slave Mode	8.	The 7-bit address that matches the 10-bit upper address value (`111_10xx`) can never be accepted, regardless of the STRICT bit.	X	
I^2C	Slave Mode	9.	When data hold is enabled and software sends a NACK, a slave interrupt is occurring after the 9th clock.	X	X
SPI	Audio	10.	At start-up, the Idle state of the SDOx pin depends on the MSB of the first data packet when the slave is configured for Left/Right Justified mode.	X	X
SPI	Audio	11.	In PCM/DSP mode, if the channel width is greater than data width, the data loaded is not transmitted as per the loading sequence.	X	X
SPI	Audio	12.	In PCM/DSP mode, if the channel width is greater than data width, the state of SDOx is latched to the LSB of the data transmitted.	X	X
PTG	Debug	13.	Single-stepping of the command sequence queue when device is in Debug mode is not functional.	X	X
PTG	`PTGADD`/ `PTGCOPY`	14.	`PTGADD` and `PTGCOPY` commands do not change the counter limit values.	X	X
PTG	Software Trigger	15.	Software trigger (PTGSWT) is not cleared by hardware.	X	X
PWM	Module Enable	16.	A glitch may be observed on the PWM pins when the PWM module is enabled after assignment of pin ownership to the PWM module.	X	X
PWM	Center-Aligned Complementary	17.	Dead time between transitions of the PWMxH and PWMxL outputs may not be asserted when Swap mode is disabled.	X	X
PWM	Master Time Base Mode	18.	Changes to the PHASEx register after enabling the PWM module may result in abnormal PWM switching waveforms.	X	X
PWM	Push-Pull Mode	19.	When EIPU = `1`, Period register writes may produce back-to-back pulses under certain conditions.	X	X

Note 1: Only those issues indicated in the last column apply to the current silicon revision.

TABLE 2: SILICON ISSUE SUMMARY (CONTINUED)

Module	Feature	Item Number	Issue Summary	Affected Revisions[1]	
				A2	B0
PWM	Redundant/ Push-Pull Output Mode	20.	Changing the duty cycle value from a non-zero value to zero will produce a glitch pulse equal to one PWM clock.	X	X
PWM	Trigger Compare Match	21.	The first PWM/ADC trigger event on a TRIGx/STRIGx match may not occur under certain conditions.	X	X
I/O	5V Tolerant	22.	Limited number of I/O pins support 5V operation.	X	X
I/O	5V Tolerant	23.	Limit input current to I/O pins that support 5V operation.	X	X
Auxiliary PLL	APLL Lock	24.	The APLL lock bit is asserted directly after enabling the APLL.	X	X
I^{2}C	Slave Mode	25.	In Slave mode, false bus collision triggers are generated during reception of a Stop bit when bus collision is enabled (SBCDE = 1).	X	
I^{2}C	Slave Mode	26.	In Slave mode, the Bus Collision bit (BCL) cannot be cleared when bus collision detection is enabled (SBCDE = 1).	X	
CPU	Data Flash Reads	27.	Given a specific set of preconditions, when two or more data Flash read instructions (via Program Space Visibility (PSV) read or Table Read) are executed back-to-back, one or more subsequent instructions will be misexecuted.	X	X
PWM	Push-Pull Mode	28.	PWM generators may exhibit an incorrect phase relationship when configured in Push-Pull mode.	X	X
I/O	Schmitt Trigger	29.	Schmitt Trigger output may produce glitches.	X	
I^{2}C	I/O Voltage Threshold	30.	With SMBus disabled, the I^{2}C V_{IL} (Max) threshold may be lower than 0.3 V_{DD}.	X	X
PLL	Lock	31.	PLL out of specification during clock switch.	X	X

Note 1: Only those issues indicated in the last column apply to the current silicon revision.

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 (**A2**).

1. 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.

This erratum only affects operations in which at least one of the following conditions is true:

a) Dividend and divisor differ in sign,

b) Dividend > 0x3FFFFFFF or

c) Dividend < 0xC0000000.

Work around

The application software must perform both the following actions in order to handle possible undetected overflow conditions:

a) The value of the dividend must always be constrained to be in the following range: 0xC0000000 ≤ Dividend ≤ 0x3FFFFFFF.

b) If the dividend and divisor differ in sign (e.g., dividend is negative and divisor is positive), then after executing the `div.sd` instruction or the compiler built-in function, `__builtin_divsd()`, inspect the sign of the resultant quotient.

If the quotient is found to be a positive number, then treat it as an overflow condition.

Affected Silicon Revisions

A2	B0						
X	X						

2. Module: CPU

An address error trap or incorrect application behavior may occur if the variable exception processing latency is enabled by setting the VAR bit (CORCON[15] = `1`).

Work around

Enable the Fixed Interrupt Latency mode by clearing the VAR bit (CORCON[15] = `0`).

Affected Silicon Revisions

A2	B0						
X	X						

3. Module: CPU

Table Write (`TBLWTL`, `TBLWTH`) instructions cannot be the first or last instruction of a `DO` loop.

Work around

None.

Affected Silicon Revisions

A2	B0						
X	X						

4. Module: ADC

When using multiple ADC cores, if one of the ADC cores completes conversion while the other ADC cores are still converting, the data in the ADC cores which are converting may be randomly corrupted.

Work around for Revision A2

Work around 1: When using multiple ADC cores, the ADC triggers must be sufficiently staggered in time to ensure that the end of conversion of one or more cores doesn't occur during the conversion process of other cores.

Work around 2: For simultaneous conversion requirements, make sure the following conditions are met:

1. All the ADC cores for simultaneous conversion should have the same configurations.
2. Avoid shared ADC core conversion with any of the dedicated ADC cores; they can be sequential.
3. The trigger to initiate ADC conversion should be from the same source and at the same time.

Work around for Revision B0

Revision B0 implements an additional bit, NRE (ADCON1[7]). By setting the NRE bit, the end of conversions between the ADC cores are automatically staggered in the hardware. This eliminates the random result corruption issue.

Affected Silicon Revisions

A2	B0						
X	X						

5. Module: ADC

If both oversampling filters are configured identically and have different input channel selections (FLCHSEL[4:0] bits in the ADFLxCON register) and the channels start conversion from the same trigger source, then the lower priority filter result will not be written to the Filter Output Data register. Both data registers will contain the result from the higher priority filter (i.e., ADFL0DAT).

Work around

Ensure oversampling filters have input channels with different trigger sources and do not complete conversions simultaneously.

Affected Silicon Revisions

A2	B0						
X	X						

6. Module: ADC

When the ADC SAR core is configured to operate in Single-Ended mode, the core's DNL performance may be out of specification at the mid-code boundary.

Work around

None.

Affected Silicon Revisions

A2	B0						
X	X						

7. Module: UART

The Transmit Shift Register Empty (TRMT) bit is unreliable when there are back-to-back Break character transmissions.

For back-to-back Break characters, the TRMT bit may not reflect the actual status. If user software is polling for this bit to be set, it may result in dummy bytes getting transmitted instead of Break characters.

Work around

Poll the UARTx Transmit Break bit, UTXBRK (UxSTA[11]), to be cleared instead of the TRMT bit (UxSTA[8]) to be set. The UTXBRK status bit will be cleared after a Break character transmission.

Affected Silicon Revisions

A2	B0						
X	X						

8. Module: I^{2}C

In 7-Bit Addressing mode, the I^{2}C module will not respond to the 7-bit address that matches with the upper byte of the 10-bit address.

If the 7-bit address matches with the upper byte of the 10-bit address, the I^{2}C module will send a NACK, irrespective of the STRICT bit setting.

Work around

None.

Affected Silicon Revisions

A2	B0						
X	X						

9. Module: I^{2}C

In Slave mode with DHEN = 1 (Data Hold Enable), if software sends a NACK, the 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

A2	B0						
X	X						

10. Module: SPI

When the SPI is a slave and configured for Left Justified or Right Justified mode at start-up, the Idle state of the SDOx pin will be configured by the Most Significant bit (MSb) of the first data packet loaded to the buffer.

For example:

If data loaded to the buffer is 0xA5A5 (where MSb is '1'), then at start-up, the Idle state of SDOx will be HIGH.

If data loaded to the buffer is 0x5A5A (where MSb is '0'), then at start-up, the Idle state of SDOx will be LOW.

Work around

None.

Affected Silicon Revisions

A2	B0						
X	X						

11. Module: SPI

When the SPI is configured for PCM/DSP mode and MODE[32,16] = 1 or MODE[32,16] = 3, the data loaded is not transmitted as per sequence. For example, if the data is loaded as 0x8880, 0x8881, 0x8882, 0x8883, …, 0x8887, the transmitted data may be 0x8880, 0x8881, 0x8883, 0x8884, 0x8886, 0x8887.

Work around

None.

Affected Silicon Revisions

A2	B0						
X	X						

12. Module: SPI

When the SPI is configured for PCM/DSP mode and MODE[32,16] = 1 or MODE[32,16] = 3, after completion of transmission of each data byte, the SDOx pin state is the same as the LSb of the data.

For example:

If data was 0x8880 (where LSb is '0'), the state of SDOx will be LOW after transmission.

If data was 0x8881 (where LSb is '1'), the state of SDOx will be HIGH after transmission.

Work around

None.

Affected Silicon Revisions

A2	B0						
X	X						

13. Module: PTG

In Debug mode, the following features do not work:

- Single-stepping through the command sequence queue, which is enabled by setting PTGSSEN (PTGCST[9]).
- Step interrupt, which is generated after each step execution in the queue.

Work around

None.

Affected Silicon Revisions

A2	B0						
X	X						

14. Module: PTG

PTGADD/PTGCOPY commands, which add the contents of PTGADJ, or copy the content of PTGHOLD to the Loop Counter Limit registers (PTGC0LIM and PTGC1LIM) do not work.

Although the loop counter value gets updated, the loop does not repeat as required.

Work around

None.

Affected Silicon Revisions

A2	B0						
X	X						

15. Module: PTG

The PTG Software Trigger bit, PTGSWT (PTGCST[10]), is not automatically cleared by hardware upon completion of the PTGCTRL command.

Work around

Manually clear the PTGSWT bit (PTGCST[10]) after execution of the PTG software trigger.

Affected Silicon Revisions

A2	B0						
X	X						

16. Module: PWM

The PENH and PENL bits in the IOCONx register are used to assign ownership of the pins to either the PWM module or the GPIO module. The correct procedure to configure the PWM module is to first assign pin ownership to the PWM module and then enable it using the PTEN bit in the PTCON register.

If the PWM module is enabled using the above sequence, then a glitch may be observed on the PWM pins before the actual switching of the PWM outputs begins. This glitch may cause a momentary turn-on of power MOSFETs that are driven by the PWM pins and may cause damage to the application hardware.

Work around

Perform the following steps to avoid any glitches from appearing on the PWM outputs at the time of enabling:

1. Configure the respective PWM pins to digital inputs using the TRISx registers. This step will put the PWM pins in a high-impedance state. The PWM outputs must be maintained in a safe state by using pull-up or pull-down resistors.
2. Assign pin ownership to the GPIO module by configuring the PENH bit (IOCONx[15] = 0) and the PENL bit (IOCONx[14] = 0).
3. Specify the PWM override state to the desired safe state for the PWM pins using the OVRDAT[1:0] bits field in the IOCONx register.
4. Override the PWM outputs by setting the OVRENH bit (IOCONx[9]) = 1 and the OVRENL bit (IOCONx[8]) = 1.
5. Enable the PWM module by setting the PTEN bit (PTCON[15]) = 1.
6. Remove the PWM overrides by making the OVRENH bit (IOCONx[9]) = 0 and the OVRENL bit (IOCONx[8]) = 0.
7. Ensure a delay of at least one full PWM cycle.
8. Assign pin ownership to the PWM module by setting the PENH bit (IOCONx[15]) = 1 and the PENL bit (IOCONx[14]) = 1.

The code in Example 1 illustrates the use of this work around.

Affected Silicon Revisions

A2	B0						
X	X						

EXAMPLE 1: CONFIGURE PWM MODULE TO PREVENT GLITCHES ON PWM1H AND PWM1L PINS AT THE TIME OF ENABLING

```
TRISAbits.TRISA4 = 1;      // Configure PWM1H/RA4 as digital input
                           // Ensure output is in safe state using pull-up or pull-down resistors
TRISAbits.TRISA3 = 1;      // Configure PWM1L/RA3 as digital input
                           // Ensure output is in safe state using pull-up or pull-down resistors

IOCON1bits.PENH = 0;       // Assign pin ownership of PWM1H/RA4 to GPIO module
IOCON1bits.PENL = 0;       // Assign pin ownership of PWM1L/RA3 to GPIO module

IOCON1bits.OVRDAT = 0;     // Configure PWM outputs override state to the desired safe state

IOCON1bits.OVRENH = 1;     // Override PWM1H output
IOCON1bits.OVRENL = 1;     // Override PWM1L output

PTCONbits.PTEN = 1;        // Enable PWM module

IOCON1bits.OVRENH = 0;     // Remove override for PWM1H output
IOCON1bits.OVRENL = 0;     // Remove override for PWM1L output

Delay(x);                  // Introduce a delay greater than one full PWM cycle

IOCON1bits.PENH = 1;       // Assign pin ownership of PWM1H/RA4 to PWM module
IOCON1bits.PENL = 1;       // Assign pin ownership of PWM1L/RA3 to PWM module
```

17. Module: PWM

In Center-Aligned Complementary mode with Independent Time Base, the expected dead time between transitions of the PWMxH and PWMxL outputs may not be asserted when SWAP is disabled under the following conditions:

- PWMx module is enabled (PTEN = 1)
- SWAP is enabled prior to this event

Work around

None.

Affected Silicon Revisions

A2	B0						
X	X						

18. Module: PWM

In Edge-Aligned Complementary mode, with Master Time Base selected (PWMCONx[9] = 0), a reduction in phase shift, such that PHASEx becomes less than DTRx (or PDCx), will result in an abnormal PWM pulse at the next master period boundary.

Similarly, an increase in phase shift, such that PHASEx becomes greater than DTRx (or PDCx), will result in a reduced dead time for one PWM cycle at the next master period boundary. The amount of dead time is shown in Table 3.

In addition, dead time may also be impacted when updates to the Phase-Shift register occur while operating below the defined dead-time region.

With immediate updates disabled (PWMCONx[0] = 0), the abnormal pulse or reduced dead time may appear at the second master period boundary from when the write to PHASEx occurred.

Table 3 summarizes the expected PWM event that occurs at the master PWM period boundary when PHASEx is updated.

TABLE 3: EXPECTED PWM EVENTS

Condition	Event at Master Period Boundary	Result
PHASEx < PHASEx$_{(new)}$ < DTRx	Reduced Dead Time	Dead Time = DTRx – (PHASEx$_{(new)}$ – PHASEx$_{(old)}$)
PHASEx > PHASEx$_{(new)}$ < DTRx	Increased Dead Time	Dead Time = DTRx + (PHASEx$_{(old)}$ – PHASEx$_{(new)}$)
PHASEx < DTRx, PHASEx$_{(new)}$ > DTRx	Reduced Dead Time	Dead Time = PHASEx$_{(old)}$
PHASEx < PDCx, PHASEx$_{(new)}$ > PDCx	Reduced Dead Time	Dead Time = PDCx + DTRx – PHASEx$_{(new)}$
PHASEx > DTRx, PHASEx$_{(new)}$ < DTRx	Abnormal PWM Pulse	Abnormal Pulse Width = PHASEx$_{(new)}$ – DTRx
PHASEx > PDCx, PHASEx$_{(new)}$ < PDCx	Abnormal PWM Pulse	Abnormal Pulse Width and Possibility for Reduced Dead Time (Zero Dead Time)

Work around

When modifying the PHASEx register on-the-fly, bound the PHASEx register to the following conditions: DTRx < PHASEx < PDCx.

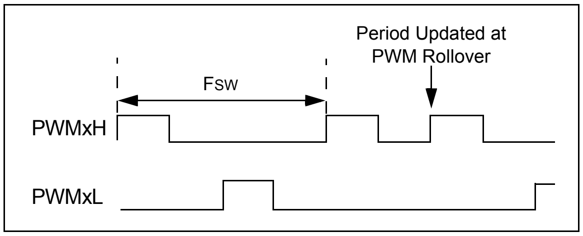
Affected Silicon Revisions

A2	B0						
X	X						

19. Module: PWM

When the PWM module is configured for Push-Pull mode (IOCONx[11:10] = `0b10`) with the Enable Immediate Period Update bit enabled (PTCON [10] = `1`), a write to the Period register that coincides with the period rollover event may cause the push-pull output logic to produce back-to-back pulses on the PWMx pins (see Figure 1).

FIGURE 1:



Work around

Ensure that the update to the PWM Period register occurs away from the PWM rollover event by setting the EIPU bit (PTCON[10] = `1`). Use either the PWM Special Event Trigger (SEVTCMP) or the PWM Primary Trigger (TRIGx) to generate a PWM Interrupt Service Routine (ISR) near the start of the PWM cycle. This ISR will ensure that period writes do not occur near the PWM period rollover event.

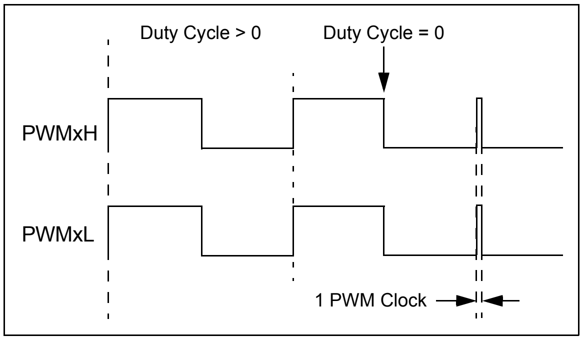
Affected Silicon Revisions

A2	B0						
X	X						

20. Module: PWM

In the Redundant Output mode (IOCONx[11:10] = `0b01`) and Push-Pull Output mode (IOCONx[11:10] = `0b10`), with the Immediate Update Enable bit disabled (PWMCONx[0] = `0`), when the Duty Cycle register is updated from a non-zero value to zero, a glitch pulse of a width equal to 1 PWM clock will appear at the next PWM period boundary, as shown in Figure 2 (for the Redundant Output mode). The Duty Cycle register refers to the PDCx register if PWMCONx[8] = `0` or the MDC register if PWMCONx[8] = `1`.

FIGURE 2: EXAMPLE FOR REDUNDANT OUTPUT MODE



Work around

If the application requires a zero duty cycle output, there are two possible work around methods:

1. Use the PWM override feature to override the PWM output to a low state instead of writing to the Duty Cycle register. In order to switch back to a non-zero duty cycle output, turn off the PWM override. The override-on and override-off events must be timed close to the PWM period boundary if the IOCONx register has been configured with IOCONx[0] = `0` (i.e., output overrides through the OVRDAT[1:0] bits occur on the next CPU clock boundary).
2. Enable the Immediate Update Enable bit (PWMCONx[0] = `1`) while configuring the PWMx module (i.e., before enabling the PWMx module, PTCON[15] = `1`). With the Immediate Update enabled, writes to the Duty Cycle register can have an immediate effect on the PWM output. Therefore, the duty cycle write operations must be timed close to the PWM period boundary in order to avoid distortions in the PWM output.

Affected Silicon Revisions

A2	B0						
X	X						

21. Module: PWM

The triggers generated by the PWMx Primary Trigger Compare Value register (TRIGx) and the PWMx Secondary Trigger Compare Value register (STRIGx) will not trigger at the point defined by the TRIGx/STRIGx register values on the first instance for the configurations listed below. Subsequent trigger instances are not affected.

- Trigger compare values for TRIGx, STRIGx are less than 8 counts
- Trigger Output Divider bits, TRGDIV[3:0] (TRGCONx[15:12]), are greater than ‘0’
- Trigger Postscaler Start Enable Select bits, TRGSTRT[5:0] (TRGCONx[5:0]), are equal to ‘0’

Work around

Configure the PWMx Primary Trigger Compare Value Register (TRIGx) and PWMx Secondary Trigger Compare Value Register (STRIGx) values to be equal to or greater than 8.

Affected Silicon Revisions

A2	B0						
X	X						

22. Module: I/O

Only the following four I/O ports support 5V operation: RC7, RC8, RB8 and RB15. All other I/O ports support 3.3V operation.

Work around

None.

Affected Silicon Revisions

A2	B0						
X	X						

23. Module: I/O

Undesired pin failure may occur on 5V tolerant I/O pins (RC7, RC8, RB8 and RB15).

Work around

If 5V input operation is desired on RC7, RC8, RB8 and RB15, use a current-limiting resistor of at least 1 kOhm.

Affected Silicon Revisions

A2	B0						
X	X						

24. Module: Auxiliary PLL

The Auxiliary PLL Locked Status bit, APLLCK (ACLKCON[14]), is asserted directly after enabling the APLL module (ACLKCON[15]).

Work around

Add a 50 µs delay routine after enabling the APLL Lock bit.

Affected Silicon Revisions

A2	B0						
X	X						

25. Module: I^{2}C

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 I^{2}C module and then re-enable the module.

Affected Silicon Revisions

A2	B0						
X							

26. Module: I^{2}C

In Slave mode, the Bus Collision bit (BCL) cannot be cleared when bus collision detection is enabled (SBCDE = 1).

Work around

Disable the I^{2}C module and then re-enable the module.

Affected Silicon Revisions

A2	B0						
X							

27. Module: CPU

Note: This issue is deterministic based on the instruction sequence executed, and is not sensitive to manufacturing process, temperature, voltage or other application operating conditions that do not affect the instruction sequence.

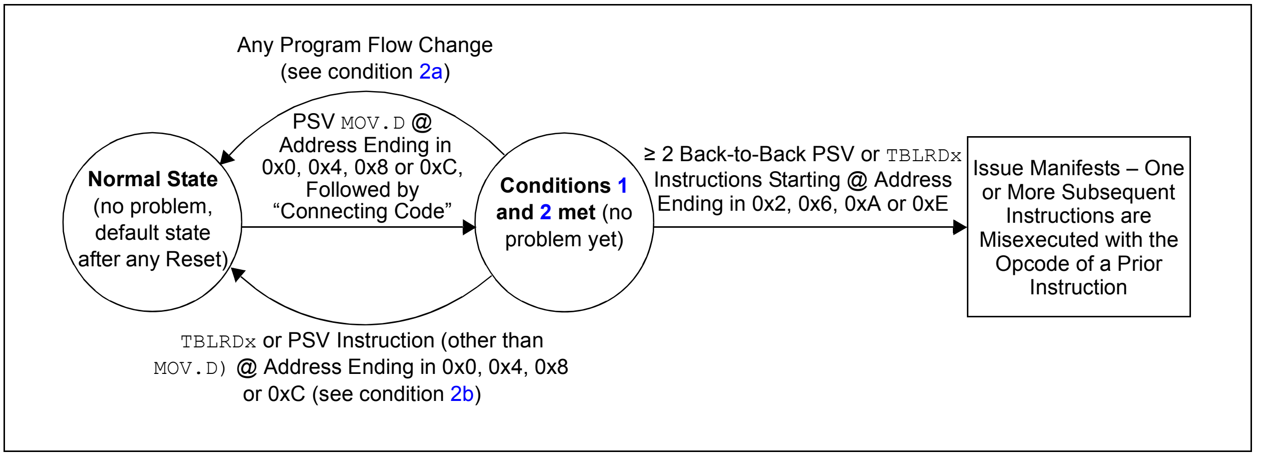
When two or more data Flash read instructions (via Program Space Visibility (PSV) read or Table Read) are executed back-to-back, one or more subsequent instructions can be misexecuted when all of the conditions in Table 4 occur.

TABLE 4: REQUIRED CONDITIONS

1.	A PSV `MOV.D` instruction is executed, with opcode at address ending in 0x0, 0x4, 0x8 or 0xC; and
2.	Some "connecting code" is executed (following the `MOV.D` of condition 1), with the properties: a) The connecting code does not include any program flow changes, including: taken branch instructions (including all versions of `BRA`, `CPBEQ`, `CPBGT`, `CPBLT`, `CPBNE`), `CALL`, `CALL.L`, `GOTO`, `GOTO.L`, `RCALL`, `RETLW`, `RETURN`, vectoring to an ISR, returning from an interrupt (`RETFIE`) and certain debug operations, such as break and one-step; and b) The connecting code does not include a `TBLRDx` or non-`MOV.D` PSV instruction, located at a Flash memory address ending in 0x0, 0x4, 0x8 or 0xC; and c) The connecting code is at least two instruction words in length; and d) The connecting code does not end with a `REPEAT` instruction, with count > `0`; and
3.	≥ 2 back-to-back PSV or `TBLRDx` instructions are executed (following the code of condition 2), where the first of the back-to-back instructions is located at an address ending in 0x2, 0x6, 0xA or 0xE.

Figure 3 provides an example of the effective behavior.

FIGURE 3: SIMPLIFIED BEHAVIOR


Any Program Flow Change
(see condition 2a)
PSV MOV.D @ Address Ending in 0x0, 0x4, 0x8 or 0xC, Followed by "Connecting Code"
Normal State (no problem, default state after any Reset)
Conditions 1 and 2 met (no problem yet)
≥ 2 Back-to-Back PSV or TBLRDx Instructions Starting @ Address Ending in 0x2, 0x6, 0xA or 0xE
Issue Manifests – One or More Subsequent Instructions are Misexecuted with the Opcode of a Prior Instruction
TBLRDx or PSV Instruction (other than MOV.D) @ Address Ending in 0x0, 0x4, 0x8 or 0xC (see condition 2b)

Work around

The issue can be avoided by ensuring any one or more of the requirements are not met. For example:

1. All instances of PSV `MOV.D` can be replaced with two PSV `MOV` instructions instead. Non-PSV `MOV.D` instructions acting on RAM/SFRs do not need to be modified; or
2. If not already present, a program flow change instruction (such as `BRA $+2`) can be inserted above back-to-back data Flash read sequences; or
3. Back-to-back data Flash read instruction sequences can be broken up by inserting a non-Flash read instruction (such as a `NOP`), in between the Flash read instructions; or
4. The alignment of the code can be shifted to avoid the required opcode location addresses.

C code built with MPLAB® XC16 Compiler Version 1.32, or later, implements the work around by default. However, if the application uses Assembly language routines, these should be manually modified to implement the work around. Additionally, if precompiled libraries are used, these should be built with XC16 Version 1.32 or later. For additional information, please visit: **www.microchip.com/erratum_psrd_psrd**

Affected Silicon Revisions

A2	B0						
X	X						

28. Module: PWM

When the PWMx Primary Phase-Shift register (PHASEx[15:0]) value is less than 0x0007 counts, PWM generators configured in Push-Pull mode (IOCONx[11:10] = `0b10`) may exhibit an incorrect phase relationship compared to other PWM generators.

Work around

Ensure that the PWMx Primary Phase-Shift register (PHASEx[15:0]) value is always greater than 0x0007 counts for all PWM generators configured in Push-Pull mode.

Affected Silicon Revisions

A2	B0						
X	X						

29. Module: I/O

If the input signal rise or fall time is greater than 300 nS, the I/O Schmitt trigger output may have glitches.

Work around

The rise/fall times must be less than 300 nS.

Affected Silicon Revisions

A2	B0						
X							

30. Module: I^{2}C

With SMBUS disabled (SMEN (I2CxCONL[8]) = `0`), the I^{2}C V_{IL} (Max) threshold may be lower than 0.3V_{DD}.

With SMBus disabled, as per the specification (DI18 parameter), Input Low Voltage maximum specification for I/O Pins with SDAx and SCLx (V_{IL} Max) should be 0.3 V_{DD}. However, for the I^{2}C to recognize low, the voltage should be driven to 0.8V or below.

Work around

1. With SMBus disabled, Input Low Voltage maximum threshold (V_{IL} Max) on SDAx and SCLx should be below 0.8V.
2. With SMBus enabled with SMEN (I2CxCONL[8]) = `1`, the V_{IL} Max is 0.8V. This voltage is closer to the 0.3 V_{DD}.

Affected Silicon Revisions

A2	B0						
X	X						

31. Module: PLL

When configuring the device to operate at or near maximum MIPS (60 < MIPS ≤ 70) using the on-chip PLL (Phase-Locked Loop), the PLL may temporarily be out of specification during the oscillator switchover event. This incorrect operating frequency may violate Flash access read times, which could result in instruction misexecution, such as device Reset, address error trap or incorrect branching.

Work around

To avoid violating Flash access read times during the oscillator switchover event, the `REPEAT` instruction, followed by the `NOP` instruction, should be inserted immediately after initiating the clock switch as shown below. This instruction sequence will ensure Flash is not being read during the clock switch event. The addition of the `REPEAT` instruction, followed by the `NOP` instruction, adds only a few hundred microseconds (less than 500 µs) to the overall oscillator switchover event.

EXAMPLE 2:

```
// Initiate Clock Switch to Primary Oscillator with PLL (or Fast RC Oscillator with PLL, 0x01)
__builtin_write_OSCCONH(0x03);
__builtin_write_OSCCONL(OSCCON | 0x01);

// Immediately fetch repeat and NOP instructions. NOP executed during clock switch-over event,
// Flash read occurs only once ensuring no miss-execution of instructions
__asm__("REPEAT, #0x7FFF)\n Nop");

// Wait for Clock switch to occur (0b001 for FRC w/ PLL)
while (OSCCONbits.COSC != 0b011);

// Wait for PLL to lock
while (OSCCONbits.LOCK != 1);
```

Affected Silicon Revisions

A2	B0						
X	X						

Data Sheet Clarifications

The following typographic corrections and clarifications are to be noted for the latest version of the device data sheet (DS70005258**C**):

> **Note:** Corrections are shown in **bold**. Where possible, the original bold text formatting has been removed for clarity.

1. Module: High-Speed, 12-Bit Analog-to-Digital Converter (ADC)

A new note should be added to **Register 22-26: ADTRIGxL** and **Register 22-27: ADTRIGxH**:

Note 1: The ADC module should be initialized and calibrated before the selected trigger source (PWM, output compare or timer) is enabled.

APPENDIX A: DOCUMENT REVISION HISTORY

Rev A Document (1/2017)

Initial release of this document; issued for Revision A2.

Rev B Document (3/2017)

Device ID for Revision A2 has been changed from 0x4002 to 0x4001.

Added data sheet clarification 1 (Direct Memory Access (DMA)) and 2 (Flash Program Memory).

Rev C Document (7/2017)

Updated silicon issue 18 (PWM).

Added silicon issues 25 (I^2C), 26 (I^2C) and 27 (CPU).

Added data sheet clarification 3 (Pin Diagrams), 4 (Device Overview), 5 (Constant Current Source) and 6 (High-Speed PWM).

Rev D Document (1/2018)

Added silicon issue 28 (PWM).

Rev E Document (10/2018)

Updated data sheet revision to ‘**C**’.

Added silicon revision B0.

Updated silicon issue 4 (ADC).

Added silicon issue 29 (I/O).

Removed all data sheet clarifications.

Rev F Document (2/2019)

Added silicon issue 30 (I^2C).

This revision of the document shows changed bit representation (e.g., <3:0> changed to [3:0]) to be consistent with SDL documents.

Rev G Document (8/2019)

Adds new silicon issue 31 (PLL).

Adds data sheet clarification: 1 (High-Speed, 12-Bit Analog-to-Digital Converter (ADC)).

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.

For information regarding Microchip's Quality Management Systems, please visit www.microchip.com/quality.

ISBN: 978-1-5224-4900-3

Worldwide Sales and Service

AMERICAS

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

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

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: 61-2-9868-6733

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: 86-25-8473-2460

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 - Xian
Tel: 86-29-8833-7252

China - Xiamen
Tel: 86-592-2388138

China - Zhuhai
Tel: 86-756-3210040

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: 84-28-5448-2100

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: 358-9-4520-820

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

Germany - Garching
Tel: 49-8931-9700

Germany - Haan
Tel: 49-2129-3766400

Germany - Heilbronn
Tel: 49-7131-72400

Germany - Karlsruhe
Tel: 49-721-625370

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

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

Israel - Ra'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: 34-91-708-08-90
Fax: 34-91-708-08-91

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