dsPIC33EPXXX(GP/MC/MU)806/810/814 and PIC24EPXXX(GP/GU)810/814 Family Silicon Errata and Data Sheet Clarification

The dsPIC33EPXXX(GP/MC/MU)806/810/814 and PIC24EPXXX(GP/GU)810/814 family devices that you have received conform functionally to the current Device Data Sheet (DS70616**G**), 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 dsPIC33EPXXX(GP/MC/MU)806/810/814 and PIC24EPXXX(GP/GU)810/814 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 (**B1**).

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 web site (www.microchip.com).

For example, to identify the silicon revision level using MPLAB IDE in conjunction with MPLAB ICD 3 or PICkit™ 3:

1. Using the appropriate interface, connect the device to the MPLAB ICD 3 programmer/debugger or PICkit 3.
2. From the main menu in MPLAB IDE, select *Configure>Select Device*, and then select the target part number in the dialog box.
3. Select the MPLAB hardware tool (*Debugger>Select Tool*).
4. Perform a "Connect" operation to the device (*Debugger>Connect*). 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 Device and Revision ID values for the various dsPIC33EPXXX(GP/MC/MU)806/810/814 and PIC24EPXXX(GP/GU)810/814 silicon revisions are shown in Table 1.

TABLE 1: SILICON DEVREV VALUES

Part Number	Device ID(1)	Revision ID for Silicon Revision(2)
		B1
dsPIC33EP256MU806	0x1861	0x4002
dsPIC33EP256MU810	0x1862	
dsPIC33EP256MU814	0x1863	
PIC24EP256GU810	0x1826	
PIC24EP256GU814	0x1827	
dsPIC33EP512MU810	0x1872	
dsPIC33EP512MU814	0x1873	
dsPIC33EP512GP806	0x187D	
dsPIC33EP512MC806	0x1879	

Note 1: The Device and Revision IDs (DEVID and DEVREV) are located at the last two implemented addresses in program memory.

2: Refer to the *"dsPIC33E/PIC24E Flash Programming Specification"* (DS70619) for detailed information on Device and Revision IDs for your specific device.

TABLE 1: SILICON DEVREV VALUES (CONTINUED)

Part Number	Device ID[1]	Revision ID for Silicon Revision[2]
		B1
PIC24EP512GU810	0x1836	0x4002
PIC24EP512GU814	0x1837	
PIC24EP512GP806	0x183D	

Note 1: The Device and Revision IDs (DEVID and DEVREV) are located at the last two implemented addresses in program memory.

2: Refer to the *"dsPIC33E/PIC24E Flash Programming Specification"* (DS70619) 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]
				B1
I/O Multiplexer	Open-Drain Feature	1.	The open-drain control signal from GPIO, PORTA bit 3, is inadvertently controlling the data output of bit 15 of PORTC.	X
CPU	DSP DO Instruction	2.	The assembly language DO instruction does not function correctly with nested loops when the inner loop count is zero (one iteration).	X
CPU	div.sd Instruction	3.	When using the div.sd instruction, the overflow bit is not getting set when an overflow occurs.	X
PPS	Virtual Remap	4.	Virtual pin remapping does not work correctly.	X
SPI	Frame Sync Pulse	5.	Frame sync pulse not generated in Master mode when FRMPOL = 0.	X
SPI	Frame Sync Pulse	6.	When in SPI Slave mode, with the frame sync pulse set as an input, FRMDLY must be set to '0'.	X
PWM	Dead-Time Compensation	7.	The duty cycle is not correct when using dead time and the programmed duty cycle is close to 0% or 100%.	X
PWM	Dead-Time Compensation	8.	Dead-time compensation is not enabled for Center-Aligned PWM Mode.	X
Power System	BOR	9.	BOR must always be enabled.	X
Reserved	—	10.	—	—
ECAN™	CANCKS	11.	The function of the CANCKS bit is reversed.	X
ECAN	ERRIF	12.	The ERRIF status bit does not get set when a CAN error condition occurs.	X
USB	Low-Speed through Hub	13.	The USB bus might not be returned to the J-state following an Acknowledgment packet when running low-speed through a hub.	X
USB	UIDLE Interrupt	14.	UIDLE interrupts cease if the UIDLE interrupt flag is cleared.	X
DMA	CAN	15.	Possible loss of interrupts when DMA is used with CAN.	X
UART	TX Interrupt	16.	A TX Interrupt may occur before the data transmission is complete.	X
UART	UTXBRK	17.	When a Read-Modify-Write operation is performed to set or clear any bit(s) in the UxSTA register while hardware is clearing the UTXBRK bit, the UTXBRK bit may remain set.	X
UART	UARTEN	18.	The Transmitter Write Pointer does not get cleared when the UART is disabled (UARTEN = 0), it requires TXEN to be set in order to clear the Write Pointer.	X
I^2C™	I2CxCON	19.	When a Read-Modify-Write operation is performed to set or clear any bit(s) in the I2CxCON register while hardware is clearing the ACKEN bit, the ACKEN bit may remain set.	X
ADC	DONE bit	20.	The ADC conversion status bit, DONE (ADxCON1<0>), does not indicate completion of conversion when the external interrupt is selected as the ADC trigger source (ADxCON1<SSRC> = 1).	X
PMP	—	21.	On the dsPIC33EPXXXGU814 or PIC2EPXXXMU814 devices, the PMCS1/PMA14 and PMCS2/PMA15 pin functionality is duplicated on the RJ14/15 pins in addition to the expected RK11/RK12 pins.	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)
				B1
Flash Memory	ICSP™	22.	If either the General or Auxiliary Segment is protected, neither segment can be read.	X
Flash Memory	Programming	23.	Stall mechanism may not function properly when erasing or programming Flash memory while executing code from the same Flash segment.	X
Power System	Flash Regulator	24.	The RCON<VREGSF> bit always reads as '0'.	X
PWM	Dead Time	25.	When operating in Edge-Aligned Complimentary mode, the dead time could become zero.	X
QEI	Index Counter	26.	The QEI Index Counter does not count correctly in Quadrature Detector mode.	X
QEI	Modulo Mode	27.	Modulo mode functionality is incorrect when the Count Polarity bit is set.	X
CPU	DO Loop	28.	PSV access, including Table Reads or Writes, in the first or last instruction of a DO loop are not allowed.	X
PWM	Master Time Base Mode	29.	In Master Time Base mode, writing to the period register and any other timing parameter of the PWM module will cause the update of the other timing parameter to take effect one PWM cycle after the period update is effective.	X
ECAN™	DMA	30.	Write collisions on a DMA-enabled ECAN™ peripheral do not generate DMAC Error traps.	X
Auxiliary Flash	Interrupt Vector	31.	When a device is set to obtain the RESET instruction from Auxiliary Flash location, 0x7FFFFC (RSTPRI = 0), and Auxiliary Flash code protection is enabled, the device does not execute the application code after a Reset.	X
Auxiliary Flash	Interrupt Vector	32.	All interrupts and traps should vector through the single auxiliary interrupt vector when executing from the Auxiliary Segment; however, the address error trap is not.	X
Output Compare	Interrupt	33.	Under certain circumstances, an output compare match may cause the interrupt flag (OCxIF) to become set prior to the Change-of-State (COS) of the OCx pin.	X
ADC	1.1 Msps Sampling	34.	Selecting the same ANx input (AN0 or AN3) for CH0 and CH1 to achieve a 1.1 Msps sampling rate results in erroneous readings for CH1.	X
CAN	ICODE Bits	35.	A transmit-interrupt event that happens immediately after the reception-interrupt event will cause the ICODE bits to point to the transmit buffer instead of the receive buffer.	X
PMP	PORTH15-PORTH8	36.	On dsPIC33EPXXXGU814 or PIC2EPXXXMU814devices, I/O signals RH15-RH8 cannot be used in PMP mode.	X
Core	Do Loop	37.	DO loops may work incorrectly if nested interrupts are enabled and interrupts occur during the last two instructions of the DO loop.	X
Core	Variable Interrupt Latency	38.	Address error trap may occur under certain circumstances if Variable Interrupt Latency mode is enabled.	X
CPU	Data Flash Reads	39.	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

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 (**B1**).

1. Module: I/O Multiplexer

The open-drain control signal from GPIO PORTA, bit 3, is inadvertently controlling the data output of bit 15 of PORTC.

Work around

There is no work around. If bit 15 of PORTC is being used as an output (FOSC<OSCIOFNC> = 0 and TRISC<15> = 0), the open-drain capability for bit 3 of PORTA cannot be enabled and used as an open-drain output (ODCA<3> ≠ 1).

This issue only applies when the OSC2 pin function is set for general purpose digital I/O (FOSC<ISCIOFNC> = 0) and the device is using the FRC or EC Primary Oscillator modes.

Affected Silicon Revisions

B1							
X							

2. Module: CPU

Note:	This silicon issue applies only to the dsPIC33EPXXXGP/MC/MU806/810/814 devices.

The assembly language DO instruction does not function correctly with nested loops when the inner loop count is zero (one iteration).

Work around

With nested DO loops where an inner loop can have a single iteration (DCOUNT = 0), one of the following precautions must be taken for proper operation:

1. Insert a NOP immediately after the DO instruction, which can contain a loop count of zero, and insert two NOP instructions immediately following the DO instruction of the next outer loop, as shown in Example 1.
2. Alternatively, the code can test for a count of zero and branch over the DO instruction for all nested loops. In this case, insertion of NOP instructions is not required.

This issue does not apply if there are no nesting of DO loops or if all of the inner loop counts are always greater than zero. When a nested DO loop has a count of zero, only the immediate next outer loop will be affected.

Affected Silicon Revisions

B1							
X							

EXAMPLE 1:

```
do      w0, out_lp              ; w0 contains the count for the outer loop
        nop                     ; 2 NOPs are added for the work around
        nop                     ;
        …                       ; user code
        do      w1, in_lp       ; w1 contains count for the inner loop
        nop                     ; A Single NOP is the required work around for the inner loop
        …                       ; user code
in_lp:                          ; end of inside loop
        …
out_lp:                         ; end of outside loop
```

3. 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

B1							
X							

4. Module: PPS

Virtual pin remapping does not work.

Work around

Virtual remapping applies to comparator outputs and the QEI Home and Index functions:

1. Comparator outputs can be connected to a peripheral input by mapping both the comparator output and the peripheral input to an unused physical pin using the peripheral pin select feature.

   The following example assumes that there is no connection to the RP127/RG15 pin on the device. The following statements connect Comparator Output 1 to Input Capture 1 (IC1) using RP127.

```
RPINR7bits.IC1R = 127;
/*assign Input Capture 1 to RP127    */
RPOR15bits.RP127R = 0b011000;
/*assign RP127 to Comparator Output 1*/
```

2. The FCLCONx register can be used to map comparator outputs to PWM fault inputs without the use of virtual remapping.

   The following statement will connect Comparator 1 output to Fault Control Signal Source for PWM 1.

```
FCLCON1bits.FLTSRC = 8;
/* value of 0b01000 selects Comp. 1 */
```

3. FHOMEx and FINDXx are not accessible, making the digital filter in the QEI module unusable for any other peripheral besides the QEI. There is no work around.

Affected Silicon Revisions

B1							
X							

5. Module: SPI

When using the frame sync pulse output feature (SPIxCON2<FRMEN> = 1) in Master mode (SPIxCON2<SPIFSD> = 0), the frame sync pulse is not being generated with an active-low pulse (SPIxCON2<FRMPOL> = 0).

Work around

The SS pin is used as the frame sync pulse when the frame sync pulse output feature is used. Mapping the $\overline{SSx}$ input function and output function to the same pad using the PPS feature resolves this issue.

The following code example assigns SPI1 $\overline{SSx}$ input and SPI1 $\overline{SSx}$ output to RP118.

```
RPINR21bits.SS1R  = 118;
/* assign the SPI1 Slave Select Input to
   RP118 */
RPOR13bits.RP118R = 0b000111;
/* assign peripheral output function SPI1
   to RP118 */
```

Affected Silicon Revisions

B1							
X							

6. Module: SPI

When in SPI Slave mode (SPIxCON1<MSTEN> = 0) and using the frame sync pulse output feature (SPIxCON2<FRMEN> = 1) in Slave mode (SPIxCON2<SPIFSD> = 0), the Frame Sync Pulse Edge Select bit must be set to '0' (SPIxCON2 <FRMDLY> = 0)

Work around

There is no work around. The Frame Sync Pulse Edge Select bit cannot be set to produce a frame sync pulse that coincides with the first bit clock.

Affected Silicon Revisions

B1							
X							

7. Module: PWM

When dead-time compensation is enabled (PWMCONx<DTC> = 11) in Edge-Aligned mode (PWMCONx<CAM> = 0), the setting of the DTCP<1:0> bits (PWMCONx<7:6>) and the external signal DTCMPx, determine whether the DTRx register is added to or subtracted from the duty cycle specified by the PDCx or MDC registers.

When DTR is being subtracted from the duty cycle, the resulting duty cycle will be 0% if the programmed duty cycle, minus two times the DTR value, is less than '0'.

Duty Cycle = 0% when (PDCx – 2 DTR) < 0 if MDCS = 0 (PWMCONx<8>)

or

(MDC – 2 DTR) < 0 if MDCS = 1 (PWMCONx<8>)

When DTR is being added to the duty cycle, the resulting duty cycle will be 100% if the programmed duty cycle, plus two times the DTR register, is greater than the period.

Duty Cycle = 100% when (PDCx + 2 DTR) ≥ Period if MDCS = 0 (PWMCONx<8>)

or

(MDC + 2 DTR) ≥ Period if MDCS = 1 (PWMCONx<8>)

The period is specified by the PTPER, STPER or PHASEx registers, depending on the ITB (PWMCON<9>) and MTBS (PWMCONx<3>) bit settings.

Note: The dead-time values, as specified in the ALTDTRx register, are not part of the equations shown above and are still applied when the duty cycle is not forced to 0% or 100%.

Work around

If using dead-time compensation, do not use duty cycle values that are less than two times the DTR value or that are greater than or equal to the period less two times the DTR value.

Affected Silicon Revisions

B1							
X							

8. Module: PWM

When dead-time compensation is enabled (PWMCONx<DTC> = 11) in Center-Aligned mode (PWMCONx<CAM> = 1), the dead time, as specified in the ALTDTRx register, is not being applied to the PWMxH output. The leading and trailing edges of the PWMxL output are extended by one-half the value of the ALTDTRx register, but the PWMxH leading and trailing edges are unaffected.

Work around

Using the values from **Section 14. "High-Speed PWM"** (DS70645), adjust the PWM parameters as follows:

- Subtract one-half of the ALTDTR dead time from PDCx
- Use twice the value for ALTDTR. For example:
  - Frequency of 60 kHz, duty cycle of 50%
  - Desired dead time of 833 ns and dead-time compensation of 833 ns

Using the specified values from **Section 14. "High-Speed PWM"** (DS70645) in the *"dsPIC33E/PIC24E Family Reference Manual"*:

- PHASEx = 1000
- PDCx = 500
- ALTDTR = 833 ns/8.33 ns = 100
- DTR = (833 ns/8.33 ns)/2 = 50

Applying the work around:

- ALTDTR = 2 * 100 = 200
- PDCx = PDCx – 25 = 475

Affected Silicon Revisions

B1							
X							

9. Module: Power System

For this version of silicon, the Brown-out Reset (BOR) must always be enabled.

Work around

Do not disable the BOR by setting BOREN = 0 (FPOR<3>) or by setting SBOREN = 0 (RCON<13>).

Affected Silicon Revisions

B1							
X							

10. Module: Reserved

11. Module: ECAN™

The CANCKS (CiCTRL1<11>) function is reversed.

Work around

Set CiCTRL1<CANCKS> = 0 to obtain an FCAN equal to twice FCY or CiCTRL1<CANCKS> = 1 to obtain an FCAN equal to FCY. This bit must be set to '1' for compatibility with the dsPIC33F or PIC24H. At Reset, it is set to '0'.

Affected Silicon Revisions

B1							
X							

12. Module: ECAN

The ERRIF status flag (CiINTF<5>) does not get set when a CAN error condition occurs, and as a result, an interrupt is not generated even if enabled.

Work around

Use the Invalid Message Interrupt (IVRIF) to inspect the individual error condition status flags TXBO, TXBP, RXBP, TXWAR, RXWAR and EWARN (CiINTF<13:8>) to determine if an error condition has occurred.

Affected Silicon Revisions

B1							
X							

13. Module: USB

While operating in Host mode and attached to a low-speed device through a full-speed USB hub, the host may persistently drive the bus to an SE0 state (both D+/D- as '0'), which would be interpreted as a bus Reset condition by the hub; or the host may persistently drive the bus to a J state, which would make the hub detach condition undetectable by the host.

Work around

Connect low-speed devices directly to the host USB port and not through a USB hub.

Affected Silicon Revisions

B1							
X							

14. Module: USB

In the case where the bus has been Idle for > 3 ms, and the UIDLE interrupt flag is set, if software clears the interrupt flag and the bus remains Idle, the UIDLE interrupt flag will not be set again.

Work around

Software can leave the UIDLE bit set until it has received some indication of bus resumption. (Resume, Reset, SOF or Error).

Note:	Resume and Reset are the only interrupts that should occur following UIDLE assertion. If, at any point in time, the UIDLE bit is set, it should be okay to suspend the USB module (as long as this code is protected by the GUARD and/or ACTPEND logic). Note that this will require software to clear the UIDLE interrupt enable bit to exit the USB ISR (if using interrupt driven code).

Affected Silicon Revisions

B1							
X							

15. Module: DMA

When the DMA is set up for CAN receive, interrupts can sometimes be lost if the DMA is held in an OFF state by the system arbiter. If a CAN receive interrupt occurs while the DMA is waiting for a grant for the previous CAN transaction, this current interrupt will be dropped.

Work arounds

There are two possible work arounds for this issue:

1. Use Dual Port RAM (If available) for target DMA memory; the DMA cannot be held "OFF" when accessing the back side of DPRAM. Only channels set up for CAN receive would need to use DPRAM; all other peripherals can use any RAM.
2. Elevate the system priority of DMA by writing a 0x20 to the MSTRPR (Master Priority) SFR register (address: 0x0058). This will also prevent the DMA from being held "OFF".

Affected Silicon Revisions

B1							
X							

16. Module: UART

When using UTXISEL = 01 (interrupt when last character is shifted out of the Transmit Shift Register), and the final character is being shifted out through the Transmit Shift Register (TSR), the TX interrupt may occur before the final bit is shifted out.

Work around

If it is critical that the interrupt processing occurs only when all transmit operations are complete, after which the following work around can be implemented:

Hold off the interrupt routine processing by adding a loop at the beginning of the routine that polls the transmit shift register empty bit, as shown in Example 2.

Affected Silicon Revisions

B1							
X							

EXAMPLE 2:

```
// in UART2 initialization code
...
U2STAbits.UTXISEL0  = 1;          // Set to generate TX interrupt when all
U2STAbits.UTXISEL1  = 0;          // transmit operations are complete.
...

U2TXInterrupt(void)
{
    while(U2STAbits.TRMT==0);     // wait for the transmit buffer to be empty
    ...                           // process interrupt
```

17. Module: UART

When a Read-Modify-Write operation is performed to set or clear any bit(s) in the UxSTA register while hardware is clearing the UTXBRK bit (UxSTA<11>), the UTXBRK bit may remain set. BSET and BCLR are examples of Read-Modify-Write instructions.

Work around

Wait for the UTXBRK bit to get cleared by hardware, before performing a Read-Modify-Write operation on the UxSTA register.

Affected Silicon Revisions

B1							
X							

18. Module: UART

The Transmitter Write Pointer does not get cleared when the UART module is disabled (UARTEN = 0) and it requires the TXEN bit to be set in order to clear the Write Pointer.

Work around

Do not load data into the TX FIFO (register) before setting the TXEN bit.

Affected Silicon Revisions

B1							
X							

19. Module: I²C™

When a Read-Modify-Write operation is performed to set or clear any bit(s) in the I2CxCON register while hardware is clearing the ACKEN bit (I2CxCON<4>), the ACKEN bit may remain set. BSET and BCLR are examples of Read-Modify-Write instructions.

Work around

Wait for the ACKEN bit to get cleared by hardware before performing a Read-Modify-Write operation on the I2CxCON register.

Affected Silicon Revisions

B1							
X							

20. Module: ADC

The ADC Conversion Status bit, DONE (ADxCON1<0>), does not indicate completion of conversion when External Interrupt is selected as the ADC trigger source (ADxCON1<SSRC> = 1).

Work around

Use ADC interrupt or poll ADxIF (in the IFSx registers) bit to determine the completion of conversion.

Affected Silicon Revisions

B1							
X							

21. Module: PMP

Note: This silicon issue applies only to dsPIC33EPXXXMU814 and PIC24EPXXXMC814 devices.

When PTEN14 = 1 (PMAEN<14>), the PMA<14> or PMCS1 functionality is present on the PMCS1/RK11 (pin 94) and RJ14 (pin 21).

When PTEN15 = 1 (PMAEN<15>), the PMA<15> or PMCS2 functionality is present on the PMCS2/ RK12 (pin 93) and RJ15 (pin 22).

Work around

None.

Affected Silicon Revisions

B1							
X							

22. Module: Flash Memory

Note: This silicon issue applies only to In-Circuit Serial Programming™ (ICSP™) mode.

If code or write protection is enabled on either the General Segment or Auxiliary Segment, neither segment can be read by the programmer. Code or write protection is enabled for the General Segment when the GSS (FGS<1>) or GWRP (FGS<0>) bits are '0'. Code or write protection is enabled for the Auxiliary Segment when the APL (FAS<1>) or AWRP (FAS<0>) bits are '0'.

Work around

None.

Affected Silicon Revisions

B1							
X							

23. Module: Flash Memory

The processor stalls while performing Run-Time Self-Programming (RTSP) erase and write operations on the same Flash segment (General or Auxiliary) from which code is being executed. The stall mechanism does not always function properly and can cause unexpected behavior.

Work arounds

Two options are available to avoid this issue:

1. If you are required to execute code, including an Interrupt Service Routine (ISR), from the same segment (either General or Auxiliary) that you are performing RTSP operations on, you must disable interrupts until the erase or programming operation is complete (see Example 3).
2. If possible, structure your project such that RTSP operations are performed on a segment from which no code is being executed. For example, place all of your executable code in the General Segment and all reprogrammable data in the Auxiliary Segment or vice versa. This solution has the added advantage that no CPU stalls occur since the programming operation is not being performed on the same segment that is executing the code.

EXAMPLE 3:

```
; Load write latches if programming
    …
; Setup NVMCON register to erase or program
    as required
    …
; Disable interrupts
    PUSH    SR
    MOV     #0x00E0, W0
    IOR     SR
; Write the KEY sequence
    MOV     #0x55, W0
    MOV     W0, NVMKEY
    MOV     #0xAA, W0
    MOV     W0, NVMKEY
; Start the programming sequence
    BSET    NVMCON, #15
; Insert two NOPs after programming
    NOP
    NOP
; Wait for operation to complete
prog_wait:
    BTSC    NVMCON, #15
    BRA     prog_wait
; Re-enable interrupts,
    POP     SR
```

Affected Silicon Revisions

B1							
X							

24. Module: Power System

The VREGSF bit functions as documented, but will always read as '0'. Because of the Read-Modify-Write process, any BSET or BCLR instruction of the RCON register will also write a '0' to the VREGSF bit.

Work around

If the VREGSF bit is intended to be set to a '1', the user must also write a '1' to the VREGSF bit when setting or clearing any other bit in the RCON register.

Affected Silicon Revisions

B1							
X							

25. Module: PWM

When operating in Edge-Aligned Complimentary mode, if the duty cycle (PDCx) becomes less than the alternate dead time (ALTDTRx), the dead time on the PWMs will become zero.

Work around

Ensure that the duty cycle (PDCx) always meets the following condition: PDCx > (ALTDTRx – 1).

Affected Silicon Revisions

B1							
X							

26. Module: QEI

In QEI mode (QEIxCON<CMM> = 00), the Index Counter registers (INDXxCNTH and INDXxCNTL) cannot be relied upon to increment when the last known direction was positive and an index pulse occurs. The Index register can decrement even if the last known direction was positive. This does not apply to external clock or internal timer QEI modes.

Work around

The index event can be used to implement a software counter. The direction could be determined by comparing the current POSxCNT value to that of the previous index event.

Affected Silicon Revisions

B1							
X							

27. Module: QEI

When Modulo Count mode (Mode 6) is selected for the position counter (QEIxCON<PIMOD> = 110) and the counter direction is set to negative (QEIxCON<CNTPOL> = 1), the functions of the QEIxLEC and QEIxGEC registers are reversed.

Work around

When using Modulo Count mode in conjunction with a negative count direction (polarity) use the QEIxLEC register as the upper count limit and the QEIxGEC register as the lower count limit.

Affected Silicon Revisions

B1							
X							

28. Module: CPU

Note:	This silicon issue applies only to dsPIC33EPXXXGP/MC/MU806/810/814 devices.

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

B1							
X							

29. Module: PWM

The PWM module can operate with variable period, duty cycle, dead-time and phase values. The master period and other timing parameters can be updated in the same PWM cycle. With immediate updates disabled, the new values should take effect at the start of the next PWM cycle.

As a result of this issue, the updated master period takes effect on the next PWM cycle, while the update of the additional timing parameter is delayed by one PWM cycle. The parameters affected by this erratum are as follows:

Master Period registers – update effective on the next PWM cycle:

- PTPER – if PWMCONx<MTBS> = 0
- STPER – if PWMCONx<MTBS> = 1

Additional PWM timing parameters – update effective one PWM cycle after master period update:

- Duty cycle – PDCx, SDCx and MDC registers
- Phase – PHASEx or SPHASEx registers
- Dead time – DTRx and ALTDTRx registers and dead-time compensation signals
- Clearing of current-limit and Fault conditions and application of External Period Reset signal

Work around

If the application requires the master period and other parameters to be updated at the same time, enable both immediate updates:

- PTCON<EIPU> = 1 – to enable immediate period updates
- PWMCONx<IUE> = 1 – to enable immediate updates of additional parameters listed above

Enabling immediate updates will allow updates to the master period and the other parameters to take effect immediately after writing to the respective registers.

Affected Silicon Revisions

B1							
X							

30. Module: ECAN™

When DMA is used with the ECAN module and the CPU and DMA write to a ECAN special function register (SFR) at the same time, the DMAC error trap is not occurring. In addition, neither the PWCOLx bit of the DMAPWC SFR or the DMACERR bit of the INTCON1 SFR are being set. Since the PWCOLx bit is not set, subsequent DMA requests to that channel are not ignored.

Work around

There is no work around; however, under normal circumstances, this situation should not arise. When DMA is used with the ECAN module, the application should not be writing to the ECAN SFRs.

Affected Silicon Revisions

B1							
X							

31. Module: Auxiliary Flash

When a device is set to obtain the RESET instruction from Auxiliary Flash location, 0x7FFFFC (RSTPRI = 0), and Auxiliary Flash code protection is enabled using the FAS register, the device does not execute the application code after a Reset. This configuration causes a security trap resulting in a Reset.

Work around

The work around is dependent on Errata Issue 22 (Flash Memory) provided that the application will accept both segments being code-protected. The Auxiliary Flash Reset vector (RSTPRI = 0) does function when code protection for the Auxiliary Segment is not enabled using the FAS register. Enabling code protection on the General Segment using the FGS register protects the General Segment and, because of Errata Issue 22, the Auxiliary Segment is protected as well.

Affected Silicon Revisions

B1							
X							

32. Module: Auxiliary Flash

When executing code in the Auxiliary Segment, all interrupts and traps should vector through the single auxiliary interrupt vector located at address, 0x7FFFFA; however, the address error trap is not. Instead, it vectors to the address error trap vector located at address, 0x000006, in the General Segment.

Work around

There is no universal work around. In a bootloader application, if the General Segment is erased and an address error trap occurs, a Reset will result. If the application routinely executes code from both segments, the error address trap could then be made to handle either circumstance. A flag could be set to indicate code is executing from the Auxiliary Segment and tested by the address error trap handler.

Affected Silicon Revisions

B1							
X							

33. Module: Output Compare

Under certain circumstances, an output compare match may cause the interrupt flag (OCxIF) to become set prior to the Change-of-State (COS) of the OCx pin. This has been observed when all of the following are true:

- The module is in One-Shot mode (OCM<2:0> = 001, 010 or 100);
- One of the timer modules is being used as the time base; and
- A timer prescaler other than 1:1 is selected

If the module is reinitialized by clearing OCM<2:0> after the One-Shot mode compare, the OCx pin may not be driven as expected.

Work around

After OCxIF is set, allow an interval (in CPU cycles) of at least twice the prescaler factor to elapse before clearing OCM<2:0>. For example, for a prescaler value of 1:8, allow 16 CPU cycles to elapse after the interrupt.

Affected Silicon Revisions

B1							
X							

34. Module: ADC

Selecting the same ANx input (AN0 or AN3) for CH0 and CH1 to achieve a 1.1 Msps sampling rate results in erroneous readings for CH1.

Work around

Bring the analog signal into the device using both AN0 and AN3, connect externally, and then assign one input to CH0 and the other to CH1.

If selecting AN0 on CH1 (CH123Sx = 0), select AN3 on CH0 (CH0Sx = 3). Conversely, if selecting AN3 on CH1 (CH123Sx = 1), select AN0 on CH0 (CH0Sx = 0).

Affected Silicon Revisions

B1							
X							

35. Module: CAN

A transmit-interrupt event (even if it is disabled as an interrupt source) that happens immediately after the reception-interrupt event will cause the ICODE bits to point to the transmit buffer instead of the receive buffer.

Work around

Use FILHIT, rather than ICODE, when processing an Rx interrupt.

Affected Silicon Revisions

B1							
X							

36. Module: PMP

Port pins RH8-RH15 are not available for use when the PMP module is enabled. These pins are available only on certain package types.

Work around

Do not enable PMP mode if any of these pins are needed for I/O.

Affected Silicon Revisions

B1							
X							

37. Module: Core

When interrupt nesting is enabled by clearing the NSTDIS bit (INTCON1<15> = 0), an interrupt that occurs during the last two instructions of the DO loop can end it prematurely. The DCOUNT is incorrectly decremented twice when:

- an interrupt occurs during the last two instructions of a DO loop, and
- the second higher priority interrupt occurs exactly four instruction cycles later

Work around

Disable interrupt nesting by setting the NSTDIS bit (INTCON1<15> = 1).

Alternatively, for interrupts of priority levels up to 6, use the DISI instruction to disable the nested interrupts while executing the last two instructions of the DO loop.

Affected Silicon Revisions

B1							
X							

38. Module: Core

An address error trap may occur if the variable exception processing latency is enabled by setting the VAR bit (CORCON<15> = 1), and the same data variables are modified both within and outside the Interrupt Service Routine.

Work around

Enable the Fixed Interrupt Latency mode by clearing the VAR bit (VAR (CORCON<15>) = 0).

Affected Silicon Revisions

B1							
X							

39. 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.

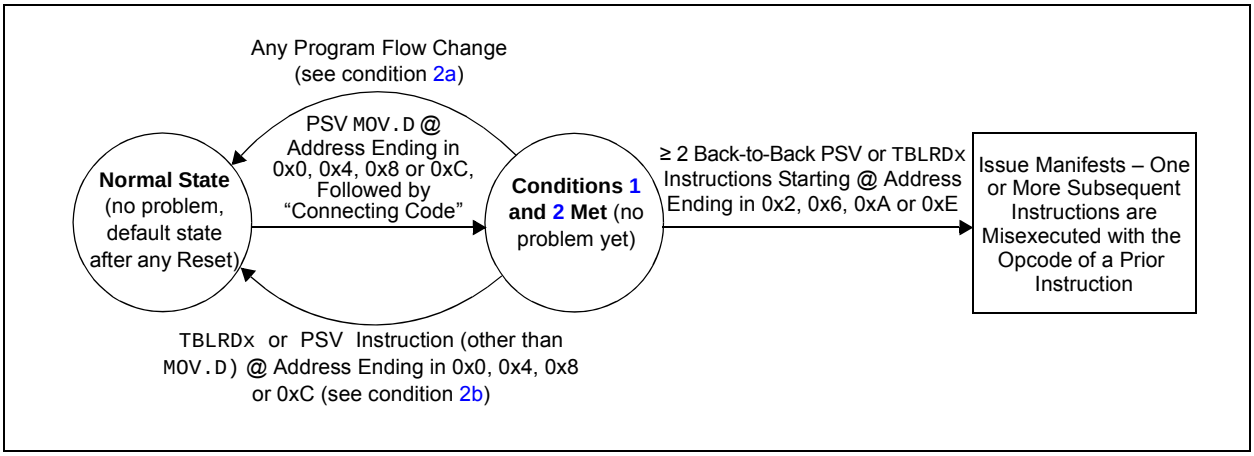
When two or more data Flash read instructions (via Program Space Visibility (PSV) read or table read) are executed back-to-back, one or more subsequent instructions can be misexecuted when all of the conditions in Table 3 occur.

TABLE 3: 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 1 provides an example of the effective behavior.

FIGURE 1: 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

B1							
X							

Data Sheet Clarifications

The following typographic corrections and clarifications are to be noted for the latest version of the device data sheet (DS70616**G**):

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

1. Module: I/O Ports

The following paragraph has been added to **Section 11.4.4 "Input Mapping"**:

External digital signals can be mapped to input only PPS pins (RPIn) only if the corresponding ANSx bit is set to '1'. If the ANSx bit associated with that pin is a '0', even though the input signal is digital, the I/O buffer will not recognize the signal.

2. Module: Enhanced CAN (ECAN)

The following section has been added:

21.3 Writing to the CAN CxFCTRL SFR

There are two rules that need to be observed when writing data to the CxFCTRL register in the CAN module.

1. The register cannot be written with a 16-bit word. The register must be written as two separate operations to the FSA<4:0> portion and the DBABS<2:0> portion.
2. The FSA<4:0> bits must be written first and then followed by the DMABS<2:0> bits.

EXAMPLE 21-1: ILLEGAL OPERATIONS

```
C1FCTRL = 0x2001;           // NOT ALLOWED to
                            // write entire
                            // 16-bits in 1
                            // operation


C1FCTRLbits.DMABS = 2;      // WRONG ORDER
C1FCTRLbits.FSA = 1;
```

3. Module: dsPIC33EPXXX(GP/MC/MU)806/810/814 and PIC24EPXXX(GP/GU)810/814 Product Families

In **Table 1**, the PIC24EP512GP806 Program Flash Memory has been changed from 586 to 536 Kbytes.

4. Module: Pin Diagrams

In the dsPIC33EP512MC806 pin diagram, SCLI has been changed to SCL1, and SDA1 and SCL1 have been shaded to show that the pins are up to 5V tolerant.

In the dsPIC33EP512GP806/PIC24EP512GP806 pin diagram, SCLI has been changed to SCL1.

5. Module: Guidelines for Getting Started with 16-Bit Digital Signal Controllers and Microcontrollers

In **Section 2.7 Oscillator Value Conditions on Device Start-up**, the first paragraph has been updated. The updated text is shown below in **bold**.

If the PLL of the target device is enabled and configured for the device start-up oscillator, the maximum oscillator source frequency must be limited to **4.8 MHz < FIN < 13.6 MHz** to comply with device PLL start-up conditions.

6. Module: Memory Organization

In **Table 4-64: PORTF Register Map**, all variables for Bit 6 have been updated as shown below in **bold**.

Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3
—	—	**TRISF6**	TRISF5	TRISF4	TRISF3
—	—	**RF6**	RF5	RF4	RF3
—	—	**LATF6**	LATF5	LATF4	LATF3
—	—	**ODCF6**	ODCF5	ODCF4	ODCF3
—	—	**CNIEF6**	CNIEF5	CNIEF4	CNIEF3
—	—	**CNPUF6**	CNPUF5	CNPUF4	CNPUF3
—	—	**CNPDF6**	CNPDF5	CNPDF4	CNPDF3
—	—	—	—	—	—

7. Module: Oscillator Configuration

In **Register 9-1: OSCCON: Oscillator Control Register**, the following Note has been added to the CF bit:

Note 4: The CF bit is writable. Writing a '1' will cause a clock fail trap.

8. Module: Oscillator Configuration

Bit 14, in **Register 9-5: ACLKCON3: Auxiliary Clock Control Register 3**, has changed from Unimplemented to:

APLLCK: APLL Locked Status bit (read-only)

1 = Indicates that Auxiliary PLL is in lock
0 = Indicates that Auxiliary PLL is not in lock

9. Module: Output Compare

In **Register 15-2: OCxCON2: Output Compare x Control Register 2**, the SYNCSEL = 11111 definition for OCxCON2 has been changed to:

11111 = OCxRS compare event is used for synchronization

10. Module: USB On-The-Go (OTG) Module (dsPIC33EPXXXMU8XX and PIC24EPGU8XX Devices Only)

The following paragraph has been added to **Section 22.0 "USB On-The-Go (OTG) Module"**:

The USB module requires a FCYC frequency of 24 MHz (minimum) in order to properly function. If the USB module is not used, this requirement can be ignored.

11. Module: Comparator Module

In **Figure 25-1: Comparator I/O Operating Modes**, the 2.20V selection for BGSEL has changed from 2.20V to 1.20V.

12. Module: Comparator Module

In **Register 25-6: CVRCON: Comparator Voltage Reference Control Register**, the BGSEL<1:0> voltage has changed as shown in **bold** below:

BGSEL<1:0>: Band Gap Reference Source Select bits

11 = IVREF = VREF+(2)
10 = IVREF = 0.20V (nominal)
01 = IVREF = 0.60V (nominal)
00 = IVREF = **1.20V** (nominal)

13. Module: Electrical Characteristics

Table 32-7: DC Characteristics: Power-Down Current (IPD) has been updated. The DC60b Maximum Power-Down Current has been changed from 500 µA to 750 µA and the DC60c Maximum Power-Down Current has been changed from 3000 µA to 4000 µA.

14. Module: High-Speed PWM Module (dsPIC33EPXXX(MC/MU)8XX Devices Only)

The PWM resolution, shown under High-Speed PWM on page 1, has been changed from 8.32 ns to 7.14 ns.

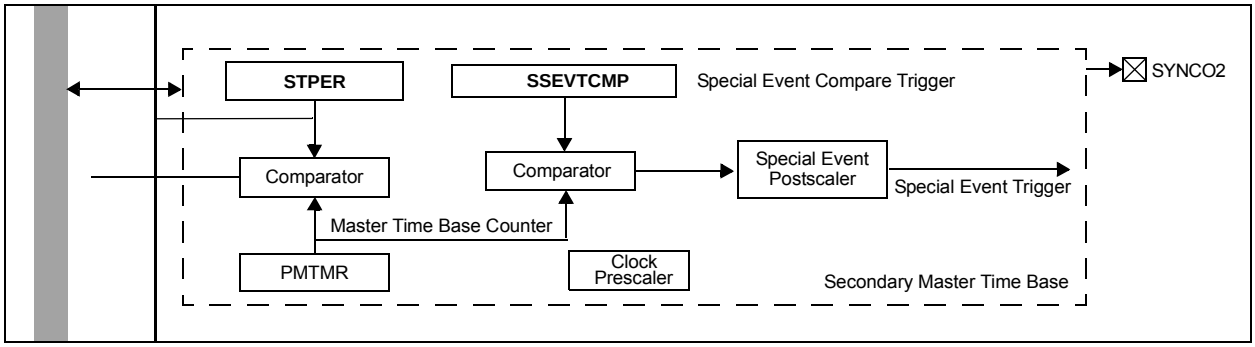
15. Module: High-Speed PWM Module (dsPIC33EPXXX(MC/MU)8XX Devices Only)

The period, duty cycle, phase shift and dead-time resolution, shown in the bulleted list in **Section 16.0 "High Speed PWM Module"**, has changed from 8.32 ns to 7.14 ns.

16. Module: High-Speed PWM Module (dsPIC33EPXXX(MC/MU)8XX Devices Only)

In **Figure 16-2: High-Speed PWM Module Register Interconnection Diagram**, the Secondary Master Time Base section has been updated. The updated text is shown in **bold** in Figure 16-2 below.

FIGURE 16-2: HIGH-SPEED PWM MODULE REGISTER INTERCONNECTION DIAGRAM



APPENDIX A: REVISION HISTORY

Rev A Document (6/2011)

Initial release of this document; issued for revision B1 silicon.

Includes silicon issues 1 (I/O Multiplexer), 2-3 (CPU), 4 (PPS), 5-6 (SPI), 7-8 (PWM), 9 (Power System), 10 (Reserved), 11-12 (ECAN™), 13-14 (USB), 15 (DMA) 16-18 (UART) and 19 (I^2C™).

Rev B Document (11/2011)

Updated issues 2 (CPU), 7 (PWM), 8 (PWM), 12 (ECAN™) and 14(USB).

Added issues 20 (ADC), 21 (PMP), 22 (Flash Memory), 23 (Flash Memory), 24 (Power System), 25 (PWM), 26 (QEI), 27 (QEI), 28 (CPU) and 29 (PWM).

Rev C Document (3/2012)

Added silicon issues 30 (ECAN™), 31 (Auxiliary Flash), 32, (Auxiliary Flash), 33 (Output Compare) and 34 (ADC).

Rev D Document (2/2015)

Added data sheet clarifications 1 (I/O Ports) and 2 (Enhanced CAN (ECAN)).

Rev E Document (3/2015)

Updated data sheet clarification 1 (I/O Ports).

Rev F Document 10/2015

Added silicon issues 35 (CAN), 36 (PMP), 37 (Core) and 38 (Core). Added data sheet clarifications 3 (dsPIC33EPXXX(GP/MC/MU)806/810/814 and PIC24EPXXX(GP/GU)810/814 Product Families), 4 (Pin Diagrams), 5 (Guidelines for Getting Started with 16-Bit Digital Signal Controllers and Microcontrollers), 6 (Memory Organization), 7 (Oscillator Configuration), 8 (Oscillator Configuration), 9 (Output Compare), 10 (USB On-The-Go (OTG) Module (dsPIC33EPXXXMU8XX and PIC24EPGU8XX Devices Only)), 11 (Comparator Module), 12 (Comparator Module), 13 (Electrical Characteristics), 14 (High-Speed PWM Module (dsPIC33EPXXX(MC/MU)8XX Devices Only), 15 (High-Speed PWM Module (dsPIC33EPXXX(MC/MU)8XX Devices Only)) and 16 (High-Speed PWM Module (dsPIC33EPXXX(MC/MU)8XX Devices Only)).

Rev G Document 8/2017

Updated silicon issue 3 (CPU) and 28 (CPU).

Added silicon issue 39 (CPU).

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: 978-1-5224-2094-1

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

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon

Hong Kong
Tel: 852-2943-5100
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

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

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

China - Hangzhou
Tel: 86-571-8792-8115
Fax: 86-571-8792-8116

China - Hong Kong SAR
Tel: 852-2943-5100
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-3326-8000
Fax: 86-21-3326-8021

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8864-2200
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

ASIA/PACIFIC

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-3019-1500

Japan - Osaka
Tel: 81-6-6152-7160
Fax: 81-6-6152-9310

Japan - Tokyo
Tel: 81-3-6880- 3770
Fax: 81-3-6880-3771

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-5778-366
Fax: 886-3-5770-955

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

Taiwan - Taipei
Tel: 886-2-2508-8600
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

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

France - Saint Cloud
Tel: 33-1-30-60-70-00

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: 47-7289-7561

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