dsPIC33CH128MP508 Family Silicon Errata and Data Sheet Clarification

The dsPIC33CH128MP508 family devices that you have received conform functionally to the current Device Data Sheet (DS70005319**D**), 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 dsPIC33CH128MP508 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 (**A2**).

Data Sheet clarifications and corrections start on page 14, following the discussion of silicon issues.

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

For example, to identify the silicon revision level using MPLAB IDE in conjunction with a hardware debugger:

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

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

The DEVREV values for the various dsPIC33CH128MP508 silicon revisions are shown in Table 1.

TABLE 1: SILICON DEVREV VALUES

Part Number	Device ID[1]	Revision ID for Silicon Revision
		A2
Main (With CAN FD)		
dsPIC33CH64MP502	0x8740	0x0002
dsPIC33CH128MP502	0x8750	
dsPIC33CH64MP503	0x8741	
dsPIC33CH128MP503	0x8751	
dsPIC33CH64MP505	0x8742	
dsPIC33CH128MP505	0x8752	
dsPIC33CH64MP506	0x8743	
dsPIC33CH128MP506	0x8753	
dsPIC33CH64MP508	0x8744	
dsPIC33CH128MP508	0x8754	

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

TABLE 1: SILICON DEVREV VALUES (CONTINUED)

Part Number	Device ID[1]	Revision ID for Silicon Revision
		A2
Secondary (With CAN FD)		
dsPIC33CH64MP502S1	0x87C0	0x0002
dsPIC33CH128MP502S1	0x87D0	
dsPIC33CH64MP503S1	0x87C1	
dsPIC33CH128MP503S1	0x87D1	
dsPIC33CH64MP505S1	0x87C2	
dsPIC33CH128MP505S1	0x87D2	
dsPIC33CH64MP506S1	0x87C3	
dsPIC33CH128MP506S1	0x87D3	
dsPIC33CH64MP508S1	0x87C4	
dsPIC33CH128MP508S1	0x87D4	
Main (Without CAN FD)		
dsPIC33CH64MP202	0x8700	0x0002
dsPIC33CH128MP202	0x8710	
dsPIC33CH64MP203	0x8701	
dsPIC33CH128MP203	0x8711	
dsPIC33CH64MP205	0x8702	
dsPIC33CH128MP205	0x8712	
dsPIC33CH64MP206	0x8703	
dsPIC33CH128MP206	0x8713	
dsPIC33CH64MP208	0x8704	
dsPIC33CH128MP208	0x8714	
Secondary (Without CAN FD)		
dsPIC33CH64MP202S1	0x8780	0x0002
dsPIC33CH128MP202S1	0x8790	
dsPIC33CH64MP203S1	0x8781	
dsPIC33CH128MP203S1	0x8791	
dsPIC33CH64MP205S1	0x8782	
dsPIC33CH128MP205S1	0x8792	
dsPIC33CH64MP206S1	0x8783	
dsPIC33CH128MP206S1	0x8793	
dsPIC33CH64MP208S1	0x8784	
dsPIC33CH128MP208S1	0x8794	

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

TABLE 2: SILICON ISSUE SUMMARY

Module	Feature	Item Number	Issue Summary	Affected Revisions
				A2
ECC	Status	1.	ECCSTATH/L registers cannot be read.	X
ECC	Status	2.	SECSYNDx bits in the ECCSTATH register cannot be read.	X
I^{2}C	Interrupt	3.	In Secondary mode, an incorrect interrupt is generated with DHEN = 1.	X
I^{2}C	Error	4.	Bus collision error cannot be cleared.	X
I^{2}C	Error	5.	False bus collision error generated.	X
I^{2}C	Idle	6.	Address cannot be received in Idle mode.	X
Oscillator	PLL	7.	FRCDIVN drives the PLL instead of the FRC.	X
Oscillator	HS, XT	8.	Removed	
PWM	Dead Time	9.	When feed-forward PCI is used for dead-time compensation (DTCMPSEL = 1), the PWMx outputs are overridden.	X
UART	FERR	10.	The OERR bit cannot be cleared by software.	X
UART	OERR	11.	The FERR bit will not get set if one Stop bit is received.	X
UART	TRMT	12.	The 9th byte received will not be available to be read.	X
UART	TRMT	13.	The TRMT bit takes time to set on the last transmit completion.	X
UART	TRMT	14.	The TRMT bit is unreliable when there is back-to-back Break character transmission.	X
UART	Idle	15.	SLPEN = 1 will not keep the UART BRG clock active in Sleep mode.	X
UART	RIDLE	16.	The RIDLE bit takes one instruction cycle to get cleared after ABAUD is set.	X
UART	TXWRE	17.	The TXWRE bit (UxSTAH[7]) cannot be cleared once it gets set.	X
UART	Address Detect	18.	When writing to UxP1 with UTXBRK = 1, the content of P1 will not get transmitted.	X
UART	DMX	19.	In DMX mode with reception enabled, no interrupt is generated on receiving start code or break of data frame.	X
UART	UART	20.	When UART is used in DMX mode (MOD[3:0] (UxMODE[3:0]) = 0b1010) with reception enabled, no interrupt is generated on receiving start code or break of data frame.	X
UART	Smart Card	21.	The Waiting Time Counter Interrupt Flag (WTCIF) is set when the last x character transmitted has the bit, LAST = 0.	X
UART	XOFF	22.	XOFF is transmitted when one empty space remains in the RX buffer.	X
CPU	FLIM Instruction	23.	When the operands are of different signs, the FLIM instruction may not force the correct data limit.	X
SCCP/MCCP	Clock Source	24.	Using FOSC as the clock source may cause synchronization issues.	X
I^{2}C	SMBus 3.0	25.	When Configuration bit, SMBEN (FDEVOPT[10]) = 1, the SMBus 3.0 VIH minimum specification may not be met.	X
I/O	POR	26.	Spike on I/O at POR.	X
CPU	MAXAB/MINAB Instructions	27.	When the operands are of different signs, the MAXAB, MINAB and MINZAB instructions may not output the correct value.	X

TABLE 2: SILICON ISSUE SUMMARY (CONTINUED)

Module	Feature	Item Number	Issue Summary	Affected Revisions
				A2
CPU	`div.sd` Instruction	28.	When using the signed 32-by-16-bit division instruction, `div.sd`, the Overflow bit is not getting set when an overflow occurs.	X
DMA	ADC Triggers	29.	DMA is triggered continuously from ADC.	X
PWM	Time Base Capture	30.	PWM Capture Status (CAP) flag will not set again under certain conditions.	X
I^2C	I^2C	31.	All instances of I^2C may exhibit errors and should not be used.	X
Oscillator	VCO and AVCO Dividers	32.	Main and auxiliary PLL external VCO dividers can fail to output the clock signal.	X
Main Secondary Interface (MSI)	DMA Transfer	33.	DMA transfer of mailbox data.	X
Secondary CPU	`REPEAT`	34.	`REPEAT` loops interrupted by nested interrupts on the Secondary Core may corrupt data and trap.	X

Silicon Errata Issues

Note: This document summarizes all silicon errata issues from all revisions of silicon, previous as well as current. Only the issues indicated by the shaded column in the following tables apply to the current silicon revision (**A2**).

1. Module: ECC

The ECCSTATH/L registers cannot be read when an ECC error happens. The ECC Double-Bit Error (ECCDBE) trap and ECC Single Bit Error (ECCSBE) interrupt will work correctly, but the ECCSTATH/ECCSTATL registers will always read as zero.

Work around

None.

Affected Silicon Revisions

Core	A2		
Main	X		
Secondary			

2. Module: ECC

In the ECCSTATH register, the SECSYNDx bits cannot be read when an ECC error happens.

Work around

None.

Affected Silicon Revisions

Core	A2		
Main			
Secondary	X		

3. Module: I^2C

In Secondary mode with DHEN = 1 (Data Hold Enable), if software sends a NACK, a Secondary interrupt is asserted at the 9th falling edge of the clock.

Work around

Software should ignore the Secondary interrupt that is asserted after sending a NACK.

Affected Silicon Revisions

Core	A2		
Main	X		
Secondary	X		

4. Module: I^2C

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

Work around

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

Affected Silicon Revisions

Core	A2		
Main	X		
Secondary	X		

5. Module: I^2C

In Secondary mode, false bus collision triggers are generated when the bus collision is enabled (SBCDE = 1) and a Stop bit is received.

Work around

Ignore the bus collision. Disable the I^2C module and then re-enable the module.

Affected Silicon Revisions

Core	A2		
Main	X		
Secondary	X		

6. Module: I^2C

In Secondary mode, an address cannot be received when the device is in Idle and the module is set for discontinue in Idle (I2CSIDL = 1).

Work around

None.

Affected Silicon Revisions

Core	A2		
Main	X		
Secondary	X		

7. Module: Oscillator

When using the 8 MHz internal FRC Oscillator with Primary PLL as either a system clock or a peripheral source, FRCDIVN drives the PLL instead of the FRC.

This means that the PLL FRC input selection is subject to the FRCDIV[2:0] bits and could lead to a condition where the minimum PLL input requirement of 8 MHz is not maintained.

Work around

Ensure FRCDIV[2:0] bits are maintained as zero when using FRCPLL as either a system clock or a peripheral source.

Affected Silicon Revisions

Core	A2		
Main	X		
Secondary	X		

8. Module: Oscillator

This errata is no longer applicable to any silicon revisions of this product. See **Section 2.5 External Oscillator Pins** in the current device data sheet (DS70005319D) for guidance on oscillator design to avoid start up related issues.

9. Module: PWM

When feed-forward PCI is used for dead-time compensation (DTCMPSEL = `1`), the PWMx outputs are overridden.

Work around

Use Sync PCI (DTCMPSEL = `0`) for dead-time compensation.

Affected Silicon Revisions

Core	A2		
Main	X		
Secondary	X		

10. Module: UART

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

Work around

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

Affected Silicon Revisions

Core	A2		
Main	X		
Secondary	X		

11. Module: UART

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

Work around

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

Affected Silicon Revisions

Core	A2		
Main	X		
Secondary	X		

12. Module: UART

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

Work around

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

Affected Silicon Revisions

Core	A2		
Main	X		
Secondary	X		

13. Module: UART

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

Work around

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

Affected Silicon Revisions

Core	A2		
Main	X		
Secondary	X		

14. Module: UART

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

Work around

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

Affected Silicon Revisions

Core	A2		
Main	X		
Secondary	X		

15. Module: UART

UART will not work correctly in Sleep mode. SLPEN = `1` will not keep the UART baud rate clock active in Sleep mode.

Work around

None.

Affected Silicon Revisions

Core	A2		
Main	X		
Secondary	X		

16. Module: UART

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

Work around

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

Affected Silicon Revisions

Core	A2		
Main	X		
Secondary	X		

17. Module: UART

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

Work around

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

Affected Silicon Revisions

Core	A2		
Main	X		
Secondary	X		

18. Module: UART

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

Work around

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

Affected Silicon Revisions

Core	A2		
Main	X		
Secondary	X		

19. Module: UART

When UART is in DMX mode, MOD[3:0] (UxMODE[3:0]) = `0b1010` with transmission enabled, any write to the UxP1 register after enabling UART would be ignored.

Work around

Write the value of the UxP1 register before enabling the UART.

Affected Silicon Revisions

Core	A2		
Main	X		
Secondary	X		

20. Module: UART

When UART is used in DMX mode, MOD[3:0] (UxMODE[3:0]) = `0b1010` with reception enabled, no interrupt is generated on receiving a start code or break of data frame. Depending on the URXISEL[2:0] (UxSTAH[10:8]) bits, an interrupt will be generated on receiving the data bytes.

Work around

None.

Affected Silicon Revisions

Core	A2		
Main	X		
Secondary	X		

21. Module: UART

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

Work around

Ignore WTC interrupt events on non-last bytes.

Affected Silicon Revisions

Core	A2		
Main	X		
Secondary	X		

22. Module: UART

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

Work around 1

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

Work around 2

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

Affected Silicon Revisions.

Core	A2		
Main	X		
Secondary	X		

23. Module: CPU

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

Work around

None.

Affected Silicon Revisions

Core	A2		
Main	X		
Secondary	X		

24. Module: SCCP/MCCP

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

Work around

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

Affected Silicon Revisions

Core	A2		
Main	X		
Secondary	X		

25. Module: I^{2}C

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

Work around

None.

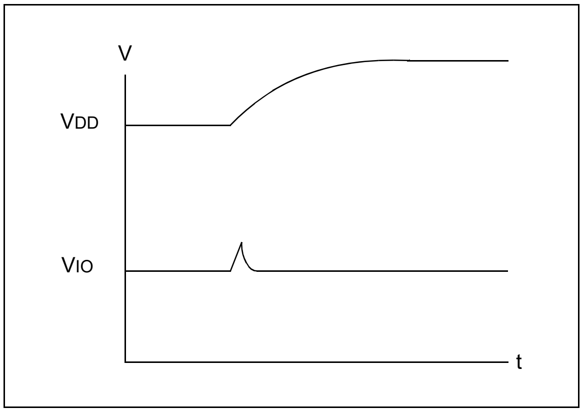
Affected Silicon Revisions

Core	A2		
Main	X		
Secondary	X		

26. Module: I/O

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

FIGURE 1: I/O RAMP



Work around

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

Example applications affected may include complementary power switches, where a transient current shoot-through might occur.

High-voltage applications with complementary switches should power the high-voltage 200 µSec later than powering the dsPIC® device to avoid the current shoot-through. This behavior is specific to each device and not affected by aging.

Affected Silicon Revisions

Core	A2		
Main	X		
Secondary	X		

27. Module: CPU

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

Work around

None.

Affected Silicon Revisions

Core	A2		
Main	X		
Secondary	X		

28. Module: CPU

When using the Signed 32-by-16-Bit Division instruction, div.sd, the Overflow bit may not always get set when an overflow occurs. This erratum only affects operations in which at least one of the following conditions is true:

- Dividend and divisor differ in sign,
- Dividend > 0x3FFFFFFF or
- Dividend < 0xC0000000

Work around

The application software must perform both of the following actions 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

Core	A2		
Main	X		
Secondary	X		

29. Module: DMA

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

Work around

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

Affected Silicon Revisions

Core	A2		
Main	X		
Secondary	X		

30. Module: PWM

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

Work around

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

Affected Silicon Revisions

Core	A2		
Main	X		
Secondary	X		

31. Module: I^{2}C

All instances of I^{2}C/SMBus may exhibit errors and should not be used. When operating I^{2}C/SMBus in a noisy environment, the I^{2}C module may exhibit various errors. These errors may include, but are not limited to, corrupted data, unintended interrupts or the I^{2}C bus getting hung up due to injected noise. Examples of system noise include, but are not limited to, PWM outputs or other pins toggled at high speed adjacent to the I^{2}C pins. Both Main and Secondary I^{2}C/SMBus modes may exhibit this issue.

Work around

If I^{2}C is required, use a software I^{2}C implementation. An example I^{2}C software library is available from Microchip:

www.microchip.com/dsPIC33C_I2C_SoftwareLibrary

Affected Silicon Revisions

Core	A2		
Main	X		
Secondary	X		

32. Module: Oscillator

At PLL start-up, the main and auxiliary PLL VCO dividers may occasionally halt and not provide a clock output. The VCO and AVCO dividers can be selected as clock sources for different peripheral modules, including the ADC, PWM, DAC, CAN FD, UART, etc.

All VCO and AVCO divider outputs, FVCO/2, FVCO/3, FVCO/4, FVCODIV, AFVCO/2, AFVCO/3, AFVCO/4 and AFVCODIV, are affected and may show the issue independently.

Any type of Reset may recover the VCO/AVCO divider clock outputs (Software Reset, WDT, $\overline{\text{MCLR}}$ or POR).

Work around 1

Use another clock source, such as the FOSC, PLL or APLL output (FPLLO and AFPLLO), instead of the VCO or AVCO dividers.

Work around 2

If the application requires the VCO/AVCO divider, peripheral activity should be verified within some time or the device should be Reset.

The Watchdog Timer (WDT) or Timer1 may be used to establish the time-out period and reset the device. The following steps may be taken to implement this work around for any given peripheral and VCO/AVCO divider combination.

1) Set up the WDT or Timer1 time-out period.
2) Set up the VCO/AVCO divider source to be used by the peripheral.
3) Start the peripheral from this source.
4) Verify peripheral activity using an interrupt or other method and disable the time-out.
5) If the time-out expires, the device should be reset; WDT will reset the device without intervention, but Timer1 will require a SWR in the Timer1 Interrupt Service Routine.

Affected Silicon Revisions

Core	A2		
Main	X		
Secondary	X		

33. Module: Main Secondary Interface (MSI)

When transferring data between cores using the MSI mailbox with DMA, if the transmitting core is running more than two times the system clock frequency of the receiving core, the data transfer may not be processed correctly and the MSI may appear to be in a Freeze state.

An example of the application includes the DMA in the transmitting core may load data to the MSI Mailbox register after the receiving core initiates the MSI interrupt to Acknowledge data reception, but prior to hardware clear of the DTRDY bit, causing the hardware to appear to be frozen in the state where DTRDY is set in the transmitting core and cleared in the receiving core.

Work around

Do not use DMA for MSI data transfer when the core sending data will be operating at more than two times the system clock frequency of the core receiving data. Instead, in the MSI ISR, clear the DTRDY bit and load the next data to the MSI buffer/FIFO directly.

Affected Silicon Revisions

Core	A2		
Main	X		
Secondary	X		

34. Module: Secondary CPU

While the Secondary CPU core is executing the instruction targeted by a `REPEAT` loop and two or more nestable interrupts occur in near simultaneous proximity, data corruption may occur within the lowest priority Interrupt Service Routine (ISR) and/or original `REPEAT` loop code. Specifically, when the CPU is vectoring to the lower priority ISR and a higher priority stimulus forces the CPU to vector to a different ISR, hardware will retarget the background `REPEAT` loop onto the first instruction of the lower priority ISR. Typically, ISRs start with a stack `PUSH` instruction, resulting in repeated stack pushes later when the lower priority ISR executes. This manifests as an Address Error Trap upon return from the low-priority ISR as the CPU attempts to use the repeated stack push data as the ISR's return address.

If the first instruction of the lower priority ISR is repeatable without harmful effects, such as a `NOP`, data corruption will still be apparent in the background code as its `REPEAT` loop will prematurely terminate.

Work around 1

Avoid using `REPEAT` instructions in projects built for the Secondary Core. For compiled code, use MPLAB® XC16, v1.70 or higher and specify `-merrata=repeat_gie` or `-merrata=repeat_nstdis` as an additional option for XC16 (Global Options). Alternatively, this option may be individually added to the command-lines invoking `xc16-gcc` and `xc16-ld`.

`-merrata=repeat_gie` will suppress generation of `REPEAT` loops unless required for a hardware divide instruction. For divides, the `REPEAT` loop will be prefixed with instructions to save INTCON2. Write GIE (INTCON2[15]) = `0` to globally disable all interrupts, then postfix with an instruction to restore INTCON2. Toolchain library calls, such as `memcpy()`/`printf()`, etc., will also link against implementations that avoid `REPEAT` loops and globally mask interrupts where needed for divide instructions.

`-merrata=repeat_nstdis` will also suppress `REPEAT` loops. However, for hardware divide loops, saving/restoring will be applied to INTCON1 and NSTDIS (INTCON1[15]) = `1` will be written to disable interrupt nesting while maintaining GIE unchanged. Toolchain libraries will continue to use GIE (INTCON2[15]) global interrupt masking instead of relying on NSTDIS (INTCON1[15]) to protect divide loops.

Note: Blocking interrupt nesting typically adds no latency to interrupt processing, but increases worst-case interrupt latency for higher priority ISRs. A low-priority interrupt triggered immediately before a high-priority stimulus adds the entire execution time of the low-priority ISR to the worst-case response latency for the higher priority ISR.

Work around 2

Ensure that all REPEAT instructions only execute in a context where back-to-back interrupts of nestable priority are impossible, such as within IPL6 and IPL7 ISRs, or anywhere all enabled interrupts are known to be configured to the same priority level. Also, if the application implements periodic interrupts corresponding to internal/synchronously timed events, it may be possible to find or wait for an execution window where a REPEAT loop can deterministically complete without two nested interrupts able to clobber it. A PWRSAV call to enter Idle mode may help find synchronized, safe windows as code flow will halt, then resume in response to the next interrupt.

As compiled code can have REPEAT loops hidden within them, this work around should only be attempted on a per source file basis with Work around 1 applied for all other files that cannot be carefully controlled or which do not require REPEAT loops. It is additionally suggested that the full project disassembly listing be searched for REPEAT instructions and that proper interrupt masking, nest disabling or contextual state and timing conditions for safe REPEAT execution have been met.

Affected Silicon Revisions

Core	A2		
Main			
Secondary	X		

Data Sheet Clarifications

The following typographic corrections and clarifications are to be noted for the latest version of the device data sheet (DS70005319**D**):

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

1. Module: Electrical Characteristics

In Table 24-29, the FRC percentage is changed from -3 to +3, to -2 to +2. All changes are shown below in **bold**.

2. Module: Functional Safety and Qualification Support

Functional Safety

- Class B Safety Library – IEC 60730
- For ASIL B and Beyond Applications – ISO 26262
- FMEDA Computation Spreadsheet (evaluation of Random Hardware Failures Metric)
- Functional Safety Manual
- Functional Safety Diagnostics Suite

Qualification Support

- AEC-Q100 REV-H (Grade 0: -40°C to +150°C) Compliant

TABLE 24-29: INTERNAL FRC ACCURACY

Operating Conditions: 3.0V to 3.6V (unless otherwise stated)
Operating temperature -40°C ≤ TA ≤ +85°C for Industrial
-40°C ≤ TA ≤ +125°C for Extended

Param No.	Characteristic	Min.	Max.	Units	Conditions
Internal FRC Accuracy @ FRC Frequency = 8 MHz[1]					
F20a	FRC	**-2**	**+2**	%	-40°C ≤ TA ≤ **-5**°C
		-1.5	+1.5	%	**-5**°C ≤ TA ≤ +85°C
		-2	+2	%	+85°C ≤ TA ≤ +125°C
F22	BFRC	-17	+17	%	-40°C ≤ TA ≤ +125°C

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

3. **Module: Guidelines for Getting Started with 16-Bit Digital Signal Controllers**

Additional information is added to **Section 2.5 "External Oscillator Pins"** and **Section 2.6 "External Oscillator Layout Guidance"** is added.

2.5 External Oscillator Pins

When the Primary Oscillator (POSC) circuit is used to connect a crystal oscillator, special care and consideration is needed to ensure proper operation. The POSC circuit should be tested across the environmental conditions that the end product is intended to be used. The load capacitors specified in the crystal oscillator data sheet can be used as a starting point, however, the parasitic capacitance from the PCB traces can affect the circuit and the values may need to be altered to ensure proper start-up and operation.

Excessive trace length and other physical interaction can lead to poor signal quality. Poorly tuned oscillator circuits can have reduced amplitude, incorrect frequency (runt pulses), distorted waveforms and long start-up times that may result in unpredictable application behavior, such as instruction misexecution, illegal op code fetch, etc. Ensure that the crystal oscillator circuit is at full amplitude and correct frequency before the system begins to execute code. In planning the application's routing and I/O assignments, ensure that adjacent port pins, and other signals in close proximity to the oscillator, do not have high frequencies, short rise and fall times and other similar noise. For further information on the Primary Oscillator, see Primary Oscillator (POSC).

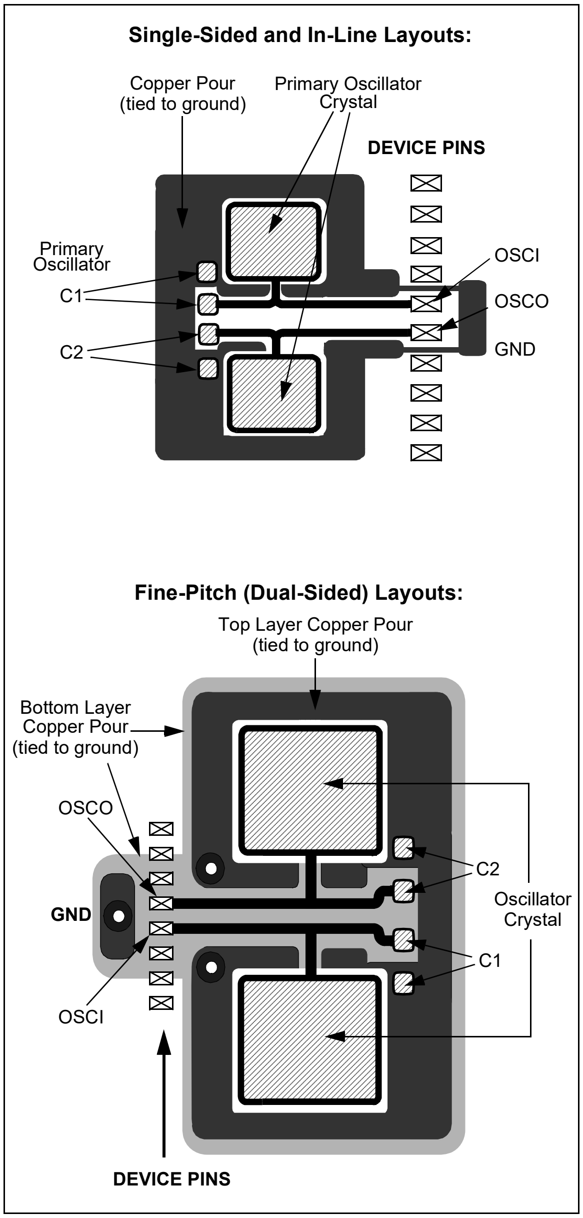
2.6 External Oscillator Layout Guidance

Use best practices during PCB layout to ensure robust start-up and operation. The oscillator circuit should be placed on the same side of the board as the device. Also, place the oscillator circuit close to the respective oscillator pins, not exceeding one-half inch (12 mm) distance between them. The load capacitors should be placed next to the oscillator itself, on the same side of the board. Use a grounded copper pour around the oscillator circuit to isolate it from surrounding circuits. The grounded copper pour should be routed directly to the MCU ground. Do not run any signal traces or power traces inside the ground pour. If using a two-sided board, avoid any traces on the other side of the board where the crystal is placed. Suggested layouts are shown in Figure 2-2. With fine-pitch packages, it is not always possible to completely surround the pins and components. A suitable solution is to tie the broken guard sections to a mirrored ground layer. In all cases, the guard trace(s) must be returned to ground.

For additional information and design guidance on oscillator circuits, please refer to these Microchip Application Notes, available at the Microchip website (www.microchip.com):

- *AN943, "Practical PICmicro® Oscillator Analysis and Design"*
- *AN949, "Making Your Oscillator Work"*
- *AN1798, "Crystal Selection for Low-Power Secondary Oscillator*

FIGURE 2-2: SUGGESTED PLACEMENT OF THE OSCILLATOR CIRCUIT


Single-Sided and In-Line Layouts:
Copper Pour (tied to ground)
Primary Oscillator Crystal
DEVICE PINS
Primary Oscillator
C1
C2
OSCI
OSCO
GND
Fine-Pitch (Dual-Sided) Layouts:
Top Layer Copper Pour (tied to ground)
Bottom Layer Copper Pour (tied to ground)
OSCO
GND
OSCI
C2
Oscillator Crystal
C1
DEVICE PINS

4. Module: dsPIC® Core Naming Convention

When referring to the dsPIC cores implemented in a dual core device, all instances of “Master” and “Slave” have been replaced with “Main” and “Secondary”, respectively. This includes the Master Slave Interface (MSI) module which is now named the Main Secondary Interface (MSI) module.

APPENDIX A: DOCUMENT REVISION HISTORY

Rev A Document (5/2018)

Initial release of this document; issued for revision A2.

Rev B Document (11/2018)

Adds silicon issue 23 (CPU), 24 (SCCP/MCCP), 25 (I^2C) and 26 (I/O).

Rev C Document (10/2019)

Adds silicon issue 27 (CPU), 28 (CPU), 29 (DMA) and 30 (PWM).

Updates silicon issue 26 (I/O).

Updates device data sheet reference to the current revision D.

Rev D Document (2/2020)

Adds silicon issue 31 (I^2C).

Rev E Document (6/2020)

Adds silicon issues 32 (Oscillator) and 33 (Main Secondary Interface (MSI)).

Adds data sheet clarification 1 (Electrical Characteristics).

Removes silicon issue 8 (Oscillator) since it is no longer applicable.

Rev F Document (10/2020)

Updates silicon issue 32 (Oscillator).

Adds silicon issue 34 (Secondary CPU).

Adds data sheet clarifications 2 (Functional Safety and Qualification Support), 3 (Guidelines for Getting Started with 16-Bit Digital Signal Controllers) and 4 (dsPIC® Core Naming Convention).

NOTES:

Information contained in this publication is provided for the sole purpose of designing with and using Microchip products. Information regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications.

THIS INFORMATION IS PROVIDED BY MICROCHIP "AS IS". MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL OR CONSEQUENTIAL LOSS, DAMAGE, COST OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP'S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

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

Trademarks

ISBN: 978-1-5224-6986-5

Worldwide Sales and Service

AMERICAS

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

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

Austin, TX
Tel: 512-257-3370

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

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

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

Detroit
Novi, MI
Tel: 248-848-4000

Houston, TX
Tel: 281-894-5983

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

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

Raleigh, NC
Tel: 919-844-7510

New York, NY
Tel: 631-435-6000

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

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

ASIA/PACIFIC

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

China - Xiamen
Tel: 86-592-2388138

China - Zhuhai
Tel: 86-756-3210040

ASIA/PACIFIC

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

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

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

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

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

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

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

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

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

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

Singapore
Tel: 65-6334-8870

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

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

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

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

Vietnam - Ho Chi Minh
Tel: 84-28-5448-2100

EUROPE

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

Denmark - Copenhagen
Tel: 45-4485-5910
Fax: 45-4485-2829

Finland - Espoo
Tel: 358-9-4520-820

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

Germany - Garching
Tel: 49-8931-9700

Germany - Haan
Tel: 49-2129-3766400

Germany - Heilbronn
Tel: 49-7131-72400

Germany - Karlsruhe
Tel: 49-721-625370

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

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

Israel - Ra'anana
Tel: 972-9-744-7705

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

Italy - Padova
Tel: 39-049-7625286

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

Norway - Trondheim
Tel: 47-7288-4388

Poland - Warsaw
Tel: 48-22-3325737

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

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

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

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

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

02/28/20