PIC18F97J60 Family Silicon Errata and Data Sheet Clarification

The PIC18F97J60 Family devices that you have received conform functionally to the current Device Data Sheet (DS39762**F**), 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 PIC18F97J60 Family silicon.

Note:	This document summarizes all silicon errata issues from all revisions of silicon, previous as well as current. Only the issues indicated in the last column of Table 2 apply to the current silicon revision (**A3**).

Data Sheet clarifications and corrections start on page 8, 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 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 PIC18F97J60 Family silicon revisions are shown in Table 1.

TABLE 1: SILICON DEVREV VALUES

Part Number	DEVICE IDs(1),(2)					
	Device ID	Silicon Revision ID				
		A0	A1	A2	A3	
PIC18F66J60	0001 1000 000	0000	0 0001	0 0001	0 0011
PIC18F66J65	0001 1111 000	0000	0 0001	0 0001	0 0011
PIC18F67J60	0001 1111 001	0000	0 0001	0 0001	0 0011
PIC18F86J60	0001 1000 001	0000	0 0001	0 0001	0 0011
PIC18F86J65	0001 1111 010	0000	0 0001	0 0001	0 0011
PIC18F87J60	0001 1111 011	0000	0 0001	0 0001	0 0011
PIC18F96J60	0001 1000 010	0000	0 0001	0 0001	0 0011
PIC18F96J65	0001 1111 100	0000	0 0001	0 0001	0 0011
PIC18F97J60	0001 1111 101	0000	0 0001	0 0001	0 0011

Note 1: The Device IDs (DEVID1 and DEVUD2) are located at addresses 3 FFFFEh:3FFFFFh, in the device configuration space. They are shown in binary in the format: 'DEVID2 DEVID1'.

2: Refer to the *"PIC18F97J60 Family Flash Microcontroller Programming Specification"* (DS39688) 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]			
				A0	A1	A2	A3
Resets	SFR Reset values	1.	$\overline{\text{MCLR}}$ and BOR Resets behave as POR Reset	X	X	X	X
I/O (PORTJ)	Port pin's impedance state	2.	PORTJ pins do not go to high-impedance state after POR Reset	X	X	X	X
I/O (PORTJ) and External Memory Bus	External Memory Bus	3.	In EMB mode PORTJ pins are not driven to Idle state	X	X	X	X
Ethernet (Buffer Memory)	Buffer memory corruption	4.	When an even address is loaded to the ERXRDPT registers, then the circular receive buffer may be corrupt	X	X	X	X
Ethernet (MIIM)	PHY register corruption	5.	MIWRL register written to PHY register through the MIIM interface can cause corruption	X	X	X	X
Ethernet (RX Filter)	Receive Filter Pattern Match	6.	RX filter may allow some packets with an incorrect data pattern to be received	X	X	X	X
Ethernet (TX)	Timing collision	7.	When operating in Half-Duplex mode, the transmit operation can encounter unusual collision timing	X	X	X	X
Ethernet (DMA)	IP Checksum computation	8.	DMA configured to compute an IP Checksum, incoming packet receive event can cause an internal deadlock	X	X	X	X
I/O (PORTJ)	Weak pull-up on Port pins	9.	RJ4 and RJ5 weak internal pull-up cannot be enabled	X	X	X	X
Timer1/3	Timer1/3 Asynchronous mode	10.	Timer 1/3 operation in Asynchronous External Input mode operation can cause an unexpected interrupt flag	X	X	X	X

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

Silicon Errata Issues

1. Module: Resets

$\overline{MCLR}$ and BOR Resets behave as a POR Reset. Special Function Registers' Reset values after a $\overline{MCLR}$ or BOR would have the same values as those after a POR. All other Resets behave as described in the data sheet.

Work around

None.

Affected Silicon Revisions

A0	A1	A2	A3				
X	X	X	X				

2. Module: I/O (PORTJ)

> **Note:** This issue is only applicable to the 100-pin device.

When configured to operate in Microcontroller mode (CONFIG3L<EMB1:0> = `11`), PORTJ pins do not go to a high-impedance state immediately after a POR Reset. Instead, PORTJ<4,0> are driven low, while all other PORTJ pins are driven high, until the device exits the Reset condition (refer to **Section 4.6.1 "Time-out Sequence"** of the Device Data Sheet for details on when the device exits the Reset condition) before transitioning to a high-impedance state. Note that since $\overline{MCLR}$ and BOR Resets are also treated as a POR Reset (see Errata Issue #1), PORTJ pins will also be driven as outputs until the device exits these Reset conditions.

Work around

If using a PORTJ pin as an input, make sure to check that your circuit will not create a short-circuit condition during a Reset. For example, if you need to have a direct pull-down to ground input, do this on PORTJ<4> or PORTJ<0>, since they are temporary driven low. If using a PORTJ pin as an output, then use a pin that will temporarily drive low for driving active-high loads, and use a pin that temporarily will drive high for driving active-low loads. This way, the temporary output signals are in the Idle state.

Affected Silicon Revisions

A0	A1	A2	A3				
X	X	X	X				

3. Module: I/O (PORTJ) and External Memory Bus

> **Note:** This issue is only applicable to the 100-pin device.

In an Extended Microcontroller mode (CONFIG3L<EMB1:0> = `00`, `01` or `10`), each control signal on PORTJ is supposed to be driven to its Idle state. However, the control signals on PORTJ pins go to a high-impedance state for a brief interval after a $\overline{MCLR}$ Reset. The brief loss of control signals may cause the corruption of data in memory devices connected to the External Memory Bus (EMB).

Work around

To maintain the default states on the control lines, use pull-up or pull-down resistors on all PORTJ pins (pull-down on PORTJ<4,0>, pull-up on all others).

Affected Silicon Revisions

A0	A1	A2	A3				
X	X	X	X				

4. Module: Ethernet (Buffer Memory)

The receive hardware may corrupt the circular receive buffer (including the Next Packet Pointer and receive status vector fields) when an even value is programmed into the ERXRDPTH:ERXRDPTL registers.

Work around

Ensure that only odd addresses are written to the ERXRDPT registers. Assuming that ERXND contains an odd value, many applications can derive a suitable value to write to ERXRDPT by subtracting 1 from the Next Packet Pointer (a value always ensured to be even because of hardware padding) and then compensating for a potential ERXST to ERXND wrap-around.

Assuming that the receive buffer area does not span the 1FFFh to 0000h memory boundary, the logic in Example 1 will ensure that ERXRDPT is programmed with an odd value.

EXAMPLE 1: WRITING OLD ADDRESSES TO ERXRDPT

```
if (Next Packet Pointer – 1 < ERXST) or
(Next Packet Pointer – 1 > ERXND)
    then:
ERXRDPT = ERXND
    else:
ERXRDPT = Next Packet Pointer – 1
```

Affected Silicon Revisions

A0	A1	A2	A3				
X	X	X	X				

5. Module: Ethernet (MIIM)

When writing to any PHY register through the MIIM interface's MIWRL and MIWRH registers, the low byte actually written to the PHY register may be corrupted. The corruption occurs when the following actions are taken:

- The application writes to MIWRL
- The PIC® MCU core executes any instruction that reads or writes to any memory address that has the Least Significant six address bits of 36h (`'b110110`)
- The application writes to MIWRH

For example, the following sequence will result in a corrupted write to a PHY register:

```
MOVFF    0xCF5, MIWRL
NOP
MOVFF    0xCF6, MIWRH
```

In this example, 0xCF5 and 0xCF6 are GPR memory locations that the application wishes to write to the current PHY register defined by the MIREGADR SFR. When the PIC MCU core reads from the GPR at address, 0xCF6 (`'b110011110110`), the value originally written to MIWRL will be corrupted.

Work around 1

Ensure that following a write to MIWRL, the firmware does not access any of the problem memory locations prior to writing to MIWRH. After finished writing to MIWRH, normal operation can resume.

If interrupts are enabled, disable them prior to writing to MIWRL and MIWRH to prevent an Interrupt Service Routine (ISR) from performing any reads or writes to a problem memory address.

Special care must be taken to ensure that the source data to be written to MIWRH does not result in a problem memory access.

The following PHY write sequence avoids the problem:

1. Copy the low byte to be written to the PHY into the PRODL register.
   PRODL is at address, FF3h, and not subject to the memory address issue.
2. Copy the high byte to be written to the PHY into the PRODH register.
   PRODH is at address, FF4h, and not subject to the memory address issue.
3. Disable all interrupts by clearing GIEH and GIEL in the INTCON register.
4. Move PRODL into MIWRL.
5. Wait one instruction cycle as required by the MAC host interface logic.
6. Move PRODH into MIWRH.
7. Enable all interrupts that are needed by restoring GIEH and GIEL in INTCON.

Work around 2

If you cannot disable interrupts, as specified in **Work around 1**, because the application cannot tolerate interrupt latency variations:

- Perform the write (with interrupts enabled), but
- Verify the correct values were written by reading the PHY register

If a corrupted value was written due to an interrupt occurring, perform the write again and reverify. The source data must be stored in a non-problem location.

The application should follow the following procedure:

1. Copy the low byte to be written to the PHY into the PRODL register.

   PRODL is at address, FF3h, and not subject to the memory address issue.
2. Copy the high byte to be written to the PHY into the PRODH register.

   PRODH is at address, FF4h, and not subject to the memory address issue.
3. Move PRODL into MIWRL.
4. Wait one instruction cycle as required by the MAC host interface logic.
5. Move PRODH into MIWRH.
6. Wait two T_{CY} and then poll the BUSY bit (MISTAT<0>) until it is clear.
7. Perform a PHY register read of the same location.
8. Compare the read result with the original value copied to the PRODH:PRODL registers. If they do not match, return to step 1.

Affected Silicon Revisions

A0	A1	A2	A3				
X	X	X	X				

6. Module: Ethernet (RX Filter)

When enabled, the Pattern Match receive filter may allow some packets with an incorrect data pattern to be received. Also, in certain configurations, packets with a valid pattern may be incorrectly discarded.

Work around

Do not use the Pattern Match hardware filter. Instead, use the Unicast, Mutlicast, Broadcast and Hash Table receive filters to accept all needed packets and filter out unwanted ones in software.

Affected Silicon Revisions

A0	A1	A2	A3				
X	X	X	X				

7. Module: Ethernet (TX)

When configured for half duplex and a transmit operation encounters unusual collision timing, there is a small chance that the Ethernet transmit engine will internally deadlock. The PHY will stop transmitting the packet and normal RX operations will continue. However, the TXRTS bit (ECON1<3>) will stay set indefinitely. The TXIF (EIR<3>) and TXERIF (EIR<1>) bits will not become set.

This deadlock condition applies only to half-duplex operation and is most readily observable when the network has a duplex mismatch (i.e., PIC18F97J60 family device is configured for half duplex and the remote node is configured for full duplex). In most cases, high network utilization is needed to observe the issue.

Work around

To prevent most transmit deadlock conditions, issue a TX Logic Reset prior to transmitting each packet:

1. Set TXRST (ECON1<7>).
2. Clear TXRST.
3. Wait 1.6 μs or longer.
4. Set TXRTS to start the transmission.

Issuing a TX Logic Reset may cause the Ethernet transmit error interrupt to occur and the associated TXERIF bit to become set, which can be ignored.

To detect and recover from any possible deadlock conditions, applications should implement a timer to poll the TXRTS bit. If the Ethernet hardware enters the deadlock state and fails to clear this bit by the time the timer expires, software should manually clear the TXRTS bit, issue a TX Logic Reset and then set the TXRTS bit to retry transmission. The timer should be cleared and restarted whenever the application sets TXRTS. The timer expiration time should be chosen to allow adequate time for ordinary packets to finish transmitting, after accounting for possible delays, due to the medium being occupied by other nodes. For example, a time-out value of 3 ms is suitable since it will allow a maximum length 1518-byte packet to be transmitted at 10Base-T speeds, while giving reasonable margin to account for potential collisions.

Affected Silicon Revisions

A0	A1	A2	A3				
X	X	X	X				

8. Module: Ethernet (DMA)

When the DMA is configured to compute an IP checksum, there is a small chance that an incoming packet receive event will cause the DMA to internally deadlock. In these cases, the DMAST bit (ECON1<5>) stays set indefinitely, and the DMA done interrupt never occurs.

Work around

Perform checksum calculations in software. Use the DMA only for copy operations.

Affected Silicon Revisions

A0	A1	A2	A3				
X	X	X	X				

9. Module: I/O (PORTJ)

Note:	This issue is only applicable to the 80-pin device.

The weak internal pull-up resistors on pins, RJ4 and RJ5, cannot be enabled on the PIC18F86J60, PIC18F86J65 and PIC18F87J60 devices. Setting the RJPU bit (PORTA<7>) has no effect on the I/O pin state.

Work around

Install external pull-up resistors on RJ4 and RJ5. Alternatively, use any of the PORTB, PORTD or PORTE pins, which all have weak internal pull ups.

Affected Silicon Revisions

A0	A1	A2	A3				
X	X	X	X				

10. Module: Timer1/3

When Timer1 or Timer3 is operated in Asynchronous External Input mode, unexpected interrupt flag generation may occur if an external clock edge arrives too soon following a firmware write to the TMRxH:TMRxL registers. An unexpected interrupt flag event may also occur when enabling the module or switching from Synchronous to Asynchronous mode.

Work around

This issue only applies when operating the timer in Asynchronous mode. Whenever possible, operate the timer module in Synchronous mode to avoid spurious timer interrupts.

If Asynchronous mode must be used in the application, potential strategies to mitigate the issue may include any of the following:

- Design the firmware so it does not rely on the TMRxIF flag or keep the respective interrupt disabled. The timer still counts normally and does not reset to `0x0000` when the spurious interrupt flag event is generated.
- Design the firmware so that it does not write to the TMRxH:TMRxL registers or does not periodically disable/enable the timer, or switch modes. Reading from the timer does not trigger the spurious interrupt flag events.
- If the firmware must use the timer interrupts and must write to the timer (or disable/enable, or mode switch the timer), implement code to suppress the spurious interrupt event, should it occur. This can be achieved by following the process shown in Example 2.

EXAMPLE 2: ASYNCHRONOUS TIMER MODE WORK AROUND TO AVOID SPURIOUS INTERRUPT

```
//Timer1 update procedure in asynchronous mode
//The code below uses Timer1 as example

T1CONbits.TMR1ON = 0;         //Stop timer from incrementing
PIE1bits.TMR1IE = 0;          //Temporarily disable Timer1 interrupt vectoring
TMR1H = 0x00;                 //Update timer value
TMR1L = 0x00;
T1CONbits.TMR1ON = 1;         //Turn on timer

//Now wait at least two full T1CKI periods + 2TCY before re-enabling Timer1 interrupts.
//Depending upon clock edge timing relative to TMR1H/TMR1L firmware write operation,
//a spurious TMR1IF flag event may sometimes assert. If this happens, to suppress
//the actual interrupt vectoring, the TMR1IE bit should be kept clear until
//after the "window of opportunity" (for the spurious interrupt flag event has passed).
//After the window is passed, no further spurious interrupts occur, at least
//until the next timer write (or mode switch/enable event).

while(TMR1L < 0x02);          //Wait for 2 timer increments more than the Updated Timer
                              //value (indicating more than 2 full T1CKI clock periods elapsed)
NOP();                        //Wait two more instruction cycles
NOP();
PIR1bits.TMR1IF = 0;          //Clear TMR1IF flag, in case it was spuriously set
PIE1bits.TMR1IE = 1;          //Now re-enable interrupt vectoring for timer 1
```

Affected Silicon Revisions

A0	A1	A2	A3				
X	X	X	X				

Data Sheet Clarifications

The following typographic corrections and clarifications are to be noted for the latest version of the device data sheet (DS39762**F**).

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

1. Module: Thermal Packaging Characteristics

The thermal packaging characteristics are the following

TABLE 3: THERMAL PACKAGING CHARACTERISTICS

Package	Pin Count	Package Body Size	Thermal Resistance Estimates (°C/W) (3)	
			θ_{JA} (typical)(1)	θ_{JC} (typical) (2)
TQFP	64	10x10x1	48.3	26.1
TQFP	80	12x12x1	48.0	22.5
TQFP	100	12x12x1	45	19.7
TQFP	100	14x14x1	43	22.5

Note 1: Junction to ambient thermal resistance, Theta-JA (θ_{JA}) numbers are achieved by package simulations at 0m/s airflow (worst-case condition).

2: Junction to case thermal resistance, Theta-JC (θ_{JC}) numbers are achieved by package simulations defined at the top center of the package (hottest point).

3: Estimates only; dependent on many factors.

4: Theta-JA (θ_{JA}) and Theta-JC (θ_{JC}) numbers is package specific and not device specific.

2. Module: Internal Voltage Regulator Specifications

In Table 28-4, the minimum external filter capacitor should be 4.7 µF:

TABLE 28-4: INTERNAL VOLTAGE REGULATOR SPECIFICATIONS

Operating Conditions: -40°C ≤ TA ≤ +85°C (unless otherwise stated)							
Param No.	Sym.	Characteristics	Min.	Typ.	Max.	Units	Comments
	VRGOUT	Regulator Output Voltage	—	2.5	—	V	
	CF	External Filter Capacitor Value	**4.7**	10	—	µF	**Capacitor must be low ESR, a low series resistance (<5Ω)**

3. Module: Clock and I/O Timing Requirements

In Table 28-10, the values in ‘Typ.’ column for 164 and 166 parameters have modified as follows:

TABLE 28-10: CLKO AND I/O TIMING REQUIREMENTS

Param. No	Symbol	Characteristics	Min.	Typ.	Max.	Units
150	TadV2alL	Address Out Valid to ALE ↓ (address setup time)	0.25 TCY – 10	—	—	ns
151	TalL2adl	ALE ↓ to Address Out Invalid (address hold time)	5	—	—	ns
155	TalL2oeL	ALE ↓ to $\overline{OE}$ ↓	10	0.125 TCY	—	ns
160	TadZ2oeL	AD high-Z to $\overline{OE}$ ↓ (bus release to $\overline{OE}$)	0	—	—	ns
161	ToeH2adD	$\overline{OE}$ ↑ to AD Driven	0.125 TCY – 5	—	—	ns
162	TadV2oeH	Least Significant Data Valid before $\overline{OE}$ ↑ (data setup time)	20	—	—	ns
163	ToeH2adl	$\overline{OE}$ ↑ to Data In Invalid (data hold time)	0	—	—	ns
164	TalH2alL	ALE Pulse Width	—	**0.25 TCY**	—	ns
165	ToeL2oeH	$\overline{OE}$ Pulse Width	0.5 TCY – 5	0.5 TCY	—	ns
166	TalH2alH	ALE ↑ to ALE ↑ (cycle time)	—	**TCY**	—	ns
167	Tacc	Address Valid to Data Valid	0.75 TCY – 25	—	—	ns
168	Toe	$\overline{OE}$ ↓ to Data Valid		—	0.5 TCY – 25	ns
169	TalL2oeH	ALE ↓ to $\overline{OE}$ ↑	0.625 TCY – 10	—	0.625 TCY + 10	ns
171	TalH2csL	Chip Enable Active to ALE ↓	0.25 TCY – 20	—	—	ns
171A	TubL2oeH	AD Valid to Chip Enable Active	—	—	10	ns

4. Module: Brown-Out Reset Pulse Width

In Table 28-12, the 35 parameter has been added as follows:

TABLE 28-12: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER AND BROWN-OUT RESET REQUIREMENTS

Param. No.	Symbol	Characteristic	Min.	Typ.	Max.	Units	Conditions
30	TMCL	$\overline{\text{MCLR}}$ Pulse Width (low)	2	—	—	μs	
31	TWDT	Watchdog Timer Time-out Period (no postscaler)	2.8	4.1	5.4	ms	
32	TOST	Oscillation Start-up Timer Period	1024 TOSC	—	1024 TOSC	—	TOSC = OSC1 period
33	TPWRT	Power-up Timer Period	46.2	66	85.8	ms	
34	TIOZ	I/O High-Impedance from $\overline{\text{MCLR}}$ Low or Watchdog Timer Reset	—	—	3 TCY + 2	μs	System clock available
			—	—	415	μs	System clock unavailable (Sleep mode or primary oscillator off)
35	**TBOR**	**Brown-out Reset Pulse Width**	**200**	**—**	**—**	**μs**	**VDD ≤ Brown-out Reset voltage (see D005)**
38	TCSD	CPU Start-up Time	—	200	—	μs	

5. Module: CCPxCON Register:0

In the CCPxCON Register, at CCPxM<3:0> bits, the `0011` should be read as 'Reserved mode', as follows:

REGISTER 18-1: CCPxCON: ENHANCED CCPx CONTROL REGISTER (ECCP1/ECCP2/ECCP3)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PxM1	PxM0	DCxB1	DCxB0	CCPxM3	CCPxM2	CCPxM1	CCPxM0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-6 **PxM<1:0>:** Enhanced PWM Output Configuration bits

If CCPxM<3:2> = `00`, `01`, `10`:
`xx` = PxA assigned as Capture/Compare input/output; PxB, PxC, PxD assigned as port pins

If CCPxM<3:2> = `11`:
`00` = Single output: PxA modulated; PxB, PxC, PxD assigned as port pins
`01` = Full-bridge output forward: P1D modulated; P1A active; P1B, P1C inactive
`10` = Half-bridge output: P1A, P1B modulated with dead-band control; P1C, P1D assigned as port pins
`11` = Full-bridge output reverse: P1B modulated; P1C active; P1A, P1D inactive

bit 5-4 **DCxB<1:0>**: ECCPx Module PWM Duty Cycle Bit 1 and Bit 0

Capture mode:
Unused.

Compare mode:
Unused.

PWM mode:
These bits are the 2 LSbs of the 10-bit PWM duty cycle. The 8 MSbs of the duty cycle are found in CCPRxL.

bit 3-0 **CCPxM<3:0>**: ECCPx Module Mode Select bits

`0000` = Capture/Compare/PWM disabled (resets ECCPx module)
`0001` = Reserved
`0010` = Compare mode; toggle output on match
`0011` = **Reserved mode**
`0100` = Capture mode; every falling edge
`0101` = Capture mode; every rising edge
`0110` = Capture mode; every 4th rising edge
`0111` = Capture mode; every 16th rising edge
`1000` = Compare mode; initialize ECCPx pin low; set output on compare match (set CCPxIF)
`1001` = Compare mode; initialize ECCPx pin high; clear output on compare match (set CCPxIF)
`1010` = Compare mode; generate software interrupt only, ECCPx pin reverts to I/O state)
`1011` = Compare mode; trigger special event (ECCPx resets TMR1 or TMR3, sets CCPxIF bit, ECCPx trigger also starts A/D conversion if A/D module is enabled)[(1)]
`1100` = PWM mode; PxA, PxC active-high; PxB, PxD active-high
`1101` = PWM mode; PxA, PxC active-high; PxB, PxD active-low
`1110` = PWM mode; PxA, PxC active-low; PxB, PxD active-high
`1111` = PWM mode; PxA, PxC active-low; PxB, PxD active-low

Note 1: Implemented only for ECCP1 and ECCP2; same as '`1010`' for ECCP3.

APPENDIX A: REVISION HISTORY

Rev A Document (2/2009)

Original version of this document. Includes silicon issues 1 (Resets), 2 (I/O – PORTJ), 3 (I/O (PORTJ) and External Memory Bus), 4 (Ethernet – Buffer Memory), 5 (Ethernet – MIIM), 6 (Ethernet – RX Filter), 7 (Ethernet – TX), 8 (Ethernet – DMA) and 9 (I/O – PORTJ).

Rev B Document (3/2010)

Changed the title to A1/A2 as this document now also covers the A2 silicon. Updated the data sheet reference revision from “D” to “E”.

Rev C Document (10/2010)

Merged silicon errata documents for Revision A0 and Revision A1/A2. Updated all affected silicon errata for each module to include Revision A3. Merged data sheet errata to create a single errata document for this part.

Rev D Document (7/2014)

Added Module 10, Timer1/3, to Silicon Errata Issues section.

Data Sheet Clarifications: Removed Module 1.

Rev E Document (12/2014)

Data Sheet Clarifications: Added Modules 1-5; Updated the Errata document to new format; Other minor corrections.

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

ISBN: 978-1-63276-852-0

QUALITY MANAGEMENT SYSTEM CERTIFIED BY DNV

== ISO/TS 16949 ==

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.

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

Cleveland
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

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

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

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110

Canada - Toronto
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: 852-2943-5100
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 - 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: 86-21-5407-5533
Fax: 86-21-5407-5066

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

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

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

ASIA/PACIFIC

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

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

Germany - Dusseldorf
Tel: 49-2129-3766400

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

Germany - Pforzheim
Tel: 49-7231-424750

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

Italy - Venice
Tel: 39-049-7625286

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

Poland - Warsaw
Tel: 48-22-3325737

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

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

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

03/25/14