Product Change Notification - SYST-25MCXX581

Date:

26 Feb 2019

Product Category:

16-Bit - Microcontrollers and Digital Signal Controllers

Affected CPNs:

Notification subject:

ERRATA - dsPIC33EPXXGS202 Family Silicon Errata and Data Sheet Clarification

Notification text:

SYST-25MCXX581

Microchip has released a new DeviceDoc for the dsPIC33EPXXGS202 Family Silicon Errata and Data Sheet Clarification of devices. If you are using one of these devices please read the document located at dsPIC33EPXXGS202 Family Silicon Errata and Data Sheet Clarification.

Notification Status: Final

Description of Change:

1) Adds new silicon issue 33.
2) This revision of the document shows changed bit representation (e.g., changed to [3:0]) to be consistent with SDL documents.

Impacts to Data Sheet: None

Reason for Change: To Improve Productivity

Change Implementation Status: Complete

Estimated First Ship Date: 26 Feb 2019

NOTE: Please be advised that after the estimated first ship date customers may receive pre and post change parts.

Markings to Distinguish Revised from Unrevised Devices: Traceability Code

Attachment(s):

dsPIC33EPXXGS202 Family Silicon Errata and Data Sheet Clarification

Please contact your local Microchip sales office with questions or concerns regarding this notification.

Terms and Conditions:

If you wish to receive Microchip PCNs via email please register for our PCN email service at our PCN home page select register then fill in the required fields. You will find instructions about registering for Microchips PCN email service in the PCN FAQ section.
If you wish to change your PCN profile, including opt out, please go to the PCN home page select login and sign into your myMicrochip account. Select a profile option from the left navigation bar and make the applicable selections.

Affected Catalog Part Numbers (CPN)

DSPIC33EP16GS202-E/M6
DSPIC33EP16GS202-E/MM
DSPIC33EP16GS202-E/MX
DSPIC33EP16GS202-E/SO
DSPIC33EP16GS202-E/SS
DSPIC33EP16GS202-E/SSC03
DSPIC33EP16GS202-E/SSVAO
DSPIC33EP16GS202-I/M6
DSPIC33EP16GS202-I/MM
DSPIC33EP16GS202-I/MX
DSPIC33EP16GS202-I/SO
DSPIC33EP16GS202-I/SS
DSPIC33EP16GS202T-E/M6
DSPIC33EP16GS202T-E/MM
DSPIC33EP16GS202T-E/MX
DSPIC33EP16GS202T-E/SO
DSPIC33EP16GS202T-E/SS
DSPIC33EP16GS202T-E/SSV02
DSPIC33EP16GS202T-E/SSVAO
DSPIC33EP16GS202T-I/M6
DSPIC33EP16GS202T-I/MM
DSPIC33EP16GS202T-I/MMC01
DSPIC33EP16GS202T-I/MX
DSPIC33EP16GS202T-I/SO
DSPIC33EP16GS202T-I/SS
DSPIC33EP32GS202-E/M6
DSPIC33EP32GS202-E/M6C01
DSPIC33EP32GS202-E/MM
DSPIC33EP32GS202-E/MX
DSPIC33EP32GS202-E/SO
DSPIC33EP32GS202-E/SS
DSPIC33EP32GS202-I/M6
DSPIC33EP32GS202-I/MM
DSPIC33EP32GS202-I/MX
DSPIC33EP32GS202-I/SO
DSPIC33EP32GS202-I/SS
DSPIC33EP32GS202T-E/M6
DSPIC33EP32GS202T-E/M6C01
DSPIC33EP32GS202T-E/MM
DSPIC33EP32GS202T-E/MX
DSPIC33EP32GS202T-E/SO
DSPIC33EP32GS202T-E/SS
DSPIC33EP32GS202T-E/SSV01
DSPIC33EP32GS202T-I/M6
DSPIC33EP32GS202T-I/MM
DSPIC33EP32GS202T-I/MMC02

DSPIC33EP32GS202T-I/MX
DSPIC33EP32GS202T-I/SO
DSPIC33EP32GS202T-I/SS

dsPIC33EPXXGS202 Family Silicon Errata and Data Sheet Clarification

The dsPIC33EPXXGS202 family devices that you have received conform functionally to the current Device Data Sheet (DS70005208**E**), 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 dsPIC33EPXXGS202 family 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 18, 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 silicon revisions of the dsPIC33EPXXGS202 family are shown in Table 1.

TABLE 1: SILICON DEVREV VALUES

Part Number	Device ID(1)	Revision ID for Silicon Revision(2)	
		A3	B0
dsPIC33EP16GS202	0x6D01	0x4003	0x4004
dsPIC33EP32GS202	0x6D11		

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 *"dsPIC33EPXXGS202 Family Flash Programming Specification"* (DS70005192) 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)	
				A3	B0
UART	Break Character Transmission	1.	The Transmit Shift Register Empty (TRMT) bit is unreliable when there are back-to-back Break character transmissions.	X	X
PWM	Latched Faults	2.	In PWM Latched Fault mode, the PWM outputs may be latched on both the rising and the falling edge of the Fault signal, regardless of the Fault input polarity selection (set with the FLTPOL bit (FCLCONx[2]) setting).	X	X
—	—	3.	—	—	—
PWM	Master Time Base Mode	4.	Changes to the PHASEx register may result in missing dead time.	X	X
CPU	`div.sd`	5.	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	`DO` Loop	6.	PSV access, including table reads or writes in the first or last instruction of a `DO` loop, is not allowed.	X	X
PWM	Redundant/ Push-Pull Output Mode	7.	Changing the duty cycle value from a non-zero value to zero will produce a glitch pulse equal to 1 PWM clock.	X	X
PWM	Status Bits	8.	PWM Fault status bits do not function if the associated PWM Fault interrupts are disabled.	X	X
PWM	Push-Pull Mode	9.	When EIPU = `1`, Period register writes may produce back-to-back pulses under certain conditions.	X	X
PWM	Trigger Compare Match	10.	The first PWM/ADC trigger event on a TRIGx/STRIGx match may not occur under certain conditions.	X	X
I^2C	Slave Mode	11.	Bus data can get corrupted when it matches with one of the slave addresses connected to the bus.	X	
PWM	Push-Pull Mode	12.	When EIPU = `0`, a period update may produce back-to-back pulses.	X	
Auxiliary PLL	APLL Lock	13.	The APLL lock bit is asserted directly after enabling the APLL.	X	X
ADC	ADC Conversion	14.	Under specific conditions, multi-core ADC conversion cross-talk noise might be present.	X	X
I^2C	Slave Mode	15.	In 10-Bit Addressing Slave mode, on receiving the upper address byte (A9 and A8 bits), the Acknowledge Time Status bit (ACKTIM) is not asserted during the Acknowledgment sequence.	X	
I^2C	Slave Receive Mode	16.	The Acknowledge Time Status bit (ACKTIM) is asserted only if Address Hold Enable (AHEN) or Data Hold Enable (DHEN) is enabled.	X	
PWM	PWM Module Enable	17.	A glitch may be observed on the PWM pins when the PWM module is enabled after the assignment of pin ownership to the PWM module.	X	X
PWM	Center-Aligned Complementary	18.	Dead time between transitions of the PWMxH and PWMxL outputs may not be asserted when SWAP mode is disabled.	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)	
				A3	B0
Comparator	Comparator Output Jitter	19.	The Comparator module may generate erroneous triggers/interrupts.	X	
CPU	Variable Interrupt Latency	20.	When Variable Interrupt Latency is selected (VAR = 1), an address error trap or incorrect application behavior may occur.	X	
CPU	Context Switching	21.	When nesting more than one interrupt (without an Alternate Working register set) within the interrupts which are using Alternate Working register sets, there will be an unexpected change in the CCTXI[2:0] bits in the CTXTSTAT register while returning from the highest priority interrupt.	X	
I2C	Address Hold	22.	In Slave mode when AHEN = 1 (Address Hold Enable), if the ACKDT bit (Acknowledge Data) is set at the beginning of address reception, clock stretching will not happen after the 8th clock.	X	
I2C	Data Hold	23.	In Slave mode when DHEN = 1 (Data Hold Enable), if the ACKDT bit (Acknowledge Data) is set at the beginning of data reception, then the slave interrupt will not occur after the 8th clock.	X	
SPI	SPI Enable	24.	When SPI is enabled for the first time, there may be a spurious clock on the SCK which causes a mismatch between the clock and data lines.	X	X
I/O	Schmitt Trigger	25.	Schmitt Trigger output may produce glitches.	X	
I2C	Bus Collisions	26.	In Slave mode, false bus collision triggers are generated when bus collision is enabled (SBCDE = 1).	X	
I2C	Hold Time	27.	Minimum hold time of 300 ns is not achieved when the SDA1 Hold Time Selection bit (SDAHT) is set.	X	
CPU	Data Flash Reads	28.	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
I2C	Slave Mode	29.	In Slave mode, the Master Bus Collision Detect bit (BCL) cannot be cleared when bus collision detection is enabled (SBCDE = 1).	X	
I2C	Slave Mode	30.	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).	X	X
I2C	Slave Mode	31.	In Slave mode, false bus collision triggers are generated when the bus collision is enabled (SBCDE = 1) and a Stop bit is received.		X
PWM	Push-Pull Mode	32.	PWM generators may exhibit an incorrect phase relationship when configured in Push-Pull mode.		X
I2C	I/O Voltage Threshold	33.	With SMBus disabled, the I2C VIL (Max) threshold may be lower than 0.3 VDD.	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 (**B0**).

1. 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 (U1STA[11]), to be cleared instead of the TRMT bit (U1STA[8]) to be set. The UTXBRK status bit will be cleared after a Break character transmission.

Affected Silicon Revisions

A3	B0						
X	X						

2. Module: PWM

In PWM Latched Fault mode, the PWM outputs may be latched on both the rising and the falling edge of the Fault signal, regardless of the Fault input polarity selection (set with the FLTPOL bit (FCLCONx[2]) setting).

Work around

None.

Affected Silicon Revisions

A3	B0						
X	X						

3. Module: —

Silicon errata issue 3 has been corrected and does not apply to the current silicon revision.

Affected Silicon Revisions

A3	B0						

4. 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

A3	B0						
X	X						

5. Module: CPU

When using the Signed 32-by-16-bit Division instruction, `div.sd`, the Overflow bit does 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

A3	B0						
X	X						

6. Module: CPU

Table Write (`TBLWTx`), Table Read (`TBLRDx`) and PSV Flash read instructions should not be used in the first or last instruction locations of a `DO` loop.

Work around

None.

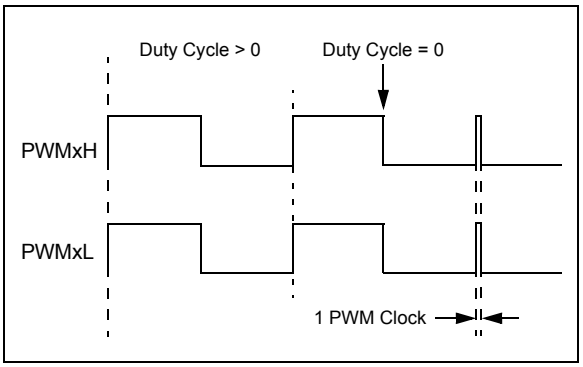
Affected Silicon Revisions

A3	B0						
X	X						

7. 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 one PWM clock will appear at the next PWM period boundary, as shown in Figure 1 (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 1: 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 OVDDAT[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

A3	B0						
X	X						

8. Module: PWM

If PWM Fault interrupts are disabled (FLTIEN = 0 or CLIEN = 0), then the associated status bits (FLTSTAT and CLSTAT) will not function.

Work around

None.

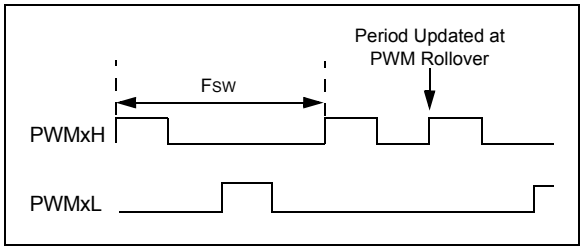
Affected Silicon Revisions

A3	B0						
X	X						

9. 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 (Figure 2).

FIGURE 2:



Work around

If the EIPU bit is set (PTCON[10] = 1), ensure that the update to the PWM Period register occurs away from the PWM rollover event. 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. The PWM Period register could be updated inside this ISR so that the period writes do not occur near the PWM period rollover event.

Affected Silicon Revisions

A3	B0						
X	X						

10. 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 and STRIGx are less than eight 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

A3	B0						
X	X						

11. Module: I^2C

In applications with multiple I^2C slaves, bus data can become corrupted when the data payload sent to an addressed slave device matches the bus address of another (unaddressed) slave device.

Work around

Keep track of the bus address and data phases in software. When Address Hold Enable is used (the AHEN bit is set), the application can assert a NACK for any of the received bytes (invalid addresses and data bytes for other slave devices) until a Stop bit is received.

Affected Silicon Revisions

A3	B0						
X							

12. Module: PWM

When the PWM module is configured for Push-Pull mode (IOCONx[11:10] = 0b10) with the Enable Immediate Period Update bit (PTCON[10] = 0), and is operating in Master Time Base mode (ITB (PWMCONx[9]) = 0), a write to the Period register occurs on PWMx cycle boundaries. This may cause the push-pull output logic to produce back-to-back pulses on the PWMx pins.

Work around

Work around 1: Ensure that the Enable Immediate Period Updates bit (PTCON[10] = 1) is set.

Work around 2: Configure the PWM Phase-Shift Value (PHASEx[15:0]) with a value higher than 0x0007. When multiple PWM generators are configured in Push-Pull mode, configure the PWM Phase Shift Value with a value higher than 0x0007 for respective PWM generators.

Affected Silicon Revisions

A3	B0						
X							

13. Module: Auxiliary PLL

The Auxiliary PLL Lock bit (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

A3	B0						
X	X						

14. Module: ADC

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

Work around for Revision A3

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

A3	B0						
X	X						

15. Module: I²C

In I²C 10-Bit Slave Addressing mode, on receiving the upper address byte (A9 and A8 bits), the Acknowledge Time status bit (ACKTIM) is not asserted during the Acknowledgment sequence. This issue is not seen during the reception of the lower address byte (A7 to A0) and data bytes.

The hardware asserts the ACKTIM on the falling edge of the eighth clock and deasserts on the rising edge of the ninth clock. In this case, ACKTIM is not asserted on upper address byte reception. When AHEN = 1, the clock is stretched after the 8th falling edge and the ACKTIM bit is asserted until the clock is released. If AHEN = 0, the clock is not stretched and ACKTIM is asserted during the Acknowledgment sequence, which is of a short duration. Therefore, the user application can see this issue of the ACKTIM bit not getting asserted when AHEN = 1.

Work around

Instead of polling for ACKTIM to be asserted, poll for the RBF flag.

Affected Silicon Revisions

A3	B0						
X							

16. Module: I²C

In I²C Slave Receive mode, the Acknowledge Time status bit (ACKTIM) has no effect if the Address Hold Enable (AHEN) and Data Hold Enable (DHEN) bits are disabled (AHEN = 0 and DHEN = 0). The Acknowledge Time status bit (ACKTIM) is asserted only if the Address Hold Enable (AHEN) or Data Hold Enable (DHEN) bit is enabled.

Work around

Instead of polling for the ACKTIM bit to be asserted, poll for the RBF flag.

Affected Silicon Revisions

A3	B0						
X							

17. 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] bit 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 configuring 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

A3	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
```

18. 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 (IOCONx[1]) 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

A3	B0						
X	X						

19. Module: Comparator

Analog comparator output may have a jitter when it is operating and this will generate erroneous triggers/interrupts. If the PWM module is configured to be controlled by an analog comparator, the output of the PWM generator may be affected by jitter in the analog comparator output.

Work around:

Configure the Comparator Hysteresis Select bits, HYSSEL[1:0], as '0b11' (20 mV hysteresis) and set the Digital Filter Enable bit, FLTREN (CMPxCON[10]), to '1'.

Affected Silicon Revisions

A3	B0						
X							

20. 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

A3	B0						
X							

21. Module: CPU

When returning from an Interrupt Service Routine (ISR) by executing the `RETFIE` instruction, in the case of a nested interrupt, the Interrupt Priority bits (IPC[3:0]) associated with the lower priority interrupt are compared with the CTXTn bits field in the FLTREG Configuration register:

- If there is a match with either of the CTXTn bits field, then the corresponding Alternate Working Register Set is chosen and the Current Context Identifier bits (CCTXI[2:0]) in the CTXTSTAT register are updated to reflect the new Alternate Working Register Set.
- If there is no match with either of the CTXTn bits field, then the expected behavior is to keep the context (defined by the value of the Current Context Identifier bits, CCTXI[2:0] in the CTXTSTAT register) unchanged. However, the context gets changed. A new context corresponding to the value in the Manual Context Identifier bits (MCTXI[2:0]) in the CTXTSTAT register is selected by the hardware and the CCTXI[2:0] bits in the CTXTSTAT register are modified to reflect this change.

When using interrupts with the Alternate Working Register Set (automatic context assignment), no more than one ISR without an Alternate Working Register Set must be nested within an ISR with an Alternate Working Register Set.

Issue and work around are illustrated in Figure 3 and Figure 4, respectively. The figures show the status bits, CCTXI[2:0] and MCTXI[2:0], after entering into an ISR from a lower priority ISR (left pane), and after returning to the same ISR from a higher priority ISR (right pane).

FIGURE 3: MISMATCH OF WORKING REGISTER SET WHEN NESTING MORE THAN ONE INTERRUPT WITHIN INTERRUPTS THAT USE ALTERNATE WORKING REGISTER SETS

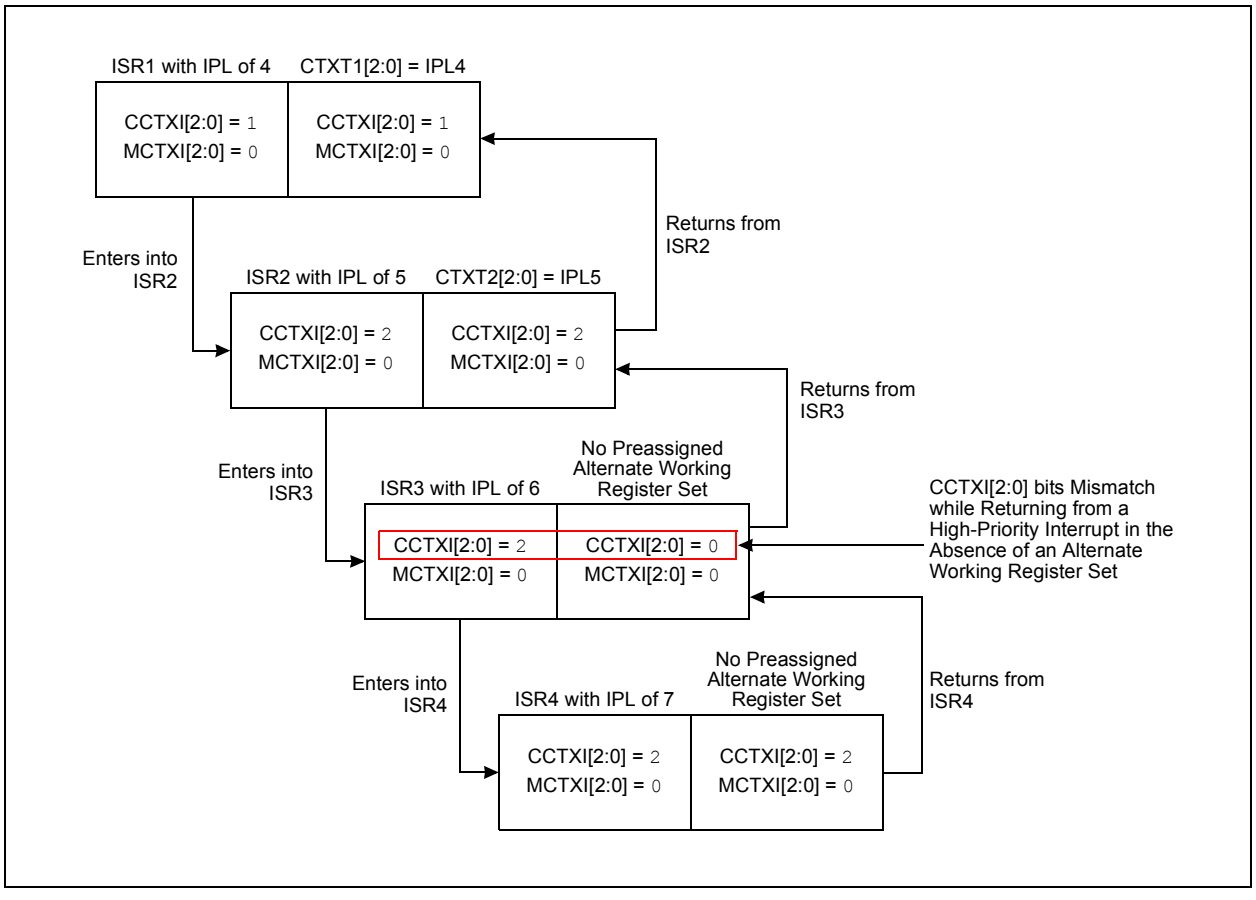
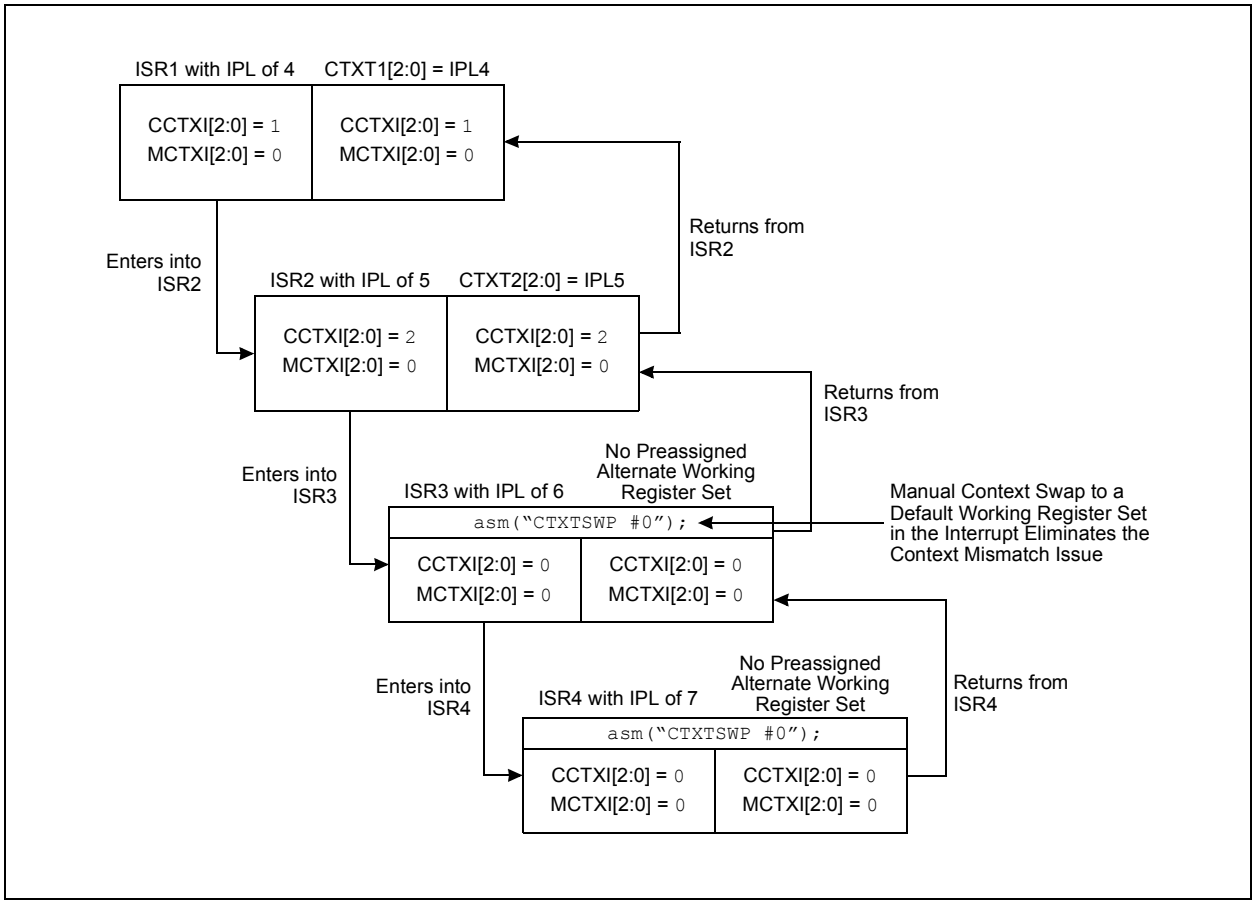


FIGURE 4: WORK AROUND FOR MISMATCH OF WORKING REGISTER SET WHEN NESTING MORE THAN ONE INTERRUPT WITHIN INTERRUPTS THAT USE ALTERNATE WORKING REGISTER SETS



Work around

Work around 1: When using interrupts with the Alternate Working Register Set, at the entry of all ISRs that do not have an Alternate Working Register Set and have a higher IPL level than the ISRs with an Alternate Working Register Set, perform a manual context swap to Context #0 as:

```
asm("CTXTSWP #0");
```

Note: The application software must not perform a manual context swap (using the `CTXTSWP` instruction) to a context other than Context #0.

Work around 2: Always assign higher IPLs for the ISRs that use an Alternate Working Register Set than for the ISRs that do not use an Alternate Working Register Set.

Affected Silicon Revisions

A3	B0						
X							

22. Module: I²C

In Slave mode, when AHEN = 1 (Address Hold Enable), if the ACKDT bit (Acknowledge Data) is set at the beginning of address reception, clock stretching will not happen after the 8th clock.

Work around

In Slave mode, user software should clear ACKDT on receiving the Start bit.

Affected Silicon Revisions

A3	B0						
X							

23. Module: I²C

In Slave mode, when DHEN = 1 (Data Hold Enable), if the ACKDT bit (Acknowledge Data) is set at the beginning of data reception then the slave interrupt will not occur after the 8th clock.

Work around

In Slave mode, user software should clear ACKDT on receiving the Start bit.

Affected Silicon Revisions

A3	B0						
X							

24. Module: SPI

When SPI is enabled for the first time, there may be a spurious clock on the SCK. This may result in one bit of data shifted out on the data line, resulting in a mismatch between the clock and data lines.

This issue may also occur when the SPI is disabled during data transmission and enabled subsequently.

Work around

1. Disable the SPI module after two SPI cycles and then re-enable SPI; this will synchronize the clock and data.
2. If the SPI is configured on PPS pins, first enable the SPI without configuring the PPS, then allow two SPI clocks to pass and then configure the PPS to connect to the SPI module. This will prevent the spurious SPI clock going out on the pin. If the SPI module is turned off periodically, ensure to turn off the PPS as well.

Affected Silicon Revisions

A3	B0						
X	X						

25. 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

A3	B0						
X							

26. Module: I²C

In Slave mode, false bus collision triggers are generated when the bus collision is enabled (SBCDE = 1).

Work around

None.

Affected Silicon Revisions

A3	B0						
X							

27. Module: I²C

Minimum hold time of 300 ns on SDAx after the falling edge of SCLx is not achieved when the Data Hold Time bit (SDAHT) is set.

Work around

None.

Affected Silicon Revisions

A3	B0						
X							

28. 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.

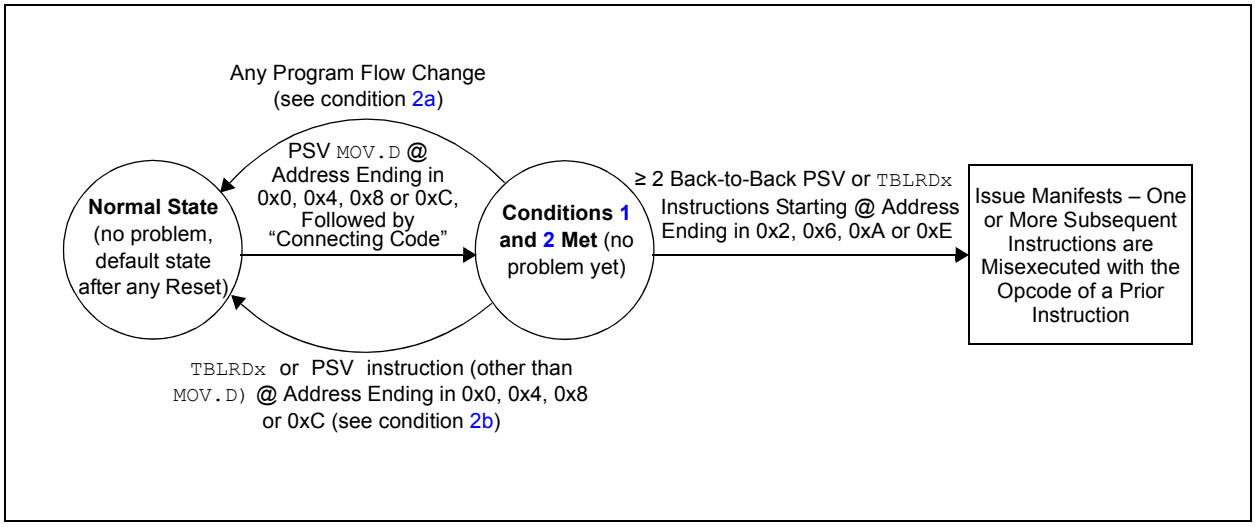
When two or more data Flash read instructions (via Program Space Visibility (PSV) read or table read) are executed back-to-back, one or more subsequent instructions can be misexecuted when all of the conditions in Table 4 occur.

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 5 provides an example of the effective behavior.

FIGURE 5: SIMPLIFIED BEHAVIOR



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

A3	B0						
X	X						

29. Module: I²C

In Slave mode, the Bus Collision 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

A3	B0						
X							

30. 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

A3	B0						
X	X						

31. 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

A3	B0						
	X						

32. 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 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

A3	B0						
	X						

33. Module: I^2C

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

With SMBus disabled, as per the specification (DI18 parameter), Input Low Voltage maximum specification for I/O pins with SDA1 and SCL1 (V_{IL} Max) should be 0.3 V_{DD}. However, for the I^2C 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 SDA1 and SCL1 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

A3	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 (DS70005208**E**):

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

None.

APPENDIX A: DOCUMENT REVISION HISTORY

Rev A Document (5/2015)

This version of the document was never released.

Rev B Document (9/2015)

Includes silicon issues: 1 (UART), 2-4 (PWM), 5-6 (CPU), 7-10 (PWM), 11 (I^2C), 12 (PWM), 13 (Auxiliary PLL), 14 (ADC), 15-16 (I^2C), 17-18 (PWM) and 19 (Comparator).

Rev C Document (3/2016)

Adds new silicon issues: 20 (CPU), 21 (CPU), 22 (I^2C), 23 (I^2C) and 24 (SPI).

Rev D Document (5/2017)

Adds new silicon issues: 25 (I/O), 26 (I^2C) and 27 (I^2C).
Removes silicon issue 3.

Rev E Document (7/2017)

Updates silicon issues: 4 (PWM), 5 (CPU) and 6 (CPU).

Adds new silicon issues: 28 (CPU), 29 (I^2C) and 30 (I^2C).

Adds data sheet clarifications 1 (Document Revision History) and 2 (Electrical Characteristics).

Rev F Document (9/2017)

Adds data sheet clarifications: 3 (Electrical Characteristics), 4 (Electrical Characteristics) and 5 (Electrical Characteristics).

Rev G Document (4/2018)

Adds new silicon revision B0.

Adds new silicon issues 31 (I^2C) and 32 (PWM).

Removes all Data Sheet Clarifications as they are addressed in the data sheet revision DS70005208**E**.

Rev H Document (2/2019)

Adds new silicon issue 33 (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.

NOTES:

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

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

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

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

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
═ ISO/TS 16949 ═

Trademarks

ISBN: 978-1-5224-4199-1

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: 49-7131-67-3636

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