# Section 10. I/O Ports

## HIGHLIGHTS

This section contains the following major topics:

**Note:** This family reference manual section is meant to serve as a complement to device data sheets. Depending on the device variant, this manual section may not apply to all dsPIC33E/PIC24E 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.

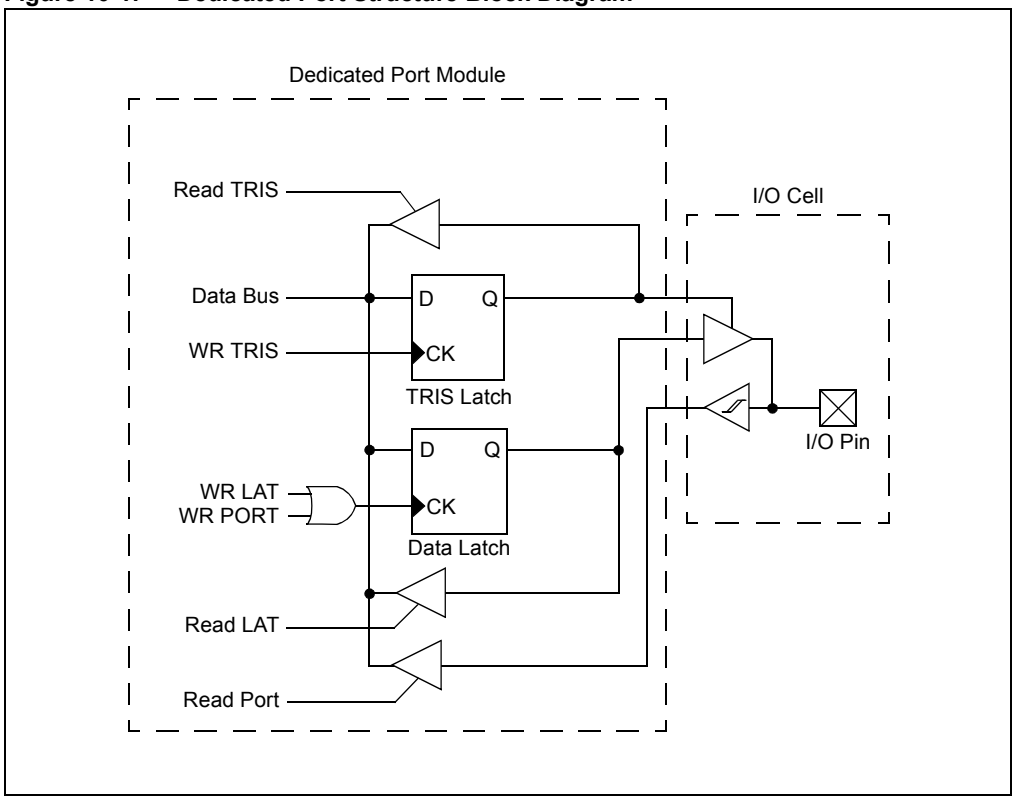
## 10.1 INTRODUCTION

This section provides information on the I/O ports with Peripheral Pin Select (PPS) for the dsPIC33E/PIC24E family of devices. Most device pins are shared between the peripherals and the general purpose I/O ports.

The general purpose I/O ports allow the dsPIC33E/PIC24E devices to monitor and control other devices. Most I/O pins are multiplexed with alternate functions. The multiplexing depends on the peripheral features of the device variant. In general, when a peripheral is functioning, that pin may not be used as a general purpose I/O pin.

Figure 10-1 illustrates a block diagram of a typical I/O port. This block diagram does not consider peripheral functions that might be multiplexed onto the I/O pin.

**Figure 10-1: Dedicated Port Structure Block Diagram**



## 10.2 I/O PORT CONTROL REGISTERS

All I/O ports have four 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
- ODCx: Open-Drain Control 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. For more information, refer to the specific device data sheet. |
|---|---|

### 10.2.1 TRISx 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 as an output. An easy way to remember this 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.

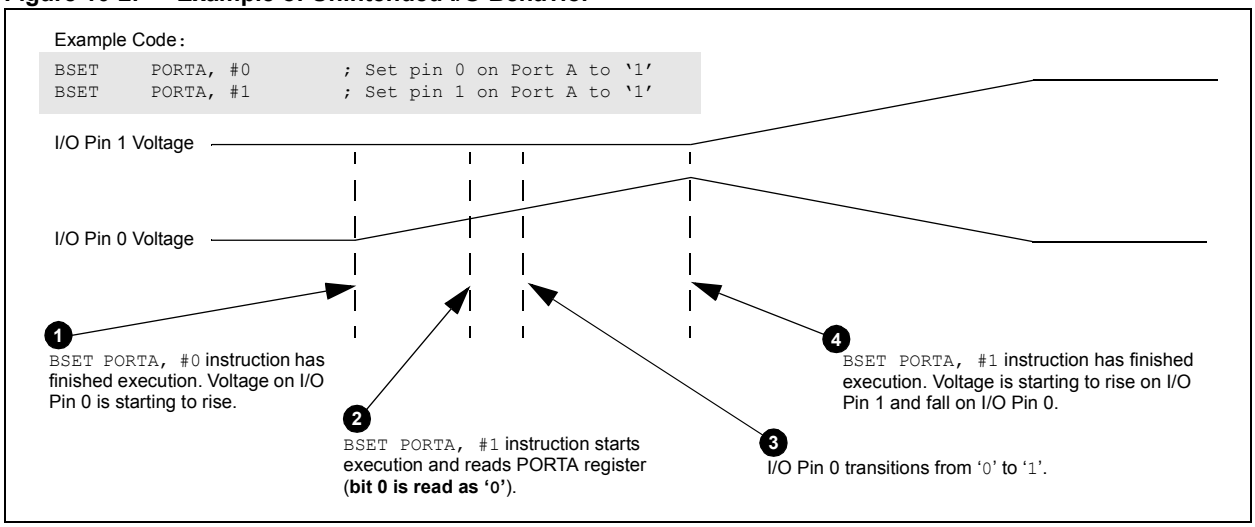
### 10.2.2 PORTx Registers

Data on an I/O pin is accessed through 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 the 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 it is written to the port data latch. Care should be taken when read-modify-write commands are used on the PORTx registers and when 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 later, 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.

In addition, if read-modify-write instructions are used on the PORTx registers while I/O pins are configured as outputs, unintended I/O behavior may occur based on the device speed and I/O capacitive loading. Figure 10-2 illustrates unintended behavior that occurs if the user-assigned application attempts to set I/O bits, 0 and 1 on PORTA, with two consecutive read-modify-write instructions in the PORTA register. At high CPU speeds and high capacitive loading on the I/O pins, the unintended result of the example code is that only I/O bit 1 is set high.

**Figure 10-2: Example of Unintended I/O Behavior**



When the first BSET instruction is executed, it writes a '1' to bit 0 in the PORTA register, which causes the voltage level to start rising to logic level '1' on Pin 0 (see Step 1 in Figure 10-2). However, if the second BSET instruction is executed before the voltage level on Pin 0 has reached the threshold for logic '1' (see Step 3 in Figure 10-2), the second BSET (read-modify-write) instruction reads '0' for bit 0, which it writes back into the PORTA register (see Step 2 in Figure 10-2). In other words, instead of reading a value of 0x0001 from the PORTA register, it reads a value of 0x0000, modifies it to 0x0002 (instead of the desired value of 0x0003) and writes that value back to the PORTA register. This causes the voltage on Pin 0 to start falling to logic level '0' and the voltage on Pin 1 to start rising to logic level '1' (see Step 4 in Figure 10-2).

### 10.2.3 LATx Registers

The LATx register associated with an I/O pin eliminates the problems that can occur with read-modify-write instructions. A read of the LATx register returns the values that are 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 following example uses the LATx register to set two I/O bits.

**Example 10-1: Setting I/O Pins with the LATx Register**

```
BSET    LATA, #0              ;Set pin 0 on Port A to '1'
BSET    LATA, #1              ;Set pin 1 on Port A to '1'
```

The differences between the PORTx and LATx 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. This means the corresponding LATx and TRISx registers, and the port pin, will read as zero.

### 10.2.4 Open-Drain Control Registers

In addition to the PORT, LAT and TRIS registers for data control, each port pin can also be individually configured for either a digital or open-drain output. This is controlled by the Open-Drain Control register, ODCx, associated with each port. Setting any of the bits configures the corresponding pin to act as an open-drain output.

The open-drain feature allows the generation of outputs higher than VDD (for example, 5V) on any desired 5V tolerant pins by using external pull-up resistors. (The open-drain I/O feature is not supported on pins that are not 5V tolerant.) The maximum open-drain voltage allowed is the same as the maximum VIH specification. The open-drain output feature is supported for both port pin and peripheral configurations.

| **Note:** | Refer to the specific device data sheet for more information on which pins are 5V tolerant. |
|---|---|

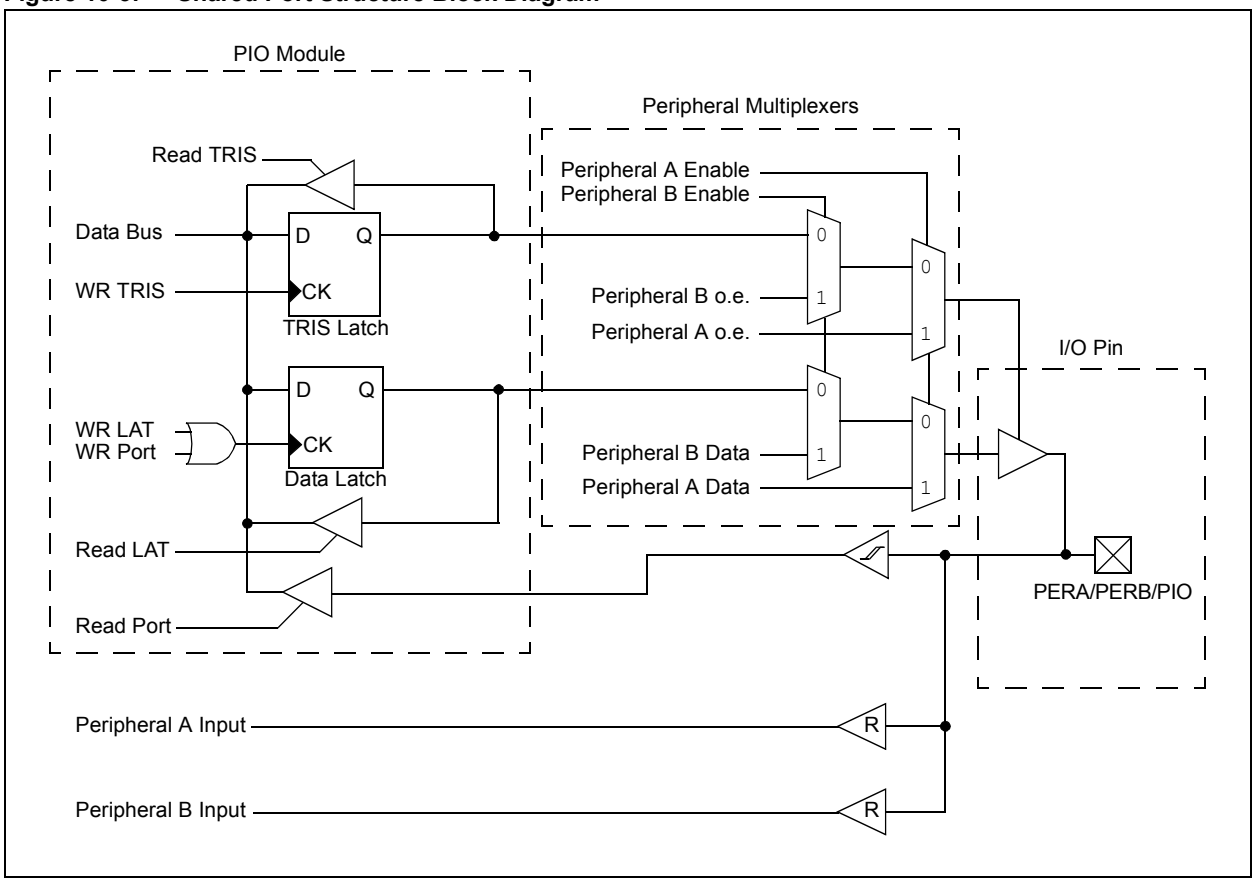
## 10.3 PERIPHERAL MULTIPLEXING

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 10-3 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 10-3: Shared Port Structure Block Diagram**




### 10.3.1 I/O Multiplexing with Multiple Peripherals

For some dsPIC33E/PIC24E devices, particularly those with a small number of I/O pins, multiple peripheral functions may be multiplexed on each I/O pin. Figure 10-3 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 10-3, 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 10-3, 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.

#### 10.3.1.1 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 10-3, 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 Figure 10-3 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 PORT 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 PORT registers. These peripherals include the following:

- SPI
- $I^2C$™
- DCI
- UART
- ECAN™
- QEI

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

#### 10.3.1.2 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 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 TRIS 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 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 dsPIC33E/PIC24E 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.

## 10.4 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 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.

### 10.4.1 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. |

### 10.4.2 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/O 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.



### 10.4.3 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.

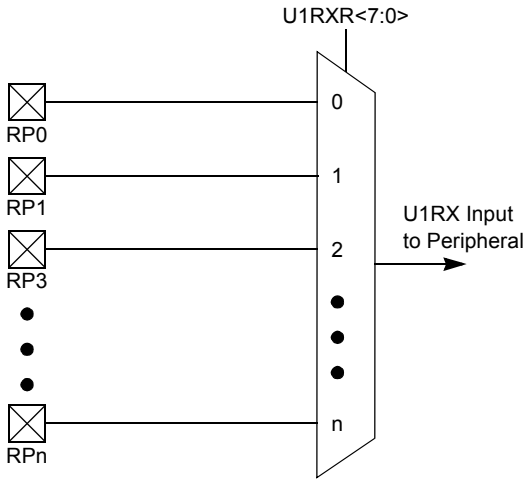
#### 10.4.3.1 INPUT MAPPING

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 10-4 illustrates remappable pin selection for the U1RX input.

**Figure 10-4: 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 10-1: 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> |
| External Interrupt 4 | INT4 | RPINR2 | INT4R<7:0> |
| Timer1 External Clock | T1CK | RPINR2 | T1CKR<7:0> |
| Timer2 External Clock | T2CK | RPINR3 | T2CKR<7:0> |
| Timer3 External Clock | T3CK | RPINR3 | T3CKR<7:0> |
| Timer4 External Clock | T4CK | RPINR4 | T4CKR<7:0> |
| Timer5 External Clock | T5CK | RPINR5 | T5CKR<7:0> |
| Timer6 External Clock | T6CK | RPINR6 | T6CKR<7:0> |
| Timer7 External Clock | T7CK | RPINR7 | T7CKR<7:0> |
| Timer8 External Clock | T8CK | RPINR8 | T8CKR<7:0> |
| Input Capture 1 | IC1 | RPINR7 | IC1R<7:0> |
| Input Capture 2 | IC2 | RPINR7 | IC2R<7:0> |
| Input Capture 3 | IC3 | RPINR8 | IC3R<7:0> |
| Input Capture 4 | IC4 | RPINR8 | IC4R<7:0> |
| Input Capture5 | IC5 | RPINR9 | IC5R<7:0> |
| Input Capture 6 | IC6 | RPINR9 | IC6R<7:0> |
| Input Capture 7 | IC7 | RPINR10 | IC7R<7:0> |
| Input Capture 8 | IC8 | RPINR10 | IC8R<7:0> |
| Output Compare Fault A | OCFA | RPINR11 | OCFAR<7:0> |
| Output Compare Fault B | OCFB | RPINR11 | OCFBR<7:0> |
| PWM Fault 1 | FLT1 | RPINR12 | FLT1R<7:0> |
| PWM Fault 2 | FLT2 | RPINR12 | FLT2R<7:0> |
| PWM Fault 3 | FLT3 | RPINR13 | FLT3R<7:0> |
| PWM Fault 4 | FLT4 | RPINR13 | FLT4R<7:0> |
| QEI1 Phase A | QEA1 | RPINR14 | QEA1R<7:0> |
| QEI1 Phase B | QEB1 | RPINR14 | QEB1R<7:0> |
| QEI1 Index | INDX1 | RPINR15 | INDX1R<7:0> |
| QEI1 Home | HOME1 | RPINR15 | HOM1R<7:0> |
| QEI2 Phase A | QEA2 | RPINR16 | QEA2R<7:0> |
| QEI2 Phase B | QEB2 | RPINR16 | QEB2R<7:0> |
| QEI2 Index | INDX2 | RPINR17 | INDX2R<7:0> |
| QEI2 Home | HOME2 | RPINR17 | HOM2R<7:0> |
| UART1 Receive | U1RX | RPINR18 | U1RXR<7:0> |
| UART1 Clear-to-Send | U1CTS | RPINR18 | U1CTSR<7:0> |
| UART2 Receive | U2RX | RPINR19 | U2RXR<7:0> |
| UART2 Clear-to-Send | U2CTS | RPINR19 | U2CTSR<7:0> |
| SPI1 Data Input | SDI1 | RPINR20 | SDI1R<7:0> |
| SPI1 Clock Input | SCK1 | RPINR20 | SCK1R<7:0> |
| SPI1 Slave Select | SS1 | RPINR21 | SS1R<7:0> |
| SPI2 Data Input | SDI2 | RPINR22 | SDI2R<7:0> |
| SPI2 Clock Input | SCK2 | RPINR22 | SCK2R<7:0> |

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

**Table 10-1: Selectable Input Sources (Maps Input to Function) (Continued)**

| Input Name(1) | Function Name | Register | Configuration Bits |
|---|---|---|---|
| SPI2 Slave Select | SS2 | RPINR23 | SS2R<7:0> |
| DCI Data Input | CSDI | RPINR24 | CSDIR<7:0> |
| DCI Clock Input | CSCKIN | RPINR24 | CSCKR<7:0> |
| DCI FSYNC Input | COFSIN | RPINR25 | COFSR<7:0> |
| CAN1 Receive | C1RX | RPINR26 | C1RXR<7:0> |
| CAN2 Receive | C2RX | RPINR26 | C2RXR<7:0> |
| UART3 Receive | U3RX | RPINR27 | U3RXR<7:0> |
| UART3 Clear-to-Send | U3CTS | RPINR27 | U3CTSR<7:0> |
| UART4 Receive | U4RX | RPINR28 | U4RXR<7:0> |
| UART4 Clear-to-Send | U4CTS | RPINR28 | U4CTSR<7:0> |
| SPI3 Data Input | SDI3 | RPINR29 | SDI3R<7:0> |
| SPI3 Clock Input | SCK3 | RPINR29 | SCK3R<7:0> |
| SPI3 Slave Select | SS3 | RPINR30 | SS3R<7:0> |
| SPI4 Data Input | SDI4 | RPINR31 | SDI4R<7:0> |
| SPI4 Clock Input | SCK4 | RPINR31 | SCK4R<7:0> |
| SPI4 Slave Select | SS4 | RPINR32 | SS4R<7:0> |
| Input Capture 9 | IC9 | RPINR33 | IC9<7:0> |
| Input Capture 10 | IC10 | RPINR33 | IC10<7:0> |
| Input Capture 11 | IC11 | RPINR34 | IC11<7:0> |
| Input Capture 12 | IC12 | RPINR34 | IC12<7:0> |
| Input Capture 13 | IC13 | RPINR35 | IC13<7:0> |
| Input Capture 14 | IC14 | RPINR35 | IC14<7:0> |
| Input Capture 15 | IC15 | RPINR36 | IC15<7:0> |
| Input Capture 16 | IC16 | RPINR36 | IC16<7:0> |
| Output Compare Fault C | OCFC | RPINR37 | OCFCR<7:0> |
| Primary Master Time Base Synchronization Input 1 | PSYNCI1 | RPINR37 | PSYCI1R<7:0> |
| Primary Master Time Base Synchronization Input 2 | PSYNCI2 | RPINR38 | PSYCI2R<7:0> |
| PWM Fault 5 | FLT5 | RPINR42 | FLT5R<7:0> |
| PWM Fault 6 | FLT6 | RPINR42 | FLT6R<7:0> |
| PWM Fault 7 | FLT7 | RPINR43 | FLT7R<7:0> |
| PWM Fault 8 | FLT8 | RPINR43 | FLT8R<7:0> |

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

**Note:** Table 10-1 and Figure 10-4 provide examples of selectable input sources for a generic device. For more information, refer to the specific device data sheet.

### 10.4.3.2 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 10-2 and Figure 10-5).

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 10-5: Multiplexing of Remappable Outputs for RPn**

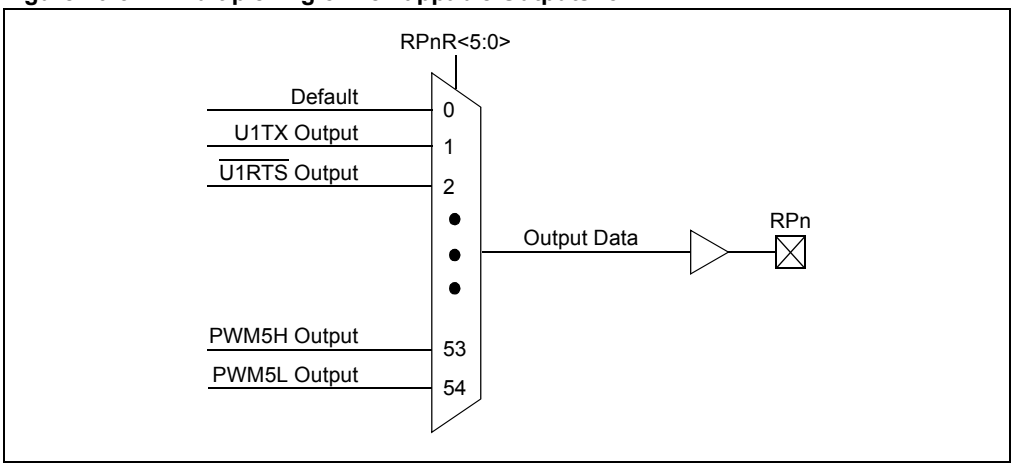



**Table 10-2: Output Selection for Remappable Pins (RPn)**

| Function | RPnR<5:0> | Output Name |
|---|---|---|
| DEFAULT PORT | 000000 | RPn tied to Default Pin |
| U1TX | 000001 | RPn tied to UART1 Transmit |
| U1RTS | 000010 | RPn tied to UART1 Ready-to-Send |
| U2TX | 000011 | RPn tied to UART2 Transmit |
| U2RTS | 000100 | RPn tied to UART2 Ready-to-Send |
| SDO1 | 000101 | RPn tied to SPI1 Data Output |
| SCK1OUT | 000110 | RPn tied to SPI1 Clock Output |
| SS1OUT | 000111 | RPn tied to SPI1 Slave Select |
| SDO2 | 001000 | RPn tied to SPI2 Data Output |
| SCK2OUT | 001001 | RPn tied to SPI2 Clock Output |
| SS2OUT | 001010 | RPn tied to SPI2 Slave Select |
| CSDO | 001011 | RPn tied to DCI Data Output |
| CSCKOUT | 001100 | RPn tied to DCI Clock Output |
| COFSOUT | 001101 | RPn tied to CAN1 FSYNC Output |
| C1TX | 001110 | RPn tied to CAN1 Transmit |
| C2TX | 001111 | RPn tied to CAN2 Transmit |
| OC1 | 010000 | RPn tied to Output Compare 1 Output |
| OC2 | 010001 | RPn tied to Output Compare 2 Output |
| OC3 | 010010 | RPn tied to Output Compare 3 Output |
| OC4 | 010011 | RPn tied to Output Compare 4 Output |
| OC5 | 010100 | RPn tied to Output Compare 5 Output |
| OC6 | 010101 | RPn tied to Output Compare 6 Output |

**Table 10-2: Output Selection for Remappable Pins (RPn) (Continued)**

| Function | RPnR<5:0> | Output Name |
|---|---|---|
| OC7 | 010110 | RPn tied to Output Compare 7 Output |
| OC8 | 010111 | RPn tied to Output Compare 8 Output |
| C1OUT (ACMP1) | 011000 | RPn tied to Comparator Output 1 |
| C2OUT (ACMP2) | 011001 | RPn tied to Comparator Output 2 |
| C3OUT (ACMP3) | 011010 | RPn tied to Comparator Output 3 |
| U3TX | 011011 | RPn tied to UART3 Transmit |
| U3RTS | 011100 | RPn tied to UART3 Ready-to-Send |
| U4TX | 011101 | RPn tied to UART4 Transmit |
| U4RTS | 011110 | RPn tied to UART4 Ready-to-Send |
| SDO3 | 011111 | RPn tied to SPI3 Data Output |
| SCK3OUT | 100000 | RPn tied to SPI3 Clock Output |
| SS3OUT | 100001 | RPn tied to SPI3 Slave Select |
| SDO4 | 100010 | RPn tied to SPI4 Data Output |
| SCK4OUT | 100011 | RPn tied to SPI4 Clock Output |
| SS4OUT | 100100 | RPn tied to SPI4 Slave Select |
| OC9 | 100101 | RPn tied to Output Compare 9 Output |
| OC10 | 100110 | RPn tied to Output Compare 10 Output |
| OC11 | 100111 | RPn tied to Output Compare 11 Output |
| OC12 | 101000 | RPn tied to Output Compare 12 Output |
| OC13 | 101001 | RPn tied to Output Compare 13 Output |
| OC14 | 101010 | RPn tied to Output Compare 14 Output |
| OC15 | 101011 | RPn tied to Output Compare 15 Output |
| OC16 | 101100 | RPn tied to Output Compare 16 Output |
| SYNCO1 | 101101 | RPn tied to PWM Primary Time Base Sync Output |
| SYNCO2 | 101110 | RPn tied to PWM Secondary Time Base Sync Output |
| QEI1CCMP | 101111 | RPn tied to QEI1 Counter Comparator Output |
| QEI2CCMP | 110000 | RPn tied to QEI2 Counter Comparator Output |
| REFCLKO | 110001 | RPn tied to Reference Clock Output |
| C4OUT (ACMP4) | 110010 | RPn tied to Comparator Output 4 |
| PWM4H | 110011 | RPn tied to PWM4H Output |
| PWM4L | 110100 | RPn tied to PWM4L Output |
| PWM5H | 110101 | RPn tied to PWM5H Output |
| PWM5L | 110110 | RPn tied to PWM5L Output |

**Note:** Figure 10-5 and Table 10-2 provide examples of a generic device. For more information, refer to the specific device data sheet.

### 10.4.3.3 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.

### 10.4.4 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 dsPIC33E/PIC24E devices include three features to prevent alterations to the peripheral map:

- Control register lock sequence
- Continuous state monitoring
- Configuration bit remapping lock

#### 10.4.4.1 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 (OSCCON<6>). Setting IOLOCK prevents writes to the control registers; clearing IOLOCK allows writes.

To set or clear IOLOCK, a specific command sequence must be executed:

1. Write 0x46 to OSCCON<7:0>.
2. Write 0x57 to OSCCON<7:0>.
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 built-in C language functions for unlocking the OSCCON register:

```
__builtin_write_OSCCONL(value)
__builtin_write_OSCCONH(value)
```

For more information, see the MPLAB C30 Help files.

#### 10.4.4.2 CONTINUOUS STATE MONITORING

In addition to being protected from direct writes, the contents of the RPINRx and RPORx registers are constantly monitored in hardware by shadow registers. If an unexpected change in any of the registers occurs (such as cell disturbances caused by ESD or other external events), a Configuration Mismatch Reset will be triggered.

#### 10.4.4.3 CONFIGURATION BIT PIN SELECT LOCK

As an additional level of safety, the device can be configured to prevent more than one write session to the RPINRx and RPORx registers. The IOL1WAY (FOSC<5>) Configuration bit blocks the IOLOCK bit from being cleared after it has set once.

In the default (unprogrammed) state, IOL1WAY is set, restricting users to one write session. Programming IOL1WAY allows users unlimited access (with the proper use of the unlock sequence) to the Peripheral Pin Select registers.

### 10.4.5 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 unlock sequence must be executed as an assembly language routine, similar to changes to the oscillator configuration, because the unlock sequence is timing critical. 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.

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/O. 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 10-2 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 10-2: Configuring UART1 Input and Output Functions**

```
//************************************************************
// Unlock Registers
//************************************************************
__builtin_write_OSCCONL(OSCCON & ~(1<<6));

//************************************************************
// Configure Input Functions
// (See Table 10-1)
//************************************************************
//******************************
// Assign U1Rx To Pin RP16
//******************************
RPINR18bits.U1RXR = 0x10;

//******************************
// Assign U1CTS To Pin RP17
//******************************
RPINR18bits.U1CTSR = 0x11;

//************************************************************
// Configure Output Functions
// (See Table 10-2)
//************************************************************
//******************************
// Assign U1Tx To Pin RP34
//******************************
RPOR1bits.RP34 = 1;

//******************************
// Assign U1RTS To Pin RP35
//******************************
RPOR1bits.RP35 = 2;

//************************************************************
// Lock Registers
//************************************************************
__builtin_write_OSCCONL(OSCCON | (1<<6));
```

### 10.4.6 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.

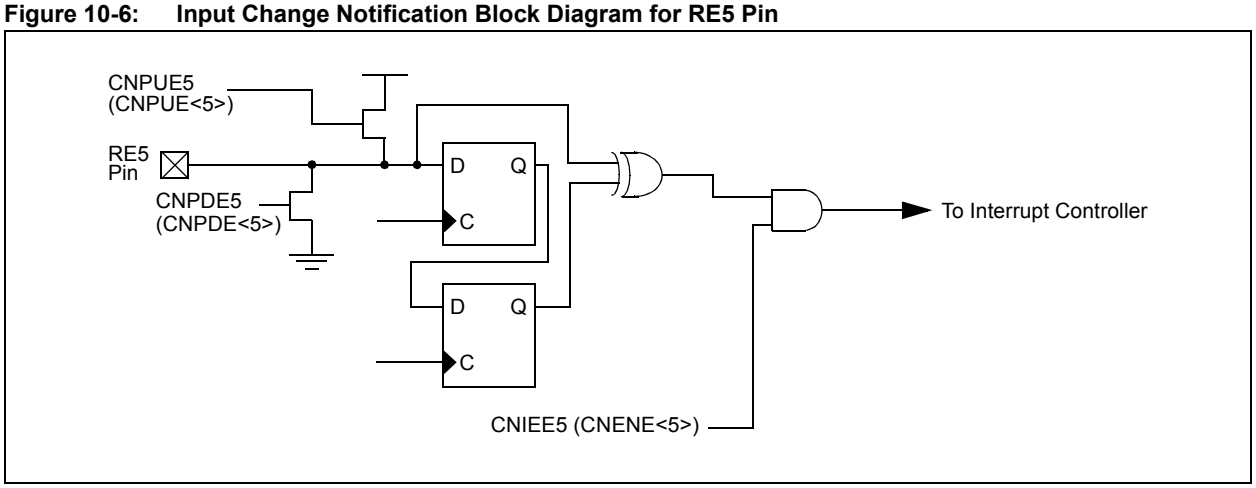
The concept of “virtual pins” enables new device features to be added to a device that were not initially conceived during the design of the original peripherals. One common use for the virtual pins is to connect the analog comparator, or output compare outputs, to the PWM module for use as current-limit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

## 10.5 CHANGE NOTIFICATION (CN)

The Change Notification (CN) functionality provides the dsPIC33E/PIC24E devices with the ability to generate interrupt requests to the processor in response to a Change-of-State (COS) on selected input pins.

Figure 10-6 illustrates the basic function of the CN hardware.

**Figure 10-6: Input Change Notification Block Diagram for RE5 Pin**




### 10.5.1 CN Control Registers

Three control registers are associated with the CN module: CNENx, CNPUx and CNPDx, where 'x' denotes the port letter.

The CNENx registers contain the CNIExy control bits, where 'x' denotes the port and where 'y' denotes the port pin number. The CNIExy bit must be set for a port input pin to interrupt the CPU.

Each CN pin has a weak pull-up and pull-down device connected to the pin, which can be enabled or disabled using the CNPUx and CNPDx control bits. The weak pull-up and pull-down devices act as a current source or sink 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 Characteristics" section of the specific device data sheet for CN pull-up and pull-down device current specifications.

### 10.5.2 CN Configuration and Operation

Change Notification is configured as follows:

1. Ensure that the port pin is configured as a digital input by setting the associated bit in the TRISx register.
2. Enable interrupts for the selected port pins by setting the appropriate bits in the CNENx register.
3. Turn on the weak pull-up devices (if desired) for the selected port pins by setting the appropriate bits in the CNPUx register.
4. Turn on the weak pull-down devices (if desired) for the selected port pins by setting the appropriate bits in the CNPDx register.
5. Clear the CNIF interrupt flag in the IFSx register.
6. Select the desired interrupt priority for CN interrupts using the CNIP<2:0> control bits in the IPCx register.
7. Enable CN interrupts using the CNIE control bit in the IECx register.

When a CN interrupt occurs, the user-assigned application should read the PORT 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 current PORT value can be compared to the PORT read value obtained at the last CN interrupt to determine the pin that changed.

The port pins have a minimum input pulse-width specification. For more information, refer to the "Electrical Characteristics" section of the specific device data sheet.

**Example 10-3: Configuring and Using CN Interrupts**

```
void configure CN(void)
{
    CNENAbits.CNIEA3 = 1;          // Enable RA3 pin for interrupt detection
    IEC1bits.CNIE = 1;             // Enable CN interrupts
    IFS1bits.CNIF = 0;             // Reset CN interrupt
}

void __attribute__ ((__interrupt__)) _CNInterrupt(void)
{
    // Insert ISR code here

    IFS1bits.CNIF = 0;             // Clear CN interrupt
}
```

## 10.6 CN OPERATION IN SLEEP AND IDLE MODES

The CN module continues to operate during Sleep or Idle mode. If one of the CN enabled port pins changes states, the CNIF status bit in the IFSx register will be set. If the CNIE bit is set in the IECx register, the device will wake from Sleep or Idle mode and resume operation.

If the assigned priority level of the CN interrupt is equal 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.

## 10.7 CONFIGURING ANALOG AND DIGITAL PORT PINS

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 ANSEL and TRIS bits set. To use port pins for I/O functionality with digital modules, such as Timers, UARTs and so on, the corresponding ANSELx bit must be cleared.

The ANSELx register has a default value of 0xFFFF. Therefore, all pins that share analog functions are analog (not digital) by default.

If the TRIS 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, like the ADC module or Comparator module.

When the PORT 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.

## 10.8 REGISTERS

### 10.8.1 Change Notification Registers

The following registers are used to enable and disable corresponding CN interrupts, and pull-up and pull-down resistors:

- **CNENx: Input Change Notification Interrupt Enable Register**
- **CNPUDx: Input Change Notification Pull-up Enable Register**
- **CNPDx: Input Change Notification Pull-Down Enable Register**
- **ANSELx: Analog Select Control Register**

### 10.8.2 Peripheral Pin Select Registers

The following registers are used to configure the input and output functionality of the dsPIC33E/PIC24E device pins:

- RPINR0 through RPINR43: Peripheral Pin Select Input Registers 0 through 43
- RPOR0 through PROR18: Peripheral Pin Select Output Registers 0 through 18

| **Note:** | dsPIC33E/PIC24E devices may have a varied number of RPINRx and RPORx registers. For more information, refer to the specific device data sheet. |
|---|---|

**Register 10-1: CNENx: Input Change Notification Interrupt Enable Register**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| CNIEx15 | CNIEx14 | CNIEx13 | CNIEx12 | CNIEx11 | CNIEx10 | CNIEx9 | CNIEx8 |
| bit 15 | | | | | | | bit 8 |

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| CNIEx7 | CNIEx6 | CNIEx5 | CNIEx4 | CNIEx3 | CNIEx2 | CNIEx1 | CNIEx0 |
| 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 15-0 **CNIExy:** Input Change Notification Interrupt Enable bits
1 = Enables interrupt on input change
0 = Disables interrupt on input change

**Register 10-2: CNPUDx: Input Change Notification Pull-up Enable Register**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| CNPUx15 | CNPUx14 | CNPUx13 | CNPUx12 | CNPUx11 | CNPUx10 | CNPUx9 | CNPUx8 |
| bit 15 | | | | | | | bit 8 |

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| CNPUx7 | CNPUx6 | CNPUx5 | CNPUx4 | CNPUx3 | CNPUx2 | CNPUx1 | CNPUx0 |
| 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 15-0 **CNPUxy:** Input Change Notification Pull-up Enable bits
1 = Enables pull-up on port pin
0 = Disables pull-up on port pin

**Register 10-3: CNPDx: Input Change Notification Pull-Down Enable Register**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| CNPDx15 | CNPDx14 | CNPDx13 | CNPDx12 | CNPDx11 | CNPDx10 | CNPDx9 | CNPDx8 |
| bit 15 | | | | | | | bit 8 |

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| CNPDx7 | CNPDx6 | CNPDx5 | CNPDx4 | CNPDx3 | CNPDx2 | CNPDx1 | CNPDx0 |
| 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 15-0 **CNPDxy:** Input Change Notification Pull-down Enable bits
1 = Enables pull-up on port pin
0 = Disables pull-up on port pin

**Register 10-4: ANSELx: Analog Select Control Register**

| R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |
|---|---|---|---|---|---|---|---|
| ANSx15 | ANSx14 | ANSx13 | ANSx12 | ANSx11 | ANSx10 | ANSx9 | ANSx8 |
| bit 15 | | | | | | | bit 8 |

| R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |
|---|---|---|---|---|---|---|---|
| ANSx7 | ANSx6 | ANSx5 | ANSx4 | ANSx3 | ANSx2 | ANSx1 | ANSx0 |
| 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 15-0 **ANSxy:** Analog Port Enable bits
1 = Enables analog port pin
0 = Enables digital port pin

## 10.9 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 dsPIC33E/PIC24E 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 with Peripheral Pin Select include the following:

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

**Note:** Visit the Microchip web site (www.microchip.com) for additional application notes and code examples for the dsPIC33E/PIC24E family of devices.

## 10.10 REVISION HISTORY

### Revision A (May 2009)

This is the initial release of this document.

### Revision B (June 2010)

This revision includes the following updates:

- Updated the first paragraph in **Section 10.1 "Introduction"** regarding the sharing of device pins.
- Updated the second paragraph in **Section 10.2.4 "Open-Drain Control Registers"**, which now refers to 5V pin tolerance.
- Updated the shaded note in **Section 10.3 "Peripheral Multiplexing"**, which now refers to analog module pins and the ANSELx register.
- Added a shaded note that provides information on RPn pin availability to **Section 10.4.3.1 "Input Mapping"**.
- Updated the Configuring UART1 Input and Output Functions code example (see Example 10-2).
- Made the following changes throughout **Section 10.5 "Change Notification (CN)"**:
  - Removed the word "PINS" from the section title
  - Removed the last three sentences from the first paragraph
  - Updated the Input Change Notification Block Diagram (see Figure 10-6)
  - References to the CNxIE bits have been changed to: CNIExy
  - Changed references to CN pins, which are now termed "port" pins
  - Updated the Configuring and Using CN Interrupts code example (see Example 10-3)
- Added the new text, **Section 10.7 "Configuring Analog and Digital Port Pins"**.
- Updated the list of available registers and revised the shaded note in **Section 10.8.2 "Peripheral Pin Select Registers"**.
- Removed all RPINRx and RPORx registers.

### Revision C (March 2013)

This revision includes the following updates:

- Updated Table 10-1 and Table 10-2
- Added the new text, **Section 10.4.6 "Virtual Output Pins"**
- In **Section 10.8 "Registers"**, changed RPIN37 to RPIN43 and changed PROR15 to PROR18.

**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**



Printed on recycled paper.

ISBN: 978-1-62077-116-7

**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

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

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

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

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

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

## ASIA/PACIFIC

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

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

**China - Beijing**
Tel: 86-10-8569-7000
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: 86-571-2819-3187
Fax: 86-571-2819-3189

**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: 91-20-2566-1512
Fax: 91-20-2566-1513

**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: 886-7-213-7828
Fax: 886-7-330-9305

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

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

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

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

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

11/29/12