

---

## Section 11. I/O Ports

---

### HIGHLIGHTS

This section of the manual contains the following topics:

11.1	Introduction .....	11-2
11.2	I/O Port Control Registers .....	11-3
11.3	Peripheral Multiplexing .....	11-4
11.4	Port Descriptions .....	11-6
11.5	Change Notification (CN) Pins .....	11-7
11.6	CN Operation in Sleep and Idle Modes .....	11-8
11.7	Related Application Notes .....	11-11
11.8	Revision History .....	11-12

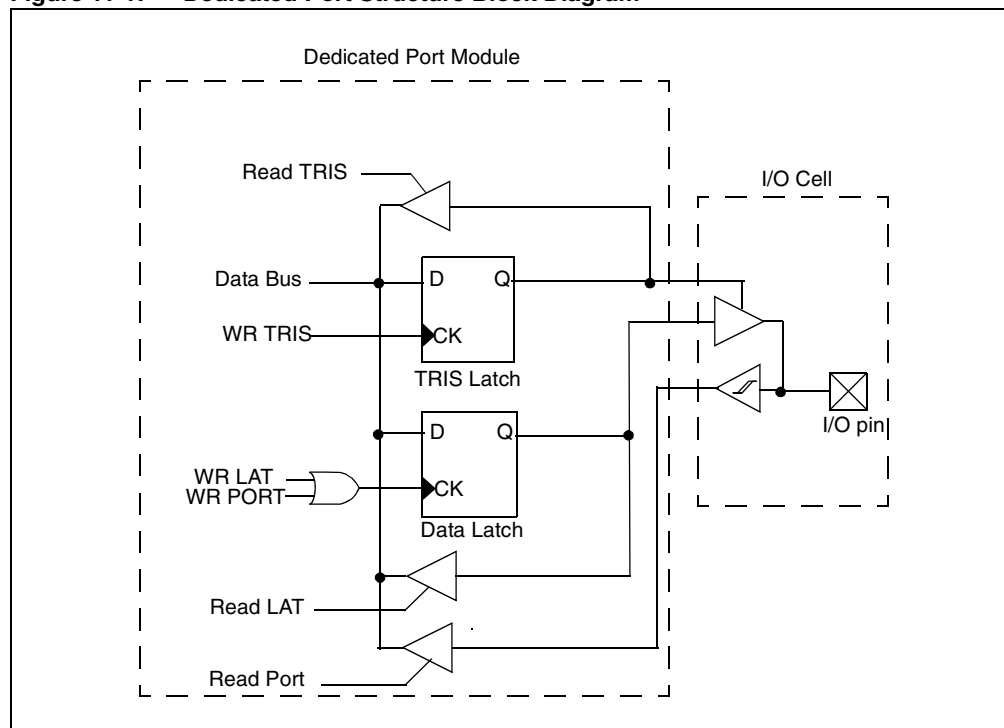
## 11.1 Introduction

This section provides information on the I/O ports for the dsPIC30F family of devices. All of the device pins (except V<sub>DD</sub>, V<sub>SS</sub>, MCLR, and OSC1/CLKI) are shared between the peripherals and the general purpose I/O ports.

The general purpose I/O ports allow the dsPIC30F to monitor and control other devices. Most I/O pins are multiplexed with alternate function(s). The multiplexing will depend on the peripheral features on the device variant. In general, when a peripheral is functioning, that pin may not be used as a general purpose I/O pin.

Figure 11-1 shows a block diagram of a typical I/O port. This block diagram does not take into account peripheral functions that may be multiplexed onto the I/O pin.

**Figure 11-1: Dedicated Port Structure Block Diagram**



## 11.2 I/O Port Control Registers

All I/O ports have three registers directly associated with the operation of the port, where 'x' is a letter that denotes the particular I/O port:

- TRISx: Data Direction register
- PORTx: I/O Port register
- LATx: I/O Latch register

Each I/O pin on the device has an associated bit in the TRIS, PORT and LAT registers.

**Note:** The total number of ports and available I/O pins will depend on the device variant. In a given device, all of the bits in a port control register may not be implemented. Refer to the specific device data sheet for further details.

### 11.2.1 TRIS Registers

The TRISx register control bits determine whether each pin associated with the I/O port is an input or an output. If the TRIS bit for an I/O pin is a '1', then the pin is an input. If the TRIS bit for an I/O pin is a '0', then the pin is configured for an output. An easy way to remember is that a '1' looks like an I (input) and a '0' looks like an O (output). All port pins are defined as inputs after a Reset.

### 11.2.2 PORT Registers

Data on an I/O pin is accessed via a PORTx register. A read of the PORTx register reads the value of the I/O pin, while a write to the PORTx register writes the value to the port data latch.

Many instructions, such as BSET and BCLR instructions, are read-modify-write operations. Therefore, a write to a port implies that the port pins are read, this value is modified, and then written to the port data latch. Care should be taken when read-modify-write commands are used on the PORTx registers and when some I/O pins associated with the port are configured as inputs. If an I/O pin configured as an input is changed to an output at some later time, an unexpected value may be output on the I/O pin. This effect occurs because the read-modify-write instruction reads the instantaneous value on the input pin and loads that value into the port data latch.

### 11.2.3 LAT Registers

The LATx register associated with an I/O pin eliminates the problems that could occur with read-modify-write instructions. A read of the LATx register returns the values held in the port output latches, instead of the values on the I/O pins. A read-modify-write operation on the LAT register, associated with an I/O port, avoids the possibility of writing the input pin values into the port latches. A write to the LATx register has the same effect as a write to the PORTx register.

The differences between the PORT and LAT registers can be summarized as follows:

- A write to the PORTx register writes the data value to the port latch.
- A write to the LATx register writes the data value to the port latch.
- A read of the PORTx register reads the data value on the I/O pin.
- A read of the LATx register reads the data value held in the port latch.

Any bit and its associated data and control registers that are not valid for a particular device will be disabled. That means the corresponding LATx and TRISx registers, and the port pin, will read as zeros.

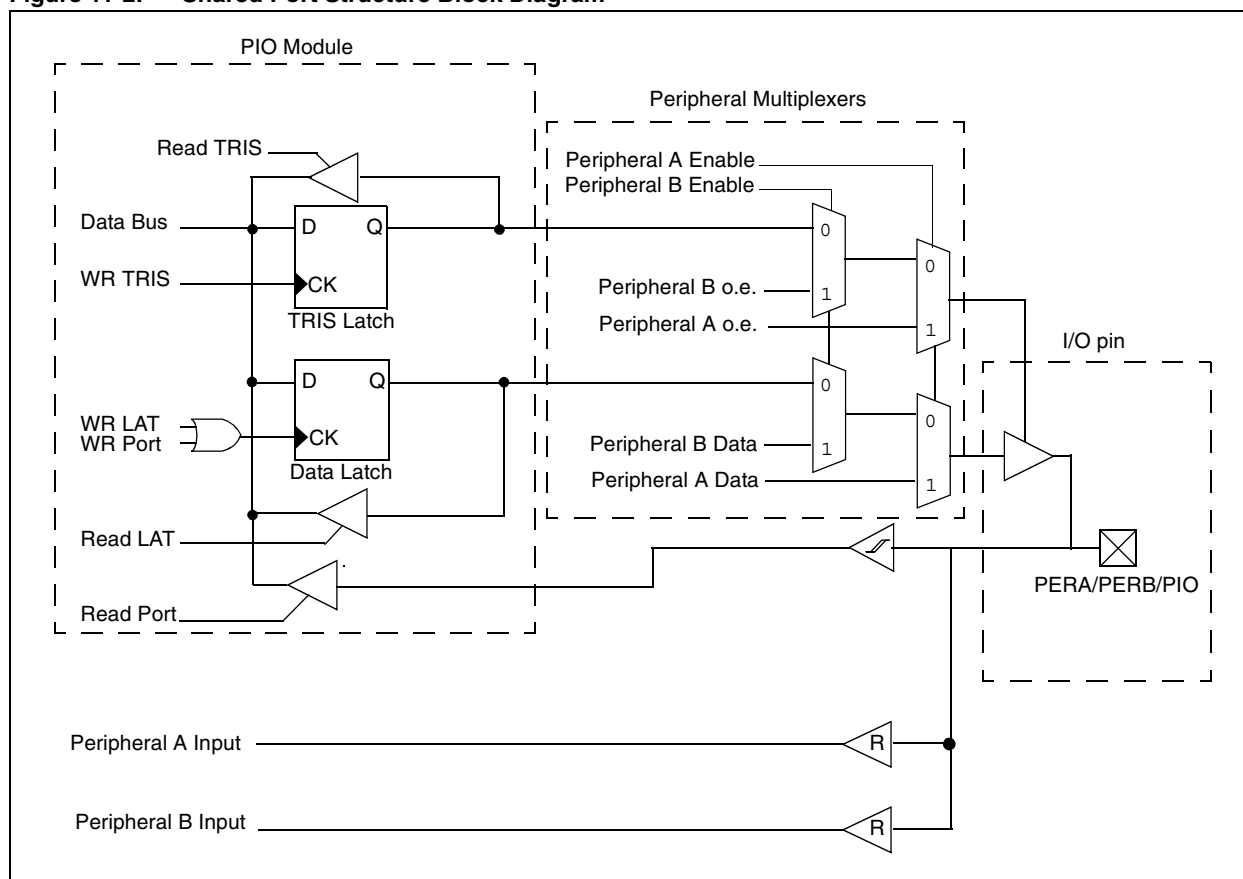
## 11.3 Peripheral Multiplexing

When a peripheral is enabled the associated pin output drivers are typically module controlled while a few are user settable. The I/O pin may be read through the input data path, but the output driver for the I/O port bit is generally disabled.

An I/O port that shares a pin with another peripheral is always subservient to the peripheral. The peripheral's output buffer data and control signals are provided to a pair of multiplexers. The multiplexers select whether the peripheral, or the associated port, has ownership of the output data and control signals of the I/O pin. Figure 11-2 shows how ports are shared with other peripherals, and the associated I/O pin to which they are connected.

**Note:** In order to use PORTB pins for digital I/O, the corresponding bits in the ADPCFG register must be set to '1', even if the A/D module is turned off.

**Figure 11-2: Shared Port Structure Block Diagram**



### 11.3.1 I/O Multiplexing with Multiple Peripherals

For some dsPIC30F devices, especially those with a small number of I/O pins, multiple peripheral functions may be multiplexed on each I/O pin. Figure 11-2 shows an example of two peripherals multiplexed to the same I/O pin.

The name of the I/O pin defines the priority of each function associated with the pin. The conceptual I/O pin, shown in Figure 11-2, has two multiplexed peripherals, 'Peripheral A' and 'Peripheral B' and is named "PERA/PERB/PIO".

The I/O pin name is chosen so that the user can easily tell the priority of the functions assigned to the pin. For the example shown in Figure 11-2, Peripheral A has the highest priority for control of the pin. If Peripheral A and Peripheral B are enabled at the same time, Peripheral A will take control of the I/O pins.

### 11.3.1.1 Software Input Pin Control

Some of the functions assigned to an I/O pin may be input functions that do not take control of the pin output driver. An example of one such peripheral is the Input Capture module. If the I/O pin associated with the Input Capture is configured as an output, using the appropriate TRIS control bit, the user can manually affect the state of the Input Capture pin through its corresponding PORT register. This behavior can be useful in some situations, especially for testing purposes, when no external signal is connected to the input pin.

Referring to Figure 11-2, the organization of the peripheral multiplexers will determine if the peripheral input pin can be manipulated in software using the PORT register. The conceptual peripherals shown in this figure disconnect the PORT data from the I/O pin when the peripheral function is enabled.

In general, the following peripherals allow their input pins to be controlled manually through the PORT registers:

- External Interrupt pins
- Timer Clock Input pins
- Input Capture pins
- PWM Fault pins

Most serial communication peripherals, when enabled, take full control of the I/O pin, so that the input pins associated with the peripheral cannot be affected through the corresponding PORT registers. These peripherals include the following:

- SPI™
- I<sup>2</sup>C™
- DCI
- UART
- CAN

### 11.3.1.2 Pin Control Summary

When a peripheral is enabled the associated pin output drivers are typically module controlled while a few are user settable. The term "Module Control" means that the associated port pin output driver is disabled and the pin can only be controlled and accessed by the peripheral. The term "User Settable" means that the associated peripheral port pin output driver is user configurable via the associated TRISx SFR. The TRISx register must be set properly for the peripheral to function properly. For "User Settable" peripheral pins, the actual port pin state can always be read via the PORTx SFR.

An Input Capture peripheral makes a good example of a User Settable peripheral. The user must write the associated TRIS register to configure the Input Capture pin as an input. Since the I/O pin circuitry is still active when the Input Capture is enabled, a 'trick' can be used to manually produce capture events using software. The Input Capture pin is configured as an output using the associated TRIS register. Then, the software can write values to the corresponding LAT register drive to internally control the Input Capture pin and force capture events.

As another example an INTx pin can be configured as an output and then by writing to the associated LATx bit an INTx interrupt, if enabled, can be generated.

The UART is an example of a Module Control peripheral. When the UART is enabled, the PORT and TRIS registers have no effect and cannot be used to read or write the RX and TX pins. Most communication peripheral functions available on the dsPIC are Module Control peripherals.

For example, the SPI module can be configured for Master mode in which only the SDO pin is required. In this scenario the SDI pin can be configured as a general purpose output pin by clearing (setting to a logic "0") the associated TRISx bit. Table 11-1 presents a summary of the dsPIC peripherals and associated Pin Output Control and Port pin read status.

**Table 11-1: Port Pin Control Summary Table**

Peripheral Module "Enabled State"	Peripheral Pins	TRISx - Pin Output Control	PORTx - Pin Read
SPI™ (x = 1 or 2)	SDOx	<DISSDO = 1>, User Settable <DISSDO = 0>, Module Control	Yes
	SDIx	User Settable	Yes
	SCKx	Module Control	Yes
	SSx	<SSEN = 0>, User Settable <SSEN = 1>, Module Control	Yes
UART (x = 1 or 2)	UxRX	Module Control	Yes
	UxTX	<UTXEN = 0>, User Settable <UTXEN = 1>, Module Control	Yes
I²C™	SCL	Module Control	Yes
	SDA	Module Control	Yes
Input Change Notice	CN0 - CN23	User Settable	Yes
Input Capture	IC1 - IC8	User Settable	Yes
Output Compare	OC1 - OC8	Module Control	Yes
Data Converter Interface	COFS	Module Control	Yes
	CSCK	Module Control	Yes
	CSDI	Module Control	Yes
	CSDO	Module Control	Yes
Motor Control PWM	PWMx	Module Control	Yes
	FLTA/B	User Settable	Yes
QEI	QEA	Module Control (QEI mode) User Settable (16-bit Timer mode)	Yes
	QEB	Module Control (QEI mode) User Settable (16-bit Timer mode)	Yes
	INDX	Module Control (QEIM<2:0> = 100 or 110) User Settable in all other modes	Yes
CAN (x = 1 or 2)	CxRX	Module Control	Yes
	CxTX	Module Control	Yes
INTx	INT0 - INT5	User Settable	Yes

## 11.4 Port Descriptions

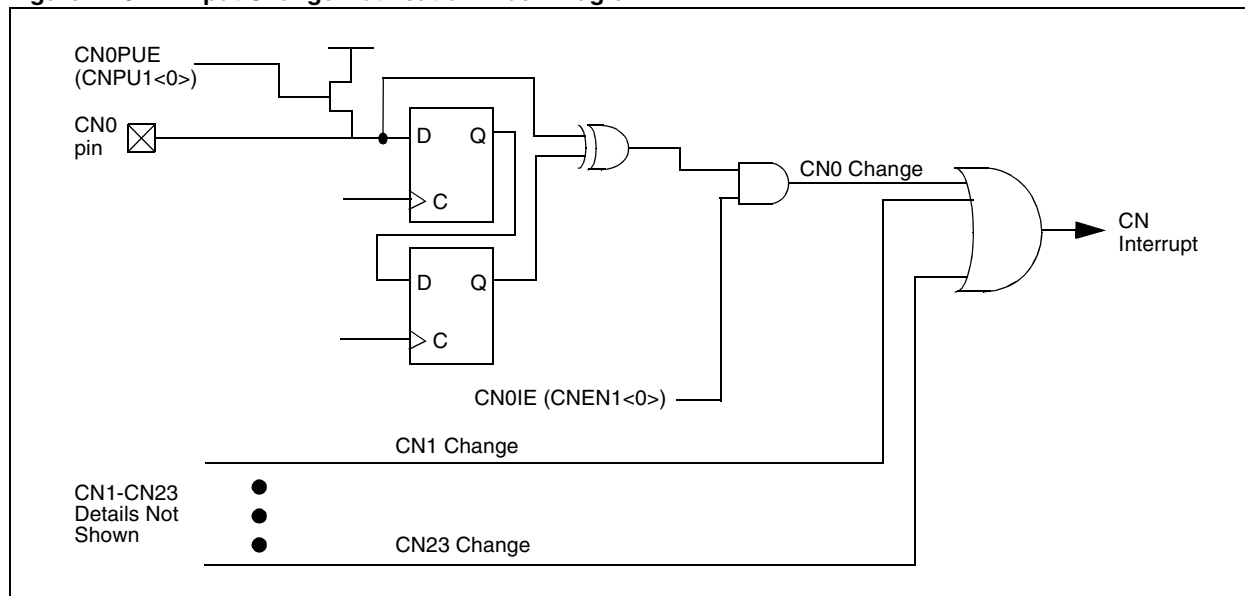
Refer to the device data sheet for a description of the available I/O ports and peripheral multiplexing details.

### 11.5 Change Notification (CN) Pins

The Change Notification (CN) pins provide dsPIC30F devices the ability to generate interrupt requests to the processor in response to a change of state on selected input pins. Up to 24 input pins may be selected (enabled) for generating CN interrupts. The total number of available CN inputs is dependent on the selected dsPIC30F device. Refer to the device data sheet for further details.

Figure 11-3 shows the basic function of the CN hardware.

**Figure 11-3: Input Change Notification Block Diagram**



#### 11.5.1 CN Control Registers

There are four control registers associated with the CN module. The CNEN1 and CNEN2 registers contain the CNxIE control bits, where 'x' denotes the number of the CN input pin. The CNxIE bit must be set for a CN input pin to interrupt the CPU.

The CNPU1 and CNPU2 registers contain the CNxPUE control bits. Each CN pin has a weak pull-up device connected to the pin, which can be enabled or disabled using the CNxPUE control bits. The weak pull-up devices act as a current source that is connected to the pin and eliminate the need for external resistors when push button or keypad devices are connected. Refer to the "Electrical Specifications" section of the device data sheet for CN pull-up device current specifications.

## 11.5.2 CN Configuration and Operation

The CN pins are configured as follows:

1. Ensure that the CN pin is configured as a digital input by setting the associated bit in the TRISx register.
2. Enable interrupts for the selected CN pins by setting the appropriate bits in the CNEN1 and CNEN2 registers.
3. Turn on the weak pull-up devices (if desired) for the selected CN pins by setting the appropriate bits in the CNPU1 and CNPU2 registers.
4. Clear the CNIF (IFS0<15>) interrupt flag.
5. Select the desired interrupt priority for CN interrupts using the CNIP<2:0> control bits (IPC3<14:12>).
6. Enable CN interrupts using the CNIE (IEC0<15>) control bit.

When a CN interrupt occurs, the user should read the PORT register associated with the CN pin(s). This will clear the mismatch condition and setup the CN logic to detect the next pin change. The current PORT value can be compared to the PORT read value obtained at the last CN interrupt to determine the pin that changed.

The CN pins have a minimum input pulse width specification. Refer to the “Electrical Specifications” section of the device data sheet for further details.

## 11.6 CN Operation in Sleep and Idle Modes

The CN module continues to operate during Sleep or Idle modes. If one of the enabled CN pins changes states, the CNIF (IFS0<15>) status bit will be set. If the CNIE bit (IEC0<15>) is set, the device will wake from Sleep or Idle mode and resume operation.

If the assigned priority level of the CN interrupt is equal to or less than the current CPU priority level, device execution will continue from the instruction immediately following the `SLEEP` or `IDLE` instruction.

If the assigned priority level of the CN interrupt is greater than the current CPU priority level, device execution will continue from the CN interrupt vector address.



**Register 11-1: CNEN1: Input Change Notification Interrupt Enable Register1**

Upper Byte:							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CN15IE	CN14IE	CN13IE	CN12IE	CN11IE	CN10IE	CN9IE	CN8IE
bit 15							bit 8

Lower Byte:							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CN7IE	CN6IE	CN5IE	CN4IE	CN3IE	CN2IE	CN1IE	CN0IE
bit 7							bit 0

bit 15-0 **CNxIE**: Input Change Notification Interrupt Enable bits

1 = Enable interrupt on input change

0 = Disable interrupt on input change

**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

**Register 11-2: CNEN2: Input Change Notification Interrupt Enable Register2**

Upper Byte:							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	—	—	—
bit 15							bit 8

Lower Byte:							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CN23IE	CN22IE	CN21IE	CN20IE	CN19IE	CN18IE	CN17IE	CN16IE
bit 7							bit 0

bit 15-8 **Unimplemented**: Read as '0'

bit 7-0 **CNxIE**: Input Change Notification Interrupt Enable bits

1 = Enable interrupt on input change

0 = Disable interrupt on input change

**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

# dsPIC30F Family Reference Manual

**Register 11-3: CNPU1: Input Change Notification Pull-up Enable Register1**

<b>Upper Byte:</b>							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CN15PUE	CN14PUE	CN13PUE	CN12PUE	CN11PUE	CN10PUE	CN9PUE	CN8PUE
bit 15							
bit 8							

<b>Lower Byte:</b>							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CN7PUE	CN6PUE	CN5PUE	CN4PUE	CN3PUE	CN2PUE	CN1PUE	CN0PUE
bit 7							
bit 0							

bit 15-0 **CNxPUE:** Input Change Notification Pull-up Enable bits

1 = Enable pull-up on input change

0 = Disable pull-up on input change

**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

**Register 11-4: CNPU2: Input Change Notification Pull-up Enable Register2**

<b>Upper Byte:</b>							
U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							
bit 8							

<b>Lower Byte:</b>							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CN23PUE	CN22PUE	CN21PUE	CN20PUE	CN19PUE	CN18PUE	CN17PUE	CN16PUE
bit 7							
bit 0							

bit 15-8 **Unimplemented:** Read as '0'

bit 7-0 **CNxPUE:** Input Change Notification Pull-up Enable bits

1 = Enable pull-up on input change

0 = Disable pull-up on input change

**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

11.7      **Related Application Notes**

This section lists application notes that are related to this section of the manual. These application notes may not be written specifically for the dsPIC30F Product Family, but the concepts are pertinent and could be used with modification and possible limitations. The current application notes related to the I/O Ports module are:

Title	Application Note #
Implementing Wake-up on Key Stroke	AN552

**Note:** Please visit the Microchip web site ([www.microchip.com](http://www.microchip.com)) for additional Application Notes and code examples for the dsPIC30F Family of devices.

## 11.8 Revision History

### **Revision A**

This is the initial released revision of this document.

### **Revision B**

This revision incorporates additional technical content for the dsPIC30F I/O Ports module.

### **Revision C**

There were no technical content revisions to this section of the manual, however, this section was updated to reflect Revision C throughout the manual.

### **Revision D**

**Section 11.3.1.2 “Pin Control Summary”** was added to this revision.