dsPIC33FJXXXMCX06/X08/X10 Rev. A2/A3 Silicon Errata

The dsPIC33FJXXXMCX06/X08/X10 (Rev. A2/A3) devices you received were found to conform to the specifications and functionality described in the following documents:

- DS70165 – "*dsPIC33F Family Data Sheet"*
- DS70157 – "*dsPIC30F/33F Programmer's Reference Manual"*
- DS70046 – *"dsPIC30F Family Reference Manual"*

The exceptions to the specifications in the documents listed above are described in this section. The specific devices for which these exceptions are described are listed below:

- dsPIC33FJ64MC506
- dsPIC33FJ64MC508
- dsPIC33FJ64MC510
- dsPIC33FJ64MC706
- dsPIC33FJ64MC710
- dsPIC33FJ128MC506
- dsPIC33FJ128MC510
- dsPIC33FJ128MC706
- dsPIC33FJ128MC708
- dsPIC33FJ128MC710
- dsPIC33FJ256MC510
- dsPIC33FJ256MC710

dsPIC33FJXXXMCX06/X08/X10 Rev. A2/A3 silicon is identified by performing a "Reset and Connect" operation to the device using MPLAB® ICD 2 with MPLAB IDE v7.40 or later. The output window will show a successful connection to the device specified in *Configure>Select Device*. The resulting DEVREV register values for Rev. A2/A3 silicon are 0x3002 and 0x3004, respectively.

The errata described in this document will be addressed in future revisions of silicon.

Silicon Errata Summary

The following list summarizes the errata described in further detail through the remainder of this document:

1. Doze Mode

   When Doze mode is enabled, any writes to a peripheral SFR can cause other updates to that register to cease to function for the duration of the current CPU clock cycle.

2. 12-bit Analog-to-Digital Converter (ADC) Module

   For this revision of silicon, the 12-bit ADC module INL, DNL and signal acquisition time parameters are not within the published data sheet specifications.

3. 10-bit ADC Module

   For this revision of silicon, the 10-bit ADC module DNL, conversion speed and signal acquisition time parameters are not within the published data sheet specifications.

4. DMA Module: Interaction with `EXCH` Instruction

   The `EXCH` instruction does not execute correctly when one of the operands contains a value equal to the address of the DMAC SFRs.

5. `DISI` Instruction

   The `DISI` instruction will not disable interrupts if a `DISI` instruction is executed in the same instruction cycle that the `DISI` counter decrements to zero.

6. Motor Control PWM

   There is a glitch in the PWMxL signal in Single-Shot mode with complementary output. Another glitch occurs when resuming from a Fault condition in Free-Running mode with complementary output.

7. Output Compare Module in PWM Mode

   The output compare module will miss one compare event when the duty cycle register value is updated from 0x0000 to 0x0001.

8. SPI Module in Frame Master Mode

   The SPI module will fail to generate frame synchronization pulses in Frame Master mode if FRMDLY = 1.

9. SPI Module in Slave Select Mode

   The SPI module Slave Select functionality will not work correctly.

10. SPI Module

    The SMP bit does not have any effect when the SPI module is configured for a 1:1 prescale factor in Master mode.

11. ECAN™ Module

    ECAN transmissions may be incorrect if multiple transmit buffers are simultaneously queued for transmission.

12. ECAN Module

    Under specific conditions, the first five bits of a transmitted identifier may not match the value in the transmit buffer ID register.

13. ECAN Module Loopback Mode

    The ECAN module (ECAN1 or ECAN2) does not function correctly in Loopback mode.

14. I^2C™ Module

    The Bus Collision Status bit does not get set when a bus collision occurs during a Restart or Stop event.

15. INT0, ADC and Sleep/Idle Mode

    ADC event triggers from the INT0 pin will not wake-up the device from Sleep or Idle mode if the SMPI bits are non-zero.

16. Doze Mode and Traps

    The address error trap, stack error trap, math error trap and DMA error trap will not wake-up a device from Doze mode.

17. JTAG Programming

    JTAG programming does not work.

18. UART

    With the parity option enabled, a parity error may occur if the Baud Rate Generator (BRG) contains an odd value.

19. UART

    The Receive Buffer Overrun Error Status bit may get set before the UART FIFO has overflowed.

20. UART

    UART receptions may be corrupted if the BRG is set up for 4x mode.

21. UART

    The UTXISEL0 bit is always read back as zero.

22. UART

    The auto-baud feature may not calculate the correct baud rate when the BRG is set up for 4x mode.

23. UART

    With the auto-baud feature selected, the sync break character (0x55) may be loaded into the FIFO as data.

24. I^2C Module

    A write collision does not prevent the transmit register from being written.

25. I^2C Module

    The ACKSTAT bit only reflects the received ACK/NACK status for master transmissions, but not for slave transmissions.

26. I^2C Module

    The D_A Status bit does not get set on a slave write to the transmit register.

27. Traps and Idle Mode

    If a clock failure occurs when the device is in Idle mode, the oscillator failure trap does not vector to the Trap Service Routine (TSR).

28. $\overline{MCLR}$ Wake-up from Sleep Mode

    An $\overline{MCLR}$ wake-up from Sleep mode does not wait for the on-chip voltage regulator to power-up.

29. ECAN Module

    The C1RXOVF2 and C2RXOVF2 registers always read back as 0x0000.

30. FRC Oscillator

    Internal FRC accuracy parameters are not within the published data sheet specifications.

31. Quadrature Encoder Interface (QEI) Module

    The QEI module does not generate an interrupt in a particular overflow condition.

32. Device ID Register

    The content of the Device ID register changes from the factory programmed value.

33. SPI Module

    SPI1 functionality for pin 34 (U1RX/SDI1/RF2) is erroneously enabled by the SPI2 module.

34. UART

    The auto-baud feature measures baud rate inaccurately for certain baud rate and clock speed combinations.

35. Motor Control PWM (Faults in Latched mode)

    Subsequent faults in the same timer cycle are missed during a latched fault.

36. Motor Control PWM (Fault-driven Wake-up)

    Fault-driven Wake-up from Idle does not function.

37. DMA Module

    DMA data transfers that are active in Single-Shot mode while the device is in Sleep or Idle mode may result in more data transfers than expected.

38. Doze Mode and Traps

    A DMA error trap may not be generated when the device is in Doze mode.

39. ECAN Module

    If receive masking is disabled for a receive filter, the module behaves as if DeviceNet™ Addressing were enabled.

40. Output Compare Module

    In Dual Compare Match mode, the OCx output is not reset when the OCxR and OCxRS registers are loaded with values having a difference of 1.

41. UART

    When the UART is in 4x mode (BRGH = 1) and using two Stop bits (STSEL = 1), it may sample the first Stop bit instead of the second one.

42. UART

    When an auto-baud is detected, the receive interrupt may occur twice.

43. Motor Control PWM – PWM Generator 2

    The PWM2H and PWM2L outputs do not function if the FLTBCON register is written while the $\overline{\text{FLTB}}$ pin is not held high.

44. Motor Control PWM – PWM Counter Register

    PTMR does not keep counting down after halting code execution in Debug mode.

45. DMA

    NULL Data Peripheral Write mode for the DMA channel does not function.

46. DMA

    DMA request Fault condition does not generate a DMA error trap.

47. DMA

    DMA channel writes an additional NULL value to the peripheral register.

48. REPEAT Instruction

    Any instruction executed inside a REPEAT loop that produces a Read-After-Write stall condition, results in the instruction being executed fewer times than was intended.

49. FRC Oscillator

    For certain values of the TUN<5:0> bits (OSCTUN<5:0>), the resultant frequencies are incorrect.

50. UART Module

    The 16x baud clock signal on the BCLK pin is present only when the module is transmitting.

51. SPI Module

    The SPIxCON1 DISSCK bit does not influence port functionality.

52. I^2C Module

    The BCL bit in I2CSTAT can be cleared only with 16-bit operation and can be corrupted with 1-bit or 8-bit operations on I2CSTAT.

53. I^2C Module: 10-bit addressing mode

    When the I^2C module is configured for 10-bit addressing using the same address bits (A10 and A9) as other I^2C device A10 and A9 bits may not work as expected.

The following sections describe the errata and work around to these errata, where they may apply.

1. Module: Doze Mode

Enabling Doze mode slows down the CPU but allows peripherals to run at full speed. When the CPU clock is slowed down by enabling Doze mode (CLKDIV<11> = 1), any writes to a peripheral SFR can cause other updates to that register to cease to function for the duration of the current CPU clock cycle. This is only an issue if the CPU attempts to write to the same register as a peripheral while in Doze mode.

For instance, if the ADC module is active and Doze mode is enabled, the main program should avoid writing to ADCCONx registers because these registers are being used by the ADC module. If the CPU does make writes before the ADC module does, then any attempts by the ADC module to write to these registers will fail.

Work around

In Doze mode, avoid writing code that will modify SFRs which may be written to by enabled peripherals.

2. Module: 12-bit ADC

When the ADC module is configured for 12-bit operation, the specifications in the data sheets are not met.

Work around

Implement the ADC module as an 11-bit ADC with a maximum conversion rate of 300 ksps.

1. The specifications provided below reflect 11-bit ADC operation. RIN source impedance is recommended as 200 ohms, and sample time is recommended as 3 TAD to ensure compatibility on future enhanced ADC modules. Missing codes are possible every 2^7 codes.
2. When used as a 10-bit ADC, the INL is <±2 Least Significant Bytes (LSBs), and DNL is <±1 LSB with no missing codes. Maximum conversion rate is 300 ksps.

TABLE 1: ADC PERFORMANCE (11-BIT OPERATION)

Param No.	Symbol	Min	Typical	Max	Units	Conditions
AD17	RIN	—	—	200	Ohm	12-bit
ADC Accuracy – Measurements taken with External VREF+/VREF-						
AD20a	Nr	—	12 bits	—	Bits	—
AD21a	INL	-2	—	2	LSB	—
AD22a	DNL	-1.5	—	1	LSB	—
AD23a	GERR	1	5	10	LSB	—
AD24a	EOFF	1	3	6	LSB	—
ADC Accuracy – Measurements taken with Internal VREF+/VREF-						
AD21aa	INL	-2	—	2	LSB	—
AD22aa	DNL	-1.5	—	1	LSB	—
AD23aa	GERR	5	10	20	LSB	—
AD24aa	EOFF	3	6	15	LSB	—
Dynamic Performance						
AD33a	FNYQ	—	—	150	KHz	—
AD34a	ENOB	9.5	9.6	10.4	Bits	—
ADC Conversion Rate						
AD56a	FCNV	—	—	300	ksps	—
AD57a	TSAMP	—	3 TAD	—	—	—

3. Module: 10-bit ADC

When the ADC module is configured for 10-bit operation, the specifications in the data sheet are not met for operation above 500 ksps.

For 500 ksps, the module meets specifications except for Gain and Offset parameters AD23bb and AD24bb.

For 600 ksps operation, the module specifications are shown in Table 2.

Work around

None. Future versions of the silicon will support the ADC performance stated in the data sheet.

TABLE 2: 600 KSPS OPERATION

Param No.	Symbol	Min	Typical	Max	Units	Conditions
AD17	RIN	—	—	200	Ohm	10-bit
ADC Accuracy – Measurements taken with External VREF+/VREF-						
AD20b	Nr	—	10 bits	—	Bits	—
AD21b	INL	-2	—	2	LSB	—
AD22b	DNL	-1.5	—	2	LSB	—
AD23b	GERR	1	3	6	LSB	—
AD24b	EOFF	1	2	5	LSB	—
ADC Accuracy – Measurements taken with Internal VREF+/VREF-						
AD21bb	INL	-2	—	2	LSB	—
AD22bb	DNL	-1.5	—	2	LSB	—
AD23bb	GERR	1	6	12	LSB	—
AD24bb	EOFF	2	5	10	LSB	—
Dynamic Performance						
AD33b	FNYQ	—	—	300	KHz	—
AD34b	ENOB	8.5	9.7	9.8	Bits	—
ADC Conversion Rate						
AD56b	FCNV	—	—	600	ksps	—
AD57b	TSAMP	—	3 TAD	—	—	—

4. Module: DMA Module: Interaction with EXCH Instruction

The EXCH instruction does not execute correctly when either of the two operands is numerically equal to the address of any of the DMAC SFRs for this revision of silicon.

Work around

If writing source code in assembly, the recommended fix is to replace:

```
EXCH Wsource, Wdestination
```

with:

```
PUSH Wdestination
MOV Wsource, Wdestination
POP Wsource
```

If using the MPLAB C30 C compiler, specify the compiler option, -merrata=exch (*Project>Build Options>Projects>MPLAB C30>Use Alternate Settings*)

5. Module: DISI Instruction

When a user executes a DISI #7, for example, this will disable interrupts for 7 + 1 cycles (7 + the DISI instruction itself). In this case, the DISI instruction uses a counter which counts down from 7 to 0. The counter is loaded with 7 at the end of the DISI instruction.

If the user code executes another DISI on the instruction cycle where the DISI counter has become zero, the new DISI count is loaded, but the DISI state machine does not properly re-engage and continue to disable interrupts. At this point, all interrupts are enabled. The next time the user code executes a DISI instruction, the feature will act normally and block interrupts.

In summary, it is only when a DISI execution is coincident with the current DISI count = 0, that the issue occurs. Executing a DISI instruction before the DISI counter reaches zero will not produce this error. In this case, the DISI counter is loaded with the new value, and interrupts remain disabled until the counter becomes zero.

Work around

When executing multiple DISI instructions within the source code, make sure that subsequent DISI instructions have at least one instruction cycle between the time that the DISI counter decrements to zero and the next DISI instruction. Alternatively, make sure that subsequent DISI instructions are called before the DISI counter decrements to zero.

6. Module: Motor Control PWM

Devices in the motor control family have a glitch in the PWMxL signal under certain conditions. The glitch is a brief high pulse during the low portion of the duty cycle. This error occurs when the module is configured in Single-Shot mode (PTMOD<1:0> = 01) with complementary output. It also occurs when resuming from a Fault condition in Free-Running mode (PTMOD<1:0) = 00) with complementary output.

Work around

None.

7. Module: Output Compare Module in PWM Mode

The output compare module will miss a compare event when the current duty cycle register (OCxRS) value is 0x0000 (0% duty cycle) and the OCxRS register is updated with a value of 0x0001. The compare event is missed only the first time a value of 0x0001 is written to OCxRS, and the PWM output remains low for one PWM period. Subsequent PWM high and low times occur as expected.

Work around

None. If the current OCxRS register value is 0x0000, avoid writing a value of 0x0001 to OCxRS. Instead, write a value of 0x0002; however, in this case the duty cycle will be slightly different from the desired value.

8. Module: SPI Module in Frame Master Mode

The SPI module will fail to generate frame synchronization pulses when configured in the Frame Master mode if the start of data is selected to coincide with the start of the frame synchronization pulse (FRMEN = 1, SPIFSD = 0, FRMDLY = 1). However, the module functions correctly in Frame Slave mode, and also in Frame Master mode if FRMDLY = 0.

Work around

If DMA is not being used, manually drive the $\overline{SSx}$ pin (x = 1 or 2) high using the associated PORT register, and then drive it low after the required 1 bit-time pulse-width. This operation needs to be performed when the transmit buffer is written.

If DMA is being used, and if no other peripheral modules are using DMA transfers, use a timer interrupt to periodically generate the frame synchronization pulse (using the method described above) after every 8- or 16-bit periods (depending on the data word size, configured using the MODE16-bit).

If FRMDLY = 0, no work around is needed.

9. Module: SPI Module in Slave Select Mode

The SPI module Slave Select functionality (enabled by setting SSEN = 1) will not function correctly. Whether the $\overline{SSx}$ pin (x = 1 or 2) is high or low, the SPI data transfer will be completed and an interrupt will be generated.

Work around

If DMA is not being used, manually poll the $\overline{SSx}$ pin state in the SPI interrupt by reading the associated LAT bit:

- If the LAT bit is '0', then perform the required data read/write.
- If the LAT bit is '1', then clear the SPI interrupt flag (SPIxIF), perform a dummy read of the SPIxBUF register, and return from the Interrupt Service Routine (ISR).

If DMA is being used, there is no work around.

10. Module: SPI Module

The SMP bit (SPIxCON1<9>, where x = 1 or 2) does not have any effect when the SPI module is configured for a 1:1 prescale factor in Master mode. In this mode, whether the SMP bit is set or cleared, the data is always sampled at the end of data output time.

Work around

If sampling at the middle of data output time is required, then configure the SPI module to use a clock prescale factor other than 1:1 using the PPRE<1:0> and SPRE<2:0> bits in the SPIxCON1 register.

11. Module: ECAN Module

If multiple ECAN transmit buffers are enabled (multiple TXREQ or TXEN bits are set to '1' simultaneously), then the message transmissions from the enabled buffers may interfere with one another. As a result, incorrect ID and data transmissions will occur intermittently.

Work around

Enable only Buffer 0 for transmission at any given time. In the user application, this can be ensured by checking that all other TXREQn and TXENn bits are clear before setting the TXREQn or TXENn bit to Buffer 0.

12. Module: ECAN Module

Under specific conditions, the first five bits of a transmitted identifier may not match the value in the transmit buffer SID. If the ECAN module detects a Start-of-Frame (SOF) in the third bit of interframe space and if a message to be transmitted is pending, the first five bits of the transmitted identifier may be corrupted.

Work around

None.

13. Module: ECAN Module Loopback Mode

The ECAN module (ECAN1 or ECAN2) does not function correctly in Loopback mode.

Work around

Do not use Loopback mode.

14. Module: I^2C Module

The Bus Collision Status bit (BCL) does not get set when a bus collision occurs during a Restart or Stop event. However, the BCL bit gets set when a bus collision occurs during a Start event.

Work around

None.

15. Module: INT0, ADC and Sleep/Idle Mode

ADC event triggers from the INT0 pin will not wake-up the device from Sleep or Idle mode if the SMPI bits are non-zero. This means that if the ADC is configured to generate an interrupt after a certain number of INT0 triggered conversions, the ADC conversions will not be triggered and the device will remain in Sleep. The ADC will perform conversions and wake-up the device only if it is configured to generate an interrupt after each INT0 triggered conversion (SMPI<3:0> = 0000).

Work around

None. If ADC event trigger from the INT0 pin is required, initialize SMPI<3:0> to '0000' (interrupt on every conversion).

16. Module: Doze Mode and Traps

The address error trap, stack error trap, math error trap and DMA error trap will not wake-up a device from Doze mode.

Work around

None.

17. Module: JTAG Programming

JTAG programming does not work.

Work around

None.

18. Module: UART

With the parity option enabled, a parity error, indicated by the PERR bit (UxSTA<3>) being set, may occur if the Baud Rate Generator contains an odd value. This affects both even and odd parity options.

Work around

Load the Baud Rate Generator (BRG) register, UxBRG, with an even value, or disable the peripheral's parity option by loading either 0b00 or 0b11 into the Parity and Data Selection bits, PDSEL<1:0> (UxMODE<2:1>).

19. Module: UART

The Receive Buffer Overrun Error Status bit, OERR (UxSTA<1>), may get set before the UART FIFO has overflowed. After the fourth byte is received by the UART, the FIFO is full. The OERR bit should set after the fifth byte has been received in the UART Shift register. Instead, the OERR bit may set after the fourth received byte with the UART Shift register empty.

Work around

After four bytes have been received by the UART, the UART Receiver Interrupt Flag bit, U1RXIF (IFS0<11>) or U2RXIF (IFS1<14>), will be set, indicating the UART FIFO is full. The OERR bit may also be set. After reading the UART receive buffer, UxRXREG, four times to clear the FIFO, clear both the OERR and UxRXIF bits in software.

20. Module: UART

UART receptions may be corrupted if the Baud Rate Generator is set up for 4x mode (BRGH = 1).

Work around

Use the 16x baud rate option (BRGH = 0) and adjust the baud rate accordingly.

21. Module: UART

The UTXISEL0 bit (UxSTA<13>) is always read as zero regardless of the value written to it. The bit can be written to either a '0' or '1', but will always be read as zero. This will affect read-modify-write operations such as bitwise or shift operations. Using a read-modify-write instruction on the UxSTA register (e.g., BSET, BLCR) will always write the UTXISEL0 bit to zero.

Work around

If a UTXISEL0 value of '1' is needed, avoid using read-modify-write instructions on the UxSTA register. Copy the UxSTA register to a temporary variable and set UxSTA<13> prior to performing read-modify-write operations. Copy the new value back to the UxSTA register.

22. Module: UART

The auto-baud feature may not calculate the correct baud rate when the High Baud Rate Enable bit, BRGH, is set. With the BRGH bit set, the baud rate calculation used is the same as BRG = 0.

Work around

If the auto-baud feature is needed, use the Low Baud Rate mode by clearing the BRGH bit.

23. Module: UART

With the auto-baud feature selected, the Sync Break character (0x55) may be loaded into the FIFO as data.

Work around

To prevent the Sync Break character from being loaded into the FIFO, load the UxBRG register with either 0x0000 or 0xFFFF prior to enabling the auto-baud feature (ABAUD = 1).

24. Module: I^2C Module

Writing to I2CxTRN during a Start bit transmission generates a write collision, indicated by the IWCOL (I2CxSTAT<7>) bit being set. In this state, additional writes to the I2CxTRN register should be blocked. However, in this condition, the I2CxTRN register can be written, although transmissions will not occur until the IWCOL bit is cleared in software.

Work around

After each write to the I2CxTRN register, read the IWCOL bit to ensure a collision has not occurred. If the IWCOL bit is set, it must be cleared in software and I2CxTRN register must be rewritten.

25. Module: I²C Module

The ACKSTAT bit (I2CxSTAT<15>) only reflects the received ACK/NACK status for Master transmissions, but not for Slave transmissions. As a result, a Slave cannot use this bit to determine if it received an ACK or a NACK from a Master. In future silicon revisions, the ACKSTAT bit will reflect received ACK/NACK status for both Master and Slave transmissions.

Work around

After transmitting a byte, the Slave should poll the SDA line (subject to a time out period dependent on the application) to determine if an ACK (0) or a NACK (1) was received.

26. Module: I²C Module

The D_A Status bit (I2CxSTAT<5>) gets set on a slave data reception in the I2CxRCV register, but does not get set on a slave write to the I2CxTRN register. In future silicon revisions, the D_A bit will get set on a slave write to the I2CxTRN register.

Work around

Use the D_A status bit only for determining slave reception status and not slave transmission status.

27. Module: Traps and Idle Mode

If a clock failure occurs when the device is in Idle mode, the oscillator failure trap does not vector to the Trap Service Routine. Instead, the device will simply wake-up from Idle mode and continue code execution if the Fail-Safe Clock Monitor (FSCM) is enabled. Regardless, the Trap Service Routine must be included in the user application.

Work around

Whenever the device wakes up from Idle (assuming the FSCM is enabled) the user software should check the state of the OSCFAIL bit (INTCON1<1>) to determine whether a clock failure occurred, and then perform the appropriate clock switch operation.

28. Module: $\overline{\text{MCLR}}$ Wake-up from Sleep Mode

If a $\overline{\text{MCLR}}$ reset pulse causes the device to wake-up from Sleep mode, the device wakes up without waiting for the on-chip voltage regulator to power-up. This will subsequently result in a Brown-out Reset (BOR).

Work around

None.

29. Module: ECAN Module

The C1RXOVF2 and C2RXOVF2 registers are non-functional. They are always read back as 0x0000, even when a receive overflow has occurred.

Work around

None.

30. Module: FRC Oscillator

The device does not meet the internal FRC accuracy specifications in the data sheet (Table 25-18 of the "*dsPIC33F Family Data Sheet*" (DS70165)). The actual accuracy specifications are shown in Table 3.

Work around

None.

TABLE 3: INTERNAL FRC ACCURACY

AC Characteristics		Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial					
Parameter No.	**Characteristic**	**Min**	**Typical**	**Max**	**Units**	**Conditions**	
Internal FRC Accuracy @ FRC Frequency = 7.37 MHz(1,2)							
F20	—	-3	—	+3	%	-40°C ≤ TA ≤ +85°C	VDD = 3.0-3.6V

Note 1: Frequency calibrated at 25°C and 3.3V. TUN bits can be used to compensate for temperature drift.

2: Devices set to initial frequency of 7.37 MHz (±2%) at 25°C.

31. Module: QEI Interrupt Generation

The Quadrature Encoder Interface (QEI) module does not generate an interrupt when MAXCNT is set to 0xFFFF and the following events occur:

1. POSCNT underflows from 0x0000 to 0xFFFF.
2. POSCNT stops.
3. POSCNT overflows from 0xFFFF to 0x0000.

This sequence of events occurs when the motor is running in one direction, which causes POSCNT to underflow to 0xFFFF. Once this happens, the motor stops and starts to run in the opposite direction, which generates an overflow from 0xFFFF to 0x0000. The QEI module does not generate an interrupt when this condition occurs.

Work around

To prevent this condition from occurring, set MAXCNT to 0x7FFF, which will cause an interrupt to be generated by the QEI module.

In addition, a global variable could be used to keep track of bit 15, so that when an overflow or underflow condition is present on POSCNT, the variable will toggle bit 15. Example 1 shows the code required for this global variable.

EXAMPLE 1:

```
unsigned int POSCNT_b15 = 0;
unsigned int Motor_Position = 0;

int main(void)
{
            // ... User's code

            MAXCNT = 0x7FFF;       // Instead of 0xFFFF

            Motor_Position = POSCNT_b15 + POSCNT;

            // ... User's code
}
void __attribute__((__interrupt__)) _QEIInterrupt(void)
{
            IFSxbits.QEIIF = 0;    // Clear QEI interrupt flag
                                   // x=2 for dsPIC30F
                                   // x=3 for dsPIC33F
            POSCNT_b15 ^= 0x8000;  // Overflow or Underflow
}
```

32. Module: Device ID Register

On a few devices, the content of the Device ID register can change from the factory programmed default value immediately after RTSP or ICSP™ Flash programming.

As a result, development tools will not recognize these devices and will generate an error message indicating that the device ID and the device part number do not match. Additionally, some peripherals will be reconfigured and will not function as described in the device data sheet.

Refer to **Section 5. "Flash Programming"** (DS70191), of the "*dsPIC33F Family Reference Manual*" for an explanation of RTSP and ICSP Flash programming.

Work around

All RTSP and ICSP Flash programming routines must be modified as follows:

1. No word programming is allowed. Any word programming must be replaced with row programming.
2. During row programming, load write latches as described in **5.4.2.3 "Loading Write Latches"** of **Section 5. "Flash Programming"** (DS70191).
3. After latches are loaded, reload any latch location (in a given row) that has 5 LSB set to 0x18, with the original data. For example, reload one of the following latch locations with the desired data:
   0xXXXX18, 0xXXXX38, 0xXXXX58, 0xXXXX78, 0xXXXX98, 0xXXXXB8, 0xXXXXD8, 0xXXXXF8
4. Start row programming by setting NVMOP<3:0> = '0001' (memory row program operation) in the NVMCON register.
5. After row programming is complete, verify the contents of Flash memory.
6. If Flash verification errors are found, repeat steps 2 through 5. If Flash verification errors are found after a second iteration, report this problem to Microchip.

Steps 1 through 5 in the work around are implemented in MPLAB IDE version 8.00 for the MPLAB ICD 2, MPLAB REAL ICE™ in-circuit emulator and PM3 tools.

33. Module: SPI Module

SPI1 functionality for pin 34 (U1RX/SDI1/RF2) is enabled by the SPI2 module. As a result, two side effects occur:

1. RF2 functionality is disabled if the SPI2 module is enabled.
2. This pin will not function as SDI1 if the SPI1 module is enabled.

This issue affects 64-pin devices only:

- dsPIC33FJ64MC506
- dsPIC33FJ64MC706
- dsPIC33FJ128MC506
- dsPIC33FJ128MC706

Work around

Two conditions apply:

1. If the SPI2 module is used, pin 34 cannot be used as an I/O (RF2). It is recommended to use another I/O pin.
2. If the SPI1 module is used, the SPI2 module must also be enabled to gain SDI1 functionality on pin 34. As an alternative, I/O (RF2) can be configured as an input, which will allow pin 34 to function as SDI1.

34. Module: UART

The auto-baud feature may miscalculate for certain baud rate and clock speed combinations, resulting in a BRG value that is greater than or less than the expected value by 1. This may result in reception or transmission failures.

Work around

Test the auto-baud rate at various clock speed and baud rate combinations that would be used in an application. If an inaccurate BRG value is generated, manually correct the baud rate in user software.

35. Module: Motor Control PWM

If a fault is cleared by user software during Latched mode while the current PWM cycle remains active, any additional faults that arrive before the end of the current PWM cycle will not be detected. The PWM outputs will return to the non-fault state at the end of that cycle and remain there, even if the fault is currently active. No interrupt will be sent to the interrupt controller.

Work around

While in the Latched mode of operation, do not attempt to clear faults during the same cycle in which they arrive.

36. Module: Motor Control PWM

An active fault while the Motor Control PWM module is stopped in Idle mode, does not result in an interrupt request being sent to the processor. Therefore, the processor does not wake-up from Idle mode.

Work around

None.

37. Module: DMA

When a DMA channel is enabled in Single-Shot mode while the device is in Idle mode, and the corresponding peripheral is active and configured to operate during Idle mode, the DMA channel may not become disabled immediately upon transferring the required amount of data.

As a result, the number of bytes or words of data transferred may exceed the DMA transfer count specified in the DMAxCNT register.

For example, if DMA transfers are active for both SPI byte transmissions and receptions, and only the receive DMA channel interrupt is enabled for waking up the device from Idle mode, an extra byte will be transmitted by the time the device wakes up from Idle mode.

Work around

None.

38. Module: Doze Mode and Traps

A DMA error trap may not be generated when the device is in Doze mode.

Work around

None.

39. Module: ECAN Module

If receive masking is disabled for any receive filter 'n' (n = 0, 1, 2, …, 14) by setting the corresponding Mask Source Select (FnMSK<1:0>) bits to '11', the ECAN module behaves as if it were configured for DeviceNet Addressing. In this case, bit 7 of Byte 0 of every incoming message is compared with EID<17> of the receive filter, and desired messages may not be received as a result.

Work around

To avoid this issue, do not disable receive masking for a filter 'n' by setting the FnMSK<1:0> bits. Instead, select any receive mask by configuring FnMSK<1:0> = 10, 01 or 00, and then disable the individual mask bits by clearing the SID<10:0> and EID<17:0> bits in the CiRXMnSID and CiRXMnEID registers (I = 1 or 2, and n = 0, 1 or 2).

40. Module: Output Compare Module

When the Output Compare Module is operated in the Dual Compare Match mode, a timer compare match with the value in the OCxR register sets the OCx output producing a rising edge on the OCx pin. Then, when a timer compare match with the value in the OCxRS register occurs, the OCx output is reset producing a falling edge on the OCx pin.

The above statement applies to all conditions except when the difference between OCxR and OCxRS is 1. In this case, the Output Compare module may miss the reset compare event, and cause the OCx pin to remain continuously high. This condition will remain until the difference between values in the OCxR and OCxRS registers is made greater than 1.

Work around

Ensure in software that the difference between values in OCxR and OCxRS registers is maintained greater than 1.

41. Module: UART

When the UART is in 4x mode (BRGH = 1) and using two Stop bits (STSEL = 1), it may sample the first Stop bit instead of the second one.

This issue does not affect the other UART configurations.

Work around

Use the 16x baud rate option (BRGH = 0) and adjust the baud rate accordingly.

42. Module: UART

When an auto-baud is detected, the receive interrupt may occur twice. The first interrupt occurs at the beginning of the Start bit and the second after reception of the Sync field character.

Work around

If an extra interrupt is detected, ignore the additional interrupt.

43. Module: Motor Control PWM – PWM Generator 2

The PWM outputs, PWM2H and PWM2L, do not generate PWM signals if the following two conditions are met:

1. FLTBCON register is written with any value.
2. Fault pin B ($\overline{\text{FLTB}}$) is low while the FLTBCON register is being written.

Work around

There are three possible work arounds for this issue:

1. Do not write to the FLTBCON register.
2. If Fault B is needed for the application, always keep the $\overline{\text{FLTB}}$ pin connected to logic high while writing to the FLTBCON register.
3. Set the $\overline{\text{FLTA}}$ pin to a logic low while writing to the FLTBCON register.

44. Module: Motor Control PWM – PWM Counter Register

If the PTDIR bit is set (when PTMR is counting down), and the CPU execution is halted (after a breakpoint is reached), PTMR will start counting up, as if PTDIR was zero.

Work around

None.

45. Module: DMA

When the DMA channel is configured for NULL Data Peripheral Write mode (DMAxCON<11> = 1), it does not execute a null (all zeros) write to the peripheral address.

Work around

Use two DMA channels to receive data from the peripheral module. One channel must be configured to transfer data from the peripheral to DMA RAM, while another channel must be configured to transfer dummy data from the DMA RAM to the peripheral. Both channels must be setup for the same DMA request.

46. Module: DMA

A low priority DMA channel request can be pre-empted by a higher priority DMA channel request. For example, if DMA Channel 0 has a higher priority than DMA Channel 1, a request to DMA channel 1 will be pending while DMA Channel 0 is processing its request. If DMA Channel 1 receives another request while it is in a pending request state, the DMA module does not generate a DMA error trap event.

Work around

None. Using higher priority DMA channels for servicing sources of frequent requests significantly reduces the possibility of the condition described above ocurring, but does not completely eliminate it.

47. Module: DMA

When the DMA channel is configured for One Shot mode with NULL write enabled, the channel will write an extra NULL to the peripheral register after completing the last transfer. In the case of the SPI module and the SPIxBUF register, this would cause the SPI module to perform an extra receive operation.

Work around

None. In the case of using DMA NULL write with the SPI module, perform a dummy read of the SPIxBUF register after the DMA transfer is completed to clear the SPIRBF flag and prevent an unexpected overflow condition on the next SPI receive operation.

48. Module: REPEAT Instruction

Any instruction executed inside a REPEAT loop, which produces a Read-After-Write stall condition, results in the instruction being executed fewer times than was intended.

An example of such code is:

```
repeat #0xf
inc [w1],[++w1]
```

Work around

Avoid repeating an instruction that creates a stall using a REPEAT instruction. Instead, use the DO instruction while using the dsPIC33F device. A code example is shown below:

```
    DO #0x15, end
    inc [w1],[++w1]
end: nop
```

49. Module: FRC Oscillator

For certain values of the TUN<5:0> bits (OSCTUN<5:0>), the resultant frequencies do not match the expected values.

As shown in Table 4, the actual frequencies obtained for different values of the TUN<5:0> bits are listed in terms of percentage change relative to the center frequency of 7.3728 MHz. The frequency errors listed in the table are approximate and may vary slightly from device to device.

It is recommended that the user application include some means of measuring the exact oscillator frequency in order to verify the frequencies listed below.

Work around

Configure your peripherals and other system parameters based on the actual frequencies listed in Table 4.

TABLE 4:

TUN<5:0>	Expected Change from 7.3728 MHz	Actual Change from 7.3728 MHz
000000	—	—
000001	+0.375%	+0.375%
000010	+0.75%	+0.75%
000011	+1.125%	+1.125%
000100	+1.5%	+1.5%
000101	+1.875%	+1.875%
000110	+2.25%	+2.25%
000111	+2.625%	+2.625%
001000	+3%	+3%
001001	+3.375%	+3.375%
001010	+3.75%	+3.75%
001011	+4.125%	+4.125%
001100	+4.5%	+4.5%
001101	+4.875%	+4.875%
001110	+5.25%	+5.25%
001111	+5.625%	+5.625%
010000	+6%	+8.325%
010001	+6.375%	+8.7%
010010	+6.75%	+9.075%
010011	+7.125%	+9.45%
010100	+7.5%	+9.825%
010101	+7.875%	+10.2%
010110	+8.25%	+10.575%
010111	+8.625%	+10.95%
011000	+9%	+11.325%
011001	+9.375%	+11.7%
011010	+9.75%	+12.075%
011011	+10.125%	+12.45%
011100	+10.5%	+12.825%
011101	+10.875%	+13.2%
011110	+11.25%	+13.575%
011111	+11.625%	+13.95%

TABLE 4: (CONTINUED)

TUN<5:0>	Expected Change from 7.3728 MHz	Actual Change from 7.3728 MHz
100000	-12%	-12%
100001	-11.625%	-11.625%
100010	-11.25%	-11.25%
100011	-10.875%	-10.875%
100100	-10.5%	-10.5%
100101	-10.125%	-10.125%
100110	-9.75%	-9.75%
100111	-9.375%	-9.375%
101000	-9%	-9%
101001	-8.625%	-8.625%
101010	-8.25%	-8.25%
101011	-7.875%	-7.875%
101100	-7.5%	-7.5%
101101	-7.125%	-7.125%
101110	-6.75%	-6.75%
101111	-6.375%	-6.375%
110000	-6%	-3.675%
110001	-5.625%	-3.3%
110010	-5.25%	-2.925%
110011	-4.875%	-2.55%
110100	-4.5%	-2.175%
110101	-4.125%	-1.8%
110110	-3.75%	-1.425%
110111	-3.375%	-1.05%
111000	-3%	-0.675%
111001	-2.625%	-0.3%
111010	-2.25%	+0.075%
111011	-1.875%	+0.45%
111100	-1.5%	+0.825%
111101	-1.125%	+1.2%
111110	-0.75%	+1.575%
111111	-0.375%	+1.95%

50. Module: UART

When the UART is configured for IR interface operations (UxMODE<9:8> = 11), the 16x baud clock signal on the BCLK pin is present only when the module is transmitting. The pin is idle at all other times.

Work around

Configure one of the output compare modules to generate the required baud clock signal when the UART is receiving data or in an idle state.

51. Module: SPI Module

Setting the DISSCK bit in the SPIxCON1 register does not allow the user application to use the SCK pin as a general purpose I/O pin.

Work around

None.

52. Module: I^2C Module

The BCL bit in I2CSTAT can be cleared only with 16-bit operation and can be corrupted with 1-bit or 8-bit operations on I2CSTAT.

Work around

Use 16-bit operations to clear BCL.

53. Module: I^2C Module

If there are two I^2C devices on the bus, one of them is acting as the Master receiver and the other as the Slave transmitter. Suppose that both devices are configured for 10-bit addressing mode, and have the same value in the A10 and A9 bits of their addresses. When the Slave select address is sent from the Master, both the Master and Slave acknowledges it. When the Master sends out the read operation, both the Master and the Slave enter into Read mode and both of them transmit the data. The resultant data will be the ANDing of the two transmissions.

Work around

Use different addresses including the higher two bits (A10 and A9) for different modules.

APPENDIX A: REVISION HISTORY

Revision A (3/2007)

Initial release of this document.

Revision B (6/2007)

Added the following silicon issues: 33 (Device ID Register), 34 (Device ID Register), 35 (SPI Module), 36 (UART), 37 (Motor Control PWM), 38 (Motor Control PWM), 39 (DMA) and 40 (Doze Mode and Traps).

Revision C (10/2007)

Removed silicon issue 7 (Output Compare). Updated silicon issue 28 (Traps and Idle Mode). Added silicon issues 40 (ECAN Module), 41 (Output Compare Module), 42-43 (UART), 44 (Motor Control PWM – PWM Generator 2), 45 (Motor Control PWM – PWM Counter Register), 46-48 (DMA), 49 (`REPEAT` Instruction), and 50 (FRC Oscillator).

Revision D (4/2008)

Updated silicon issues 4 (DMA Module: Interaction with EXCH Instruction) and 11 (ECAN Module). Removed silicon issue 24 (ECAN). Added silicon issues 50 (UART), 51 (SPI Module) and 52-53 (I2C Module).

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.

Trademarks

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
ISO/TS 16949:2002

Microchip received ISO/TS-16949:2002 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.

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://support.microchip.com
Web Address:
www.microchip.com

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

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
Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

Kokomo
Kokomo, IN
Tel: 765-864-8360
Fax: 765-864-8387

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

Santa Clara
Santa Clara, CA
Tel: 408-961-6444
Fax: 408-961-6445

Toronto
Mississauga, Ontario,
Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

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

China - Beijing
Tel: 86-10-8528-2100
Fax: 86-10-8528-2104

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

China - Hong Kong SAR
Tel: 852-2401-1200
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: 86-21-5407-5533
Fax: 86-21-5407-5066

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

China - Shenzhen
Tel: 86-755-8203-2660
Fax: 86-755-8203-1760

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

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

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

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

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-4182-8400
Fax: 91-80-4182-8422

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

India - Pune
Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

Japan - Yokohama
Tel: 81-45-471- 6166
Fax: 81-45-471-6122

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: 886-3-572-9526
Fax: 886-3-572-6459

Taiwan - Kaohsiung
Tel: 886-7-536-4818
Fax: 886-7-536-4803

Taiwan - Taipei
Tel: 886-2-2500-6610
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

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

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

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

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

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

UK - Wokingham
Tel: 44-118-921-5869
Fax: 44-118-921-5820

01/02/08