I/O Ports with Edge Detect

HIGHLIGHTS

This section of the manual contains the following topics:

Note: This family reference manual section is meant to serve as a complement to device data sheets. Some dsPIC33/PIC24 devices are dual core and contain both a Master and Slave CPU core. For single core dsPIC33/PIC24 devices, disregard any Slave-specific references. Depending on the device variant, this section may not apply to all dsPIC33/PIC24 devices.

Please consult the note at the beginning of the **"I/O Ports"** chapter in the current device data sheet to check whether this document supports the device you are using.

Device data sheets and family reference manual sections are available for download from the Microchip Worldwide Web site at: http://www.microchip.com.

1.0 INTRODUCTION

The general purpose I/O pins can be considered the simplest of peripherals. These I/O pins allow the dsPIC33/PIC24 microcontrollers to monitor and control other devices. To add flexibility and functionality to a device, some pins are multiplexed with alternate function(s). These functions depend on which peripheral features are on the device. In general, when a peripheral is functioning, that pin may not be used as a general purpose I/O pin.

The Master and Slave core output functionality of the ports is defined by device-specific Configuration registers, FCFGPRA0 to FCFGPRE0. When these Configuration bits are maintained as '1', the Master owns the pin (only the output function); when the bits are '0', the ownership of that specific pin belongs to the Slave. The input function of the I/O is valid for both Master and Slave cores. The Configuration registers, FCFGPRA0 to FCFGPRE0, do not have any control over the input function.

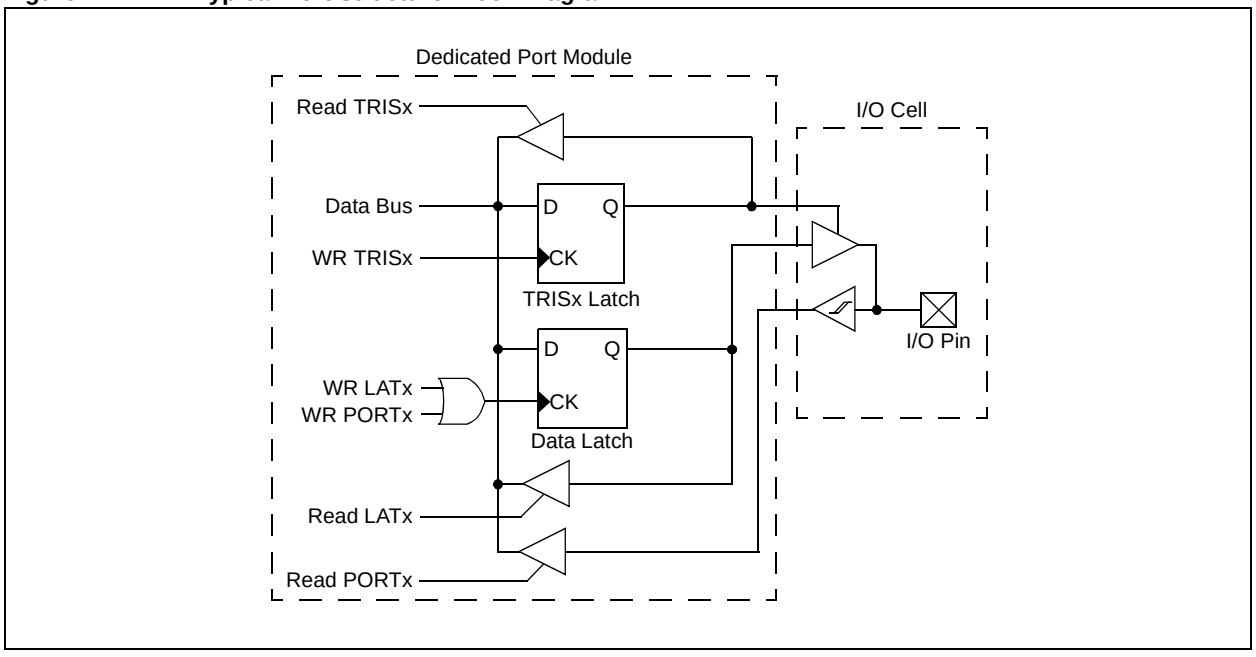
Please refer to the specific device data sheet for more information.

Following are some of the key features of the I/O Ports with Edge Detect module:

- Individual output pin open-drain enable/disable
- Individual input pin pull-up enable/disable
- Monitor select inputs and generate interrupt on a mismatch condition
- Operate during Sleep and Idle modes

A block diagram of a typical I/O port structure is shown in Figure 1-1. This block diagram does not consider peripheral functions that might be multiplexed onto the I/O pin.

Figure 1-1: Typical Port Structure Block Diagram



2.0 CONTROL REGISTERS

Note:	Not all registers and associated bits are available on all devices. Refer to the **"I/O Ports"** chapter in the specific device data sheet to determine availability.

Before reading and writing any I/O port, the desired pin or pins should be properly configured for the application. Each I/O port has nine registers directly associated with the operation of the port and one control register. Each I/O port pin has a corresponding bit in these registers. Throughout this section, the letter 'x' denotes any or all port module instances. For example, TRISx would represent TRISA, TRISB, TRISC and so on. Any bit and its associated data and control registers that are not valid for a particular device will be disabled and will read as zeros.

For additional information on the registers in this section, refer to the **"I/O Ports"** chapter in the specific device data sheet.

2.1 Registers for Configuring Tri-State Functions (TRISx)

The TRISx registers configure the data direction flow through port I/O pins. The TRISx register bits determine whether a PORTx I/O pin is an input or an output:

- If a data direction bit is '1', the corresponding I/O port pin is an input.
- If a data direction bit is '0', the corresponding I/O port pin is an output.
- A read from a TRISx register reads the last value written to that register.
- All I/O port pins are defined as inputs after a Power-on Reset (POR).

Note:	It is recommended to make the pin an output and drive to zero (TRISx = 0,LATx = 0) prior to making the I/O pin an input (TRISx = 1); this will help in discharging the parasitic capacitance internal to the I/O pin.

2.2 Registers for Configuring Port Functions (PORTx)

The PORTx registers allow I/O pins to be accessed:

- A write to a PORTx register writes to the corresponding LATx register (PORTx data latch). Those I/O port pin(s) configured as outputs are updated.
- A write to a PORTx register is effectively the same as a write to a LATx register.
- A read from a PORTx register reads the synchronized signal applied to the port I/O pins.

2.3 Registers for Configuring Latch Functions (LATx)

The LATx registers (PORTx Data Latch) hold data written to port I/O pins:

- A write to a LATx register latches data to corresponding port I/O pins. Those I/O port pins configured as outputs are updated.
- A read from a LATx register reads the data held in the PORTx data latch, not from the port I/O pins.

2.4 Registers for Open-Drain Configuration (ODCx)

Each I/O pin can be individually configured for either normal digital output or open-drain output. This is controlled by the Open-Drain Control register, ODCx, associated with each I/O pin. If the ODCx bit for an I/O pin is a '1', the pin acts as an open-drain output. If the ODCx bit for an I/O pin is a '0', the pin is configured for a normal digital output (the ODCx bit is valid only for output pins). After a Reset, the status of all the bits of the ODCx register is set to '0'.

The open-drain feature allows the generation of outputs higher than V_{DD} (for example, 5V) on any desired 5V tolerant pins by using external pull-up resistors. The maximum open-drain voltage allowed is the same as the maximum V_{IH} specification. The ODCx register setting takes effect in all of the I/O modes, allowing the output to behave as an open-drain, even if a peripheral is controlling the pin. Although the user could achieve the same effect by manipulating the corresponding LATx and TRISx bits, this procedure will not allow the peripheral to operate in Open-Drain mode (except for the default operation of the I^2C pins). Since I^2C pins are already

open-drain pins, the ODCx settings do not affect the I^2C pins. Also, the ODCx settings do not affect the JTAG output characteristics as the JTAG scan cells are inserted between the ODCx logic and the I/O.

2.5 Registers for Configuring Analog and Digital Port Pins (ANSELx)

The ANSELx register controls the operation of the analog port pins. The port pins that are to function as analog inputs must have their corresponding ANSELx and TRISx bits set. To use port pins for I/O functionality with digital modules, such as Timers, UARTs, etc., the corresponding ANSELx bit must be cleared.

The ANSELx register has a default value of 0xFFFF; therefore, all pins that share analog functions are, by default, analog and *not* digital.

If the TRISx bit is cleared (output) while the ANSELx bit is set, the digital output level (V_{OH} or V_{OL}) is converted by an analog peripheral, such as the ADC module or the comparator module.

When the PORTx register is read, all pins configured as analog input channels are read as cleared (a low level).

Pins configured as digital inputs do not convert an analog input. Analog levels on any pin defined as a digital input (including the ANx pins) can cause the input buffer to consume current that exceeds the device specifications.

2.6 Registers for Input Change Notification (CN)

The input Change Notification (CN) function of the I/O ports allows dsPIC33/PIC24 devices to generate interrupt requests to the processor in response to a Change-of-State (COS) on selected input pins. This feature can detect input Change-of-States, even in Sleep mode, when the clocks are disabled.

The following control registers are associated with the CN functionality of each I/O port:

- Change Notice Pull-up Enable (CNPUx)
- Change Notice Pull-Down Enable (CNPDx)
- Change Notice Control (CNCONx) (Register 2-1)
- Change Notice Enable/Negative Edge Control (CNEN0x)
- Change Notice Positive Edge Control (CNEN1x)
- Change Notice Status Mismatch Event (CNSTATx)
- Change Notice Status Enabled Edge Event (CNFx)

Each I/O pin also has a weak pull-up and a weak pull-down connected to it. The pull-ups act as a current source or sink source connected to the pin, and eliminate the need for external resistors when push button or keypad devices are connected. The pull-ups and pull-downs are enabled separately using the CNPUx and the CNPDx registers, which contain the control bits for each of the pins. Setting any of the control bits enables the weak pull-ups and/or pull-downs for the corresponding pins.

Note:	Pull-ups and pull-downs on Change Notification pins should always be disabled when the port pin is configured as a digital output.

The CNCONx registers provide change notice control. The CNSTYLE bit will select either a standard Mismatch mode style or a configurable Edge Detect mode.

When CNSTLYE is cleared (Mismatch mode), the CNEN0x registers contain the CN interrupt enable control bits for each of the input pins. Setting any of these bits enables a CN interrupt for the corresponding pins. CNFx bits are used as Mismatch mode event status flags.

When CNSTYLE is set (Edge Detect mode), CNEN0x registers are used to enable/disable negative edge events and CNEN1x is used to enable/disable positive events for a given pin. CNSTATx bits are used as edge detect event status flags.

On devices that do not have a CNSTYLE bit, Edge Detect mode is not available and the CN logic will function in standard Mismatch mode.

Register 2-1: CNCONx: Change Notice Control x Register

R/W-0	U-0	U-0	U-0	R/W-0	U-0	U-0	U-0
ON	—	—	—	CNSTYLE	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
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 15 **ON:** Change Notice (CN) Control On bit
1 = CN is enabled
0 = CN is disabled

bit 14-12 **Unimplemented:** Read as ‘0’

bit 11 **CNSTYLE:** CN Edge Detection Type Control bit
1 = Edge style (detects edge transitions configured in CNEN0x/CNEN1x, CNFx<15:0> bits are used for a Change Notification event)
0 = Mismatch style (detects change from last port read, CNSTATx<15:0> bits are used for a Change Notification event)

bit 10-0 **Unimplemented:** Read as ‘0’

2.7 Registers for Peripheral Pin Select (PPS)

The Peripheral Pin Select Input registers, RPINRx, and the Peripheral Pin Select Output registers, RPORx, provide control for PPS input and output mapping. See **Section 3.3.1 “Input Mapping”** and **Section 3.3.2 “Output Mapping”** for detailed information on configuring these registers.

Note: dsPIC33/PIC24 devices may have a varied number of RPINRx and RPORx registers. For more information, refer to the specific device data sheet.

3.0 PERIPHERAL PIN SELECT (PPS)

A major challenge in general purpose devices is providing the largest possible set of peripheral features, while minimizing the conflict of features on the I/O pins. The challenge is even greater on low pin count devices. In an application where more than one peripheral needs to be assigned to a single pin, inconvenient work arounds in application code, or a complete redesign, may be the only option.

Peripheral Pin Select (PPS) configuration provides an alternative to these choices by enabling peripheral set selection and their placement on a wide range of I/O pins. By increasing the pinout options available on a particular device, users can better tailor the device to their entire application, rather than trimming the application to fit the device.

The Peripheral Pin Select configuration feature operates over a fixed subset of digital I/O pins. Users may independently map the input and/or output of most digital peripherals to any one of these I/O pins. Peripheral Pin Select is performed in software and generally does not require the device to be reprogrammed. Hardware safeguards are included that prevent accidental or spurious changes to the peripheral mapping once it has been established.

3.1 Available Pins

The number of available pins is dependent on the particular device and its pin count. Pins that support the Peripheral Pin Select feature include the designation, “RPn”, in their full pin designation, where “RP” designates a remappable peripheral and “n” is the remappable pin number. Note that on a device’s schematic symbol, remappable input pins are designated as RPIn and remappable output pins are designated as RPn.

Note:	Some “RPn” pins are not available for output functionality. Refer to the specific device data sheet for the available RPn pins and their functionality.

3.2 Available Peripherals

The peripherals managed by the Peripheral Pin Select are all digital only peripherals. These include general serial communications (UART and SPI), general purpose timer clock inputs, timer-related peripherals (input capture and output compare) and interrupt-on-change inputs.

In comparison, some digital only peripheral modules are never included in the Peripheral Pin Select feature. This is because the peripheral’s function requires special I/O circuitry on a specific port and cannot be easily connected to multiple pins. These modules include I^2C and the motor control PWM. A similar requirement excludes all modules with analog inputs, such as an A/D Converter.

Note:	For a specific list of Peripheral Pin Select supported peripherals, refer to the device data sheet.

A key difference between remappable and non-remappable peripherals is that remappable peripherals are not associated with a default I/O pin. The peripheral must always be assigned to a specific I/O pin before it can be used. In contrast, non-remappable peripherals are always available on a default pin, assuming that the peripheral is active and not conflicting with another peripheral.

When a remappable peripheral is active on a given I/O pin, it takes priority over all other digital I/Os and digital communication peripherals associated with the pin. Priority is given regardless of the type of peripheral that is mapped. Remappable peripherals never take priority over any analog functions associated with the pin.

3.3 Controlling Peripheral Pin Select

Peripheral Pin Select features are controlled through two sets of SFRs: one to map peripheral inputs and another one to map outputs. Because they are separately controlled, a particular peripheral’s input and output (if the peripheral has both) can be placed on any selectable function pin without constraint.

The association of a peripheral to a peripheral-selectable pin is handled in two different ways, depending on whether an input or output is being mapped.

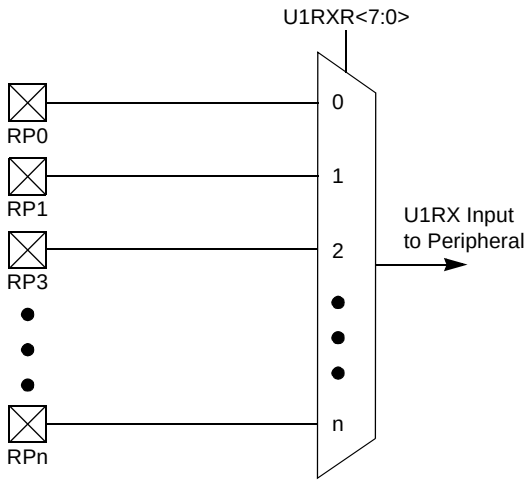
3.3.1 INPUT MAPPING

The inputs of the Peripheral Pin Select options are mapped on the basis of the peripheral; that is, a control register associated with a peripheral dictates the pin it will be mapped to. The RPINRx registers are used to configure peripheral input mapping (refer to the specific device data sheet for register details). Each register contains sets of 8-bit fields, with each set associated with one of the remappable peripherals. Programming a given peripheral's bit field with an appropriate 8-bit value maps the RPn pin with the corresponding value to that peripheral. For any given device, the valid range of values for any bit field corresponds to the maximum number of Peripheral Pin Selections supported by the device.

Note:	Not all RPn pins may be available on all devices. Refer to the specific device data sheet for the available RPn pins.

For example, Figure 3-1 illustrates remappable pin selection for the U1RX input.

Figure 3-1: Remappable Input for U1RX



Note: For input only, Peripheral Pin Select functionality does not have priority over TRISx settings. Therefore, when configuring the RPn pin for an input, the corresponding bit in the TRISx register must be configured for input (set to '1').

Table 3-1: Selectable Input Sources (Maps Input to Function)

Input Name[1]	Function Name	Register	Configuration Bits
External Interrupt 1	INT1	RPINR0	INT1R<7:0>
External Interrupt 2	INT2	RPINR1	INT2R<7:0>
External Interrupt 3	INT3	RPINR1	INT3R<7:0>
Timer1 External Clock	T1CK	RPINR2	T1CK<7:0>
SCCP Timer1	TCKI1	RPINR3	TCKI1R<7:0>
SCCP Capture 1	ICM1	RPINR3	ICM1R<7:0>
SCCP Timer2	TCKI2	RPINR4	TCKI2R<7:0>
SCCP Capture 2	ICM2	RPINR4	ICM2R<7:0>
SCCP Timer3	TCKI3	RPINR5	TCKI3R<7:0>
SCCP Capture 3	ICM3	RPINR5	ICM3R<7:0>
SCCP Timer4	TCKI4	RPINR6	TCKI4R<7:0>
SCCP Capture 4	ICM4	RPINR6	ICM4R<7:0>
SCCP Timer5	TCKI5	RPINR7	TCKI5R<7:0>
SCCP Capture 5	ICM5	RPINR7	ICM5R<7:0>
SCCP Timer6	TCKI6	RPINR8	TCKI6R<7:0>
SCCP Capture 6	ICM6	RPINR8	ICM6R<7:0>
SCCP Timer7	TCKI7	RPINR9	TCKI7R<7:0>
SCCP Capture 7	ICM7	RPINR9	ICM7R<7:0>
SCCP Timer8	TCKI8	RPINR10	TCKI8R<7:0>
SCCP Capture 8	ICM8	RPINR10	ICM8R<7:0>
SCCP Fault A	OCFA	RPINR11	OCFAR<7:0>
SCCP Fault B	OCFB	RPINR11	OCFBR<7:0>
PWM Input 8	PCI8	RPINR12	PCI8R<7:0>
PWM Input 9	PCI9	RPINR12	PCI9R<7:0>
PWM Input 10	PCI10	RPINR13	PCI10R<7:0>
PWM Input 11	PCI11	RPINR13	PCI11R<7:0>
QEI Input A	QEIA1	RPINR14	QEIA1R<7:0>
QEI Input B	QEIB1	RPINR14	QEIB1R<7:0>
QEI Index 1 Input	QEINDX1	RPINR15	QEINDX1R<7:0>
QEI Home 1 Input	QEIHOM1	RPINR15	QEIHOM1R<7:0>
UART1 Receive	U1RX	RPINR18	U1RXR<7:0>
UART1 Data-Set-Ready	$\overline{\text{U1DSR}}$	RPINR18	U1DSRR<7:0>
UART2 Receive	U2RX	RPINR19	U2RXR<7:0>
UART2 Data-Set-Ready	$\overline{\text{U2DSR}}$	RPINR19	U2DSRR<7:0>
SPI1 Data Input	SDI1	RPINR20	SDI1R<7:0>
SPI1 Clock Input	SCK1IN	RPINR20	SCK1R<7:0>
SPI1 Slave Select	$\overline{\text{SS1}}$	RPINR21	SS1R<7:0>
Reference Clock Input	REFOI	RPINR21	REFOIR<7:0>
SPI2 Data Input	SDI2	RPINR22	SDI2R<7:0>
SPI2 Clock Input	SCK2IN	RPINR22	SCK2R<7:0>
SPI2 Slave Select	$\overline{\text{SS2}}$	RPINR23	SS2R<7:0>
UART1 Clear-to-Send	$\overline{\text{U1CTS}}$	RPINR23	U1CTSR<7:0>
CAN1 Input	CAN1RX	RPINR26	CAN1RXR<7:0>

Note 1: Unless otherwise noted, all inputs use the Schmitt Trigger input buffers.

Table 3-1: Selectable Input Sources (Maps Input to Function) (Continued)

Input Name[1]	Function Name	Register	Configuration Bits
UART2 Clear-to-Send	$\overline{\text{U2CTS}}$	RPINR30	U2CTSR<7:0>
PWM Input 17	PCI17	RPINR37	PCI17R<7:0>
PWM Input 18	PCI18	RPINR38	PCI18R<7:0>
PWM Input 12	PCI12	RPINR42	PCI12R<7:0>
PWM Input 13	PCI13	RPINR42	PCI13R<7:0>
PWM Input 14	PCI14	RPINR43	PCI14R<7:0>
PWM Input 15	PCI15	RPINR43	PCI15R<7:0>
PWM Input 16	PCI16	RPINR44	PCI16R<7:0>
SENT1 Input	SENT1	RPINR44	SENT1R<7:0>
SENT2 Input	SENT2	RPINR45	SENT2R<7:0>
CLC Input A	CLCINA	RPINR45	CLCINAR<7:0>
CLC Input B	CLCINB	RPINR46	CLCINBR<7:0>
CLC Input C	CLCINC	RPINR46	CLCINCR<7:0>
CLC Input D	CLCIND	RPINR47	CLCINDR<7:0>
ADC Trigger Input (ADTRIG31)	ADCTRG	RPINR47	ADCTRGR<7:0>

Note 1: Unless otherwise noted, all inputs use the Schmitt Trigger input buffers.

Note: Table 3-1 and Figure 3-1 provide examples of selectable input sources for a generic device. For more information, refer to the specific device data sheet.

3.3.2 OUTPUT MAPPING

In contrast to inputs, the outputs of the PPS options are mapped on the basis of the pin. In this case, a control register associated with a particular pin dictates the peripheral output to be mapped. The RPORx registers are used to control output mapping. Each register contains sets of six-bit fields, with each set associated with one RPn pin (refer to the specific device data sheet for register details). The value of the bit field corresponds to one of the peripherals and that peripheral's output is mapped to the pin (see Table 3-2 and Figure 3-2).

A null output is associated with the Output Register Reset value of '0'. This is done to ensure that, by default, remappable outputs remain disconnected from all output pins.

Figure 3-2: Multiplexing of Remappable Outputs for RPn

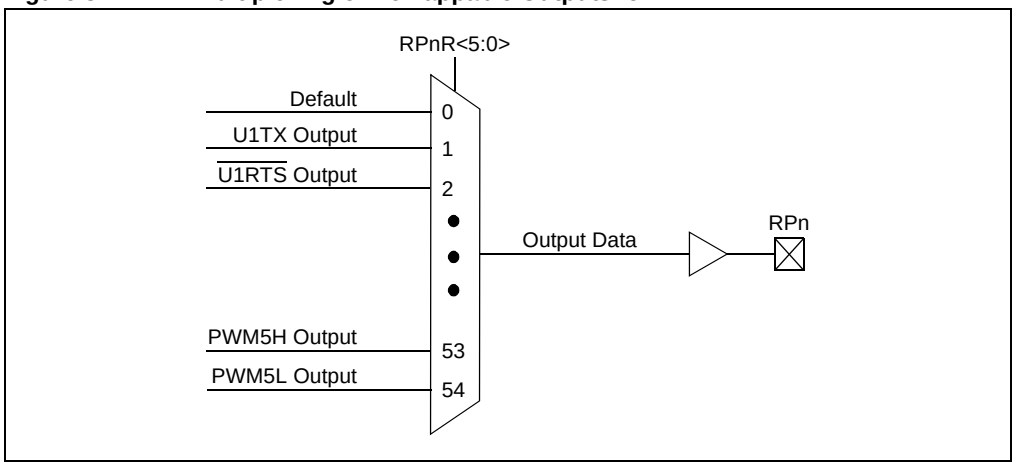


Table 3-2: Output Selection for Remappable Pins (RPn)

Function	RPnR<5:0>	Output Name
Default PORT	000000	RPn tied to Default Pin
U1TX	000001	RPn tied to UART1 Transmit
$\overline{\text{U1RTS}}$	000010	RPn tied to UART1 Request-to-Send
U2TX	000011	RPn tied to UART2 Transmit
$\overline{\text{U2RTS}}$	000100	RPn tied to UART2 Request-to-Send
SDO1	000101	RPn tied to SPI1 Data Output
SCK1	000110	RPn tied to SPI1 Clock Output
$\overline{\text{SS1}}$	000111	RPn tied to SPI1 Slave Select
SDO2	001000	RPn tied to SPI2 Data Output
SCK2	001001	RPn tied to SPI2 Clock Output
$\overline{\text{SS2}}$	001010	RPn tied to SPI2 Slave Select
REFCLKO	001110	RPn tied to Reference Clock Output
OCM1	001111	RPn tied to SCCP1 Output
OCM2	010000	RPn tied to SCCP2 Output
OCM3	010001	RPn tied to SCCP3 Output
OCM4	010010	RPn tied to SCCP4 Output
OCM5	010011	RPn tied to SCCP5 Output
OCM6	010100	RPn tied to SCCP6 Output
CAN1	010101	RPn tied to CAN1 Output
CMP1	010111	RPn tied to Comparator 1 Output
PWM4H	100010	RPn tied to PWM4H Output

Table 3-2: Output Selection for Remappable Pins (RPn) (Continued)

Function	RPnR<5:0>	Output Name
PWM4L	100011	RPn tied to PWM4L Output
PWMEA	100100	RPn tied to PWM Event A Output
PWMEB	100101	RPn tied to PWM Event B Output
QEICMP	100110	RPn tied to QEI Comparator Output
CLC1OUT	101000	RPn tied to CLC1 Output
CLC2OUT	101001	RPn tied to CLC2 Output
OCM7	101010	RPn tied to SCCP7 Output
OCM8	101011	RPn tied to SCCP8 Output
PWMEC	101100	RPn tied to PWM Event C Output
PWMED	101101	RPn tied to PWM Event D Output
PTGTRG24	101110	RPn tied to PTG Trigger Output 24
PTGTRG25	101111	RPn tied to PTG Trigger Output 25
SENT1OUT	110000	RPn tied to SENT1 Output
SENT2OUT	110001	RPn tied to SENT2 Output
CLC3OUT	110010	RPn tied to CLC3 Output
CLC4OUT	110011	RPn tied to CLC4 Output
U1DTR	110100	RPn tied to Data Terminal Ready Output 1
U2DTR	110101	RPn tied to Data Terminal Ready Output 2

Note: Figure 3-2 and Table 3-2 provide examples of a generic device. For more information, refer to the specific device data sheet.

3.3.3 MAPPING LIMITATIONS

The control scheme of the peripheral select pins is not limited to a small range of fixed peripheral configurations. There are no mutual or hardware enforced lockouts between any of the peripheral mapping SFRs. Literally, any combination of peripheral mappings across any or all of the RPn pins is possible. This includes both many-to-one and one-to-many mappings of peripheral inputs and outputs to pins. While such mappings may be technically possible from a configuration point of view, they may not be supportable from an electrical point of view.

3.4 Controlling Configuration Changes

Because peripheral mapping can be changed during run time, some restrictions on peripheral remapping are needed to prevent accidental configuration changes. All dsPIC33/PIC24 devices include a control register lock bit to prevent alterations to the peripheral map.

3.4.1 CONTROL REGISTER LOCK

Under normal operation, writes to the RPINRx and RPORx registers are not allowed. Attempted writes will appear to execute normally, but the contents of the registers will remain unchanged. To change these registers, they must be unlocked in hardware. The register lock is controlled by the IOLOCK bit (RPCON<11>). Setting IOLOCK prevents writes to the control registers; clearing IOLOCK allows writes.

To set or clear IOLOCK, the NVMKEY unlock sequence must be executed:

1. Write 0x55 to NVMKEY.
2. Write 0xAA to NVMKEY.
3. Clear (or set) IOLOCK as a single operation.

IOLOCK remains in one state until changed. This allows all of the Peripheral Pin Selects to be configured with a single unlock sequence, followed by an update to all of the control registers. Then, IOLOCK can be set with a second lock sequence.

Note: MPLAB® C30 provides a built-in C language function for unlocking and modifying the RPCON register:

```
__builtin_write_RPCON(value);
```

For more information, see the MPLAB C30 Help files.

3.5 Considerations for Peripheral Pin Selection

The ability to control Peripheral Pin Selection introduces several considerations into application design that most users would never think of otherwise. This is particularly true for several common peripherals, which are only available as remappable peripherals.

The main consideration is that the Peripheral Pin Selects are not available on default pins in the device's default (Reset) state. More specifically, because all RPINRx registers reset to '1's and RPORx registers reset to '0's, this means all PPS inputs are tied to VSS, while all PPS outputs are disconnected. This means that before any other application code is executed, the user must initialize the device with the proper peripheral configuration. Because the IOLOCK bit resets in the unlocked state, it is not necessary to execute the unlock sequence after the device has come out of Reset. For application safety, however, it is always better to set IOLOCK and lock the configuration after writing to the control registers.

The NVMKEY unlock sequence must be executed as an assembly language routine. If the bulk of the application is written in 'C', or another high-level language, the unlock sequence should be performed by writing in-line assembly or by using the `__builtin_write_RPCON(value)` function provided by the compiler.

Choosing the configuration requires a review of all Peripheral Pin Selects and their pin assignments, particularly those that will not be used in the application. In all cases, unused pin-selectable peripherals should be disabled completely. Unused peripherals should have their inputs assigned to an unused RPn pin function. I/O pins with unused RPn functions should be configured with the null peripheral output.

The assignment of a peripheral to a pin does not perform any other configuration of the pin's I/O circuitry. This means that adding a pin-selectable output to a pin may mean inadvertently driving an existing peripheral input when the output is driven. Users must be familiar with the behavior of other fixed peripherals that share a remappable pin and know when to enable or disable them. To be safe, fixed digital peripherals that share the same pin should be disabled when not in use.

Along these lines, configuring a remappable pin for a specific peripheral does not automatically turn that feature on. The peripheral must be specifically configured for operation and enabled as if it were tied to a fixed pin. Where this happens in the application code (immediately following device Reset and peripheral configuration or inside the main application routine) depends on the peripheral and its use in the application.

A final consideration is that Peripheral Pin Select functions neither override analog inputs nor reconfigure pins with analog functions for digital I/Os. If a pin is configured as an analog input on device Reset, it must be explicitly reconfigured as a digital I/O when used with a Peripheral Pin Select.

Example 3-1 provides a configuration for bidirectional communication with flow control using UART1. The following input and output functions are used:

- Input Functions: U1RX, $\overline{\text{U1CTS}}$
- Output Functions: U1TX, $\overline{\text{U1RTS}}$

Example 3-1: Configuring UART1 Input and Output Functions

```
//************************************************************
// Unlock Registers
//************************************************************
__builtin_write_RPCON(0x0000);

//************************************************************
// Configure Input Functions
// (See Table 3-1)
//************************************************************
//******************************
// Assign U1Rx To Pin RP35
//******************************
_U1RXR = 35;

//******************************
// Assign U1CTS To Pin RP36
//******************************
_U1CTSR = 36;

//************************************************************
// Configure Output Functions
// (See Table 3-2)
//************************************************************
//******************************
// Assign U1Tx To Pin RP37
//******************************
_RP37 = 1;

//******************************
// Assign U1RTS To Pin RP38
//******************************
_RP38 = 2;

//************************************************************
// Lock Registers
//************************************************************
__builtin_write_RPCON(0x0800);
```

3.6 Virtual Output Pins

The virtual pins enable the user to connect internal peripherals, whose signals may be of significant use to other peripherals, but these outputs may not need to be presented to a device pin.

The concept of “virtual pins” enables new device features to be added to a device that were not initially conceived during the design of the original peripherals. One common use for the virtual pins is to connect the analog comparator, or output compare outputs, to the PWM module for use as current-limit or Fault input signals. Refer to the specific device data sheet for more information.

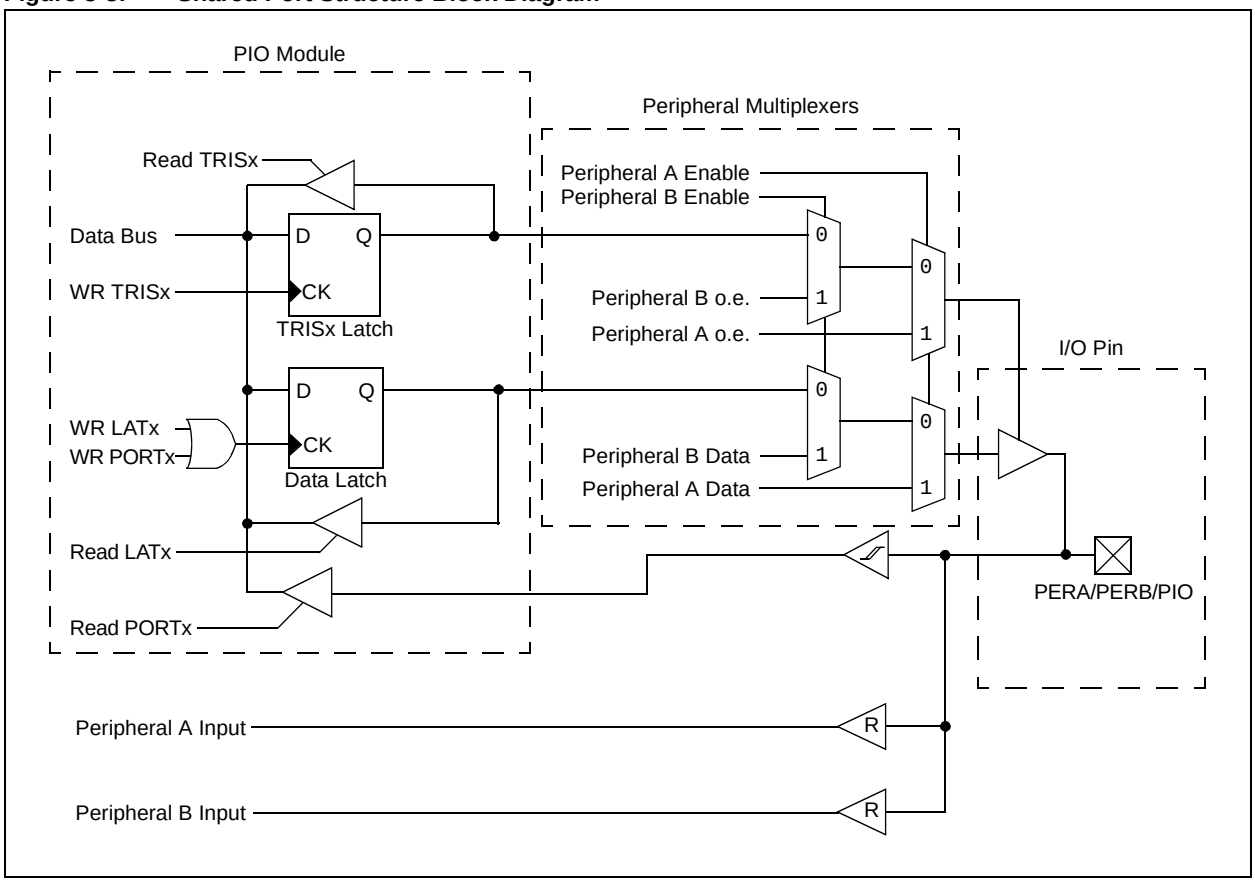
3.7 Peripheral Multiplexing

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 3-3 illustrates how ports are shared with other peripherals and the associated I/O pin to which they are connected.

Note:	Some ports are shared with analog module pins. The corresponding bits in the ANSELx registers, if present, must be set to '0' for I/O port functionality.

Figure 3-3: Shared Port Structure Block Diagram



3.8 I/O Multiplexing with Multiple Peripherals

For some dsPIC33/PIC24 devices, particularly those with a small number of I/O pins, multiple peripheral functions may be multiplexed on each I/O pin. Figure 3-3 illustrates 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, illustrated in Figure 3-3, has two multiplexed peripherals, Peripheral A and Peripheral B, and is named as PERA/PERB/PIO.

The I/O pin name is chosen because the user-assigned application can easily determine the priority of the functions assigned to the pin. As shown in Figure 3-3, 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 pin.

3.8.1 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 TRISx control bit, the user can manually affect the state of the input capture pin through its corresponding PORTx 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 3-3, the organization of the peripheral multiplexers will determine if the peripheral input pin can be manipulated in software using the PORTx register. The conceptual peripherals shown in Figure 3-3 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 PORTx registers:

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

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

- SPI
- I^2C
- DCI
- UART
- CAN FD
- QEI

Note:	Some peripherals may not be present on all device variants. For more information, refer to the specific device data sheet.

3.8.2 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 in software through the associated TRISx Special Function Register (SFR). The TRISx register must be set for the peripheral to function properly. For “user-settable” peripheral pins, the actual port pin state can always be read through the PORTx SFR.

An input capture peripheral provides an example of a user-settable peripheral. The user application must write the associated TRISx register to configure the input capture pin as an input. Because the I/O pin circuitry is still active when the input capture is enabled, the following method can be used to manually produce capture events using software:

- The input capture pin is configured as an output using the associated TRISx register.
- Then, the software can write values to the corresponding LATx 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 PORTx and TRISx registers have no effect and cannot be used to read or write the RX and TX pins. Most communication peripheral functions available on the dsPIC33/PIC24 devices 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. For more information on how pins can be configured for a module, refer to the specific module section.

3.8.3 MULTIPLEXING DIGITAL INPUT PERIPHERAL

The following conditions are characteristic of a multiplexed digital input peripheral:

- Peripheral does not control the TRISx register. Some peripherals require the pin be configured as an input by setting the corresponding TRISx bit = 1.
- Peripheral input path is independent of I/O input path and uses an input buffer that is dependent on the peripheral.
- PORTx register data input path is not affected and is able to read the pin value.

3.8.4 MULTIPLEXING DIGITAL OUTPUT PERIPHERAL

The following conditions are characteristic of a multiplexed digital output peripheral:

- Peripheral controls the output data. Some peripherals require the pin be configured as an output by setting the corresponding TRISx bit = 0.
- If a peripheral pin has an automatic tri-state feature (e.g., PWM outputs), the peripheral has the ability to tri-state the pin.
- Pin output driver type could be affected by the peripheral (e.g., drive strength, slew rate, etc.).
- PORTx register output data has no effect.

3.8.5 MULTIPLEXING DIGITAL BIDIRECTIONAL PERIPHERAL

The following conditions are characteristic of a multiplexed digital bidirectional peripheral:

- Peripheral automatically configures the pin as an output, but not as an input. Some peripherals require the pin be configured as an input by setting the corresponding TRISx bit = 1.
- Peripherals control output data.
- Pin output driver type could be affected by the peripheral (e.g., drive strength, slew rate, etc.).
- PORTx register data input path is not affected and is able to read the pin value.
- PORTx register output data has no effect.

3.8.6 MULTIPLEXING ANALOG INPUT PERIPHERAL

The following condition is characteristic of a multiplexed analog input peripheral:

- All digital port input buffers are disabled and PORTx registers read '0' to prevent "crowbar" current.

3.8.7 MULTIPLEXING ANALOG OUTPUT PERIPHERAL

The following conditions are characteristic of a multiplexed analog output peripheral:

- All digital port input buffers are disabled and PORTx registers read '0' to prevent crowbar current.
- Analog output is driven onto the pin independent of the associated TRISx setting.

Note:	To use pins that are multiplexed with the ADC module for digital I/Os, the corresponding bits in the ANSELx register, if present, must be set to '0', even if the ADC module is turned off.

3.8.8 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 TRISx control bit, the user can manually affect the state of the input capture pin through its corresponding LATx register. This behavior can be useful in some situations, especially for testing purposes, when no external signal is connected to the input pin.

As shown previously in Figure 3-2, the organization of the peripheral multiplexers determines whether the peripheral input pin can be manipulated in software using the PORTx register. The conceptual peripherals shown in this figure disconnect the PORTx 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 LATx registers:

- External Interrupts pins
- Input Capture pins
- Timer Clock Input 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 PORTx registers. These peripherals include the following modules:

- SPI
- I^2C
- UART

3.9 Change Notice (CN) Pins

The CN pins provide dsPIC33/PIC24 devices the ability to generate interrupt requests to the processor in response to a Change-of-State (COS) on selected input pins (corresponding TRISx bits must be = 1). The total number of available CN inputs is dependent on the selected dsPIC33/PIC24 device. Refer to the **"I/O Ports"** chapter in the specific device data sheet for further details.

The enabled pin values are compared with the values sampled during the last read operation of the designated PORTx register. If the pin value is different from the last value read, a mismatch condition is generated. The mismatch condition can occur on any of the enabled input pins. The mismatches are "ORed" together to provide a single interrupt-on-change signal. The enabled pins are sampled on every internal system clock cycle, SYSCLK.

3.9.1 CN CONFIGURATION AND OPERATION

The CN pins are configured as follows:

1. Disable CPU interrupts.
2. Set the desired CN I/O pin as an input by setting the corresponding TRISx register bits = 1.

Note:	If the I/O pin is shared with an analog peripheral, it may be necessary to configure this pin as digital input.

3. Enable the CN Module by setting the ON bit (CNCONx<15>) = 1.
4. Enable individual CN input pins, enable optional pull-ups or pull-downs.
5. Read the corresponding PORTx registers to clear the CN interrupt.
6. Configure the CNx Interrupt Priority bits, CNxIP<2:0>.
7. Clear the CNx Interrupt Flag bit by setting the CNxIF bit (IFSx register) = 0.
8. Configure the CNx pin interrupt for either Mismatch mode or Edge Detect mode using the CNSTYLE bit (CNCONx<11>). If Mismatch mode is selected, enable the individual CNx function using the CNEN0x bits. If Edge Detect mode is selected, use the CNEN0x bits to enable positive edge detection and the CNEN1x bits to enable negative edge detection.
9. Enable the CNx Interrupt Enable bit by setting the CNxIE bit (IECx register) = 1.
10. Enable CPU interrupts.

The CNSTATx/CNFx registers indicate whether a change occurred on the corresponding pin since the last read of the PORTx bit.

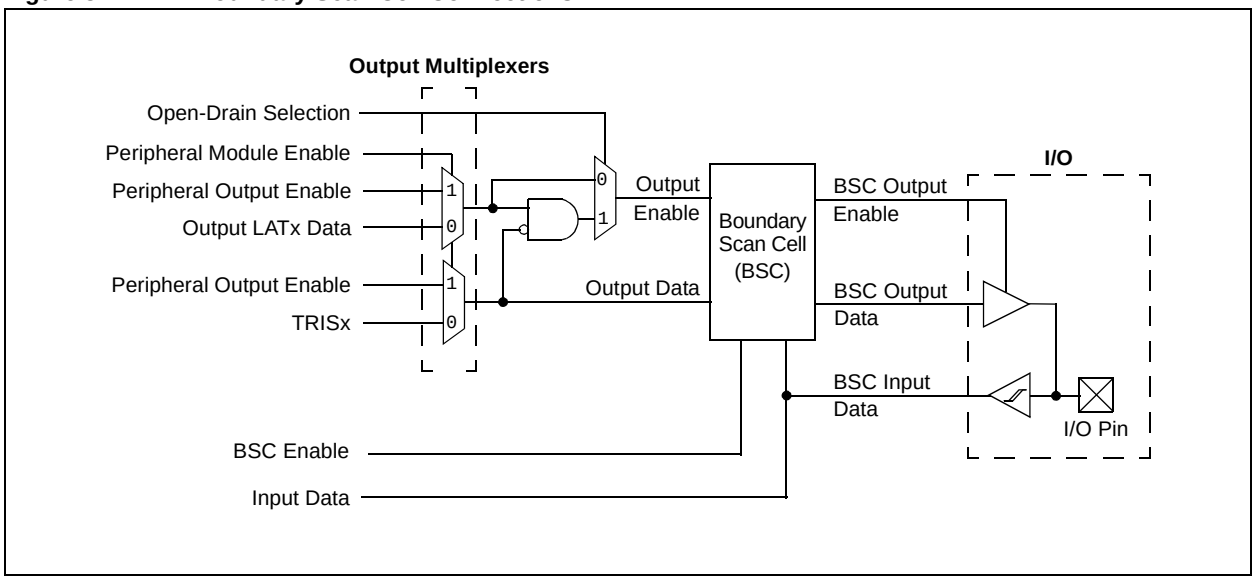
The CNFx registers indicate a valid edge detect event has occurred when CNSTYLE = 1. CNFx bits need to be cleared by the user to set up the CN logic to detect the next edge transition. In Edge Detect mode, a CN interrupt can be controlled to occur only during a rising or falling edge condition on a pin. The CNSTATx are read-only registers that indicate a valid Mismatch mode event has occurred when CNSTYLE = 0.

When a CN interrupt occurs in Mismatch mode, the user should read the PORTx register associated with the CN pins. This will clear the mismatch condition and set up the CN logic to detect the next pin change. The CN pins have a minimum input pulse-width specification. Refer to the **"Electrical Characteristics"** chapter of the specific device data sheet to learn more.

3.10 Boundary Scan Cell Connections

The dsPIC33/PIC24 devices support JTAG boundary scan. A Boundary Scan Cell (BSC) is inserted between the internal I/O logic circuit and the I/O pin, as shown in Figure 3-4. Most of the I/O pads have Boundary Scan Cells; however, JTAG pads do not. For normal I/O operation, the BSC is disabled, and therefore, is bypassed. The output enable input of the BSC is directly connected to the BSC output enable, and the output data input of the BSC is directly connected to the BSC output data. The pads that do not have BSC are the power supply pads (V_{DD}, V_{SS} and V_{CAP}/V_{CORE}) and the JTAG pads (TCK, TDI, TDO and TMS).

Figure 3-4: Boundary Scan Cell Connections



4.0 OPERATION IN POWER-SAVING MODES

4.1 I/O Port Operation in Sleep Mode

As the device enters Sleep mode, the system clock is disabled; however, the CN module continues to operate asynchronously. If one of the enabled CN pins changes state, the CNxIF bit (IFSx register) will be set. If the CNxIE bit (IECx register) is set, and its priority is greater than the current CPU priority, the device will wake from Sleep mode and execute the CN Interrupt Service Routine (ISR).

If the assigned priority level of the CN interrupt is less than or equal to the current CPU priority level, the CPU will not be awakened and the device will enter Idle mode.

4.2 I/O Port Operation in Idle Mode

As the device enters Idle mode, the system clock sources remain functional and the CN module continues to operate synchronously. If one of the enabled CN pins changes state, the CNxIF bit (IFSx register) will be set. If the CNxIE bit (IECx register) is set, and its priority is greater than the current CPU priority, the device will wake from Idle mode and execute the CN Interrupt Service Routine (ISR).

5.0 EFFECTS OF VARIOUS RESETS

5.1 Device Reset

All I/O registers are forced to their Reset states upon a device Reset.

5.2 Power-on Reset

All I/O registers are forced to their Reset states upon a Power-on Reset (POR).

5.3 Watchdog Timer Reset

All I/O registers are unchanged upon a Watchdog Timer Reset.

6.0 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 dsPIC33/PIC24 device families, but the concepts are pertinent and could be used with modification and possible limitations. The current application notes related to the I/O Ports with Edge Detect are:

Title	Application Note #
No related application notes at this time	N/A

Note: Please visit the Microchip web site (http://www.microchip.com) for additional application notes and code examples for the dsPIC33/PIC24 families of devices.

7.0 REVISION HISTORY

Revision A (August 2017)

This is the initial version of this document.

Revision B (February 2018)

Removed Continuous State Monitoring information from **Section 3.0 "Peripheral Pin Select (PPS)"**. Added note to **Section 2.1 "Registers for Configuring Tri-State Functions (TRISx)"**.

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
═ ISO/TS 16949 ═

Trademarks

ISBN: 978-1-5224-2689-9

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: 45-4450-2828
Fax: 45-4485-2829

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

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

Germany - Garching
Tel: 49-8931-9700

Germany - Haan
Tel: 49-2129-3766400

Germany - Heilbronn
Tel: 49-7131-67-3636

Germany - Karlsruhe
Tel: 49-721-625370

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

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

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

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

Italy - Padova
Tel: 39-049-7625286

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

Norway - Trondheim
Tel: 47-7289-7561

Poland - Warsaw
Tel: 48-22-3325737

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

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

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

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

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