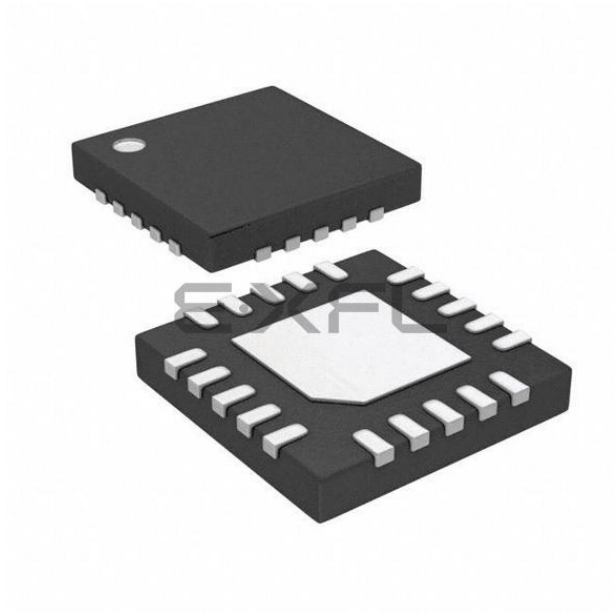


**Welcome to E-XFL.COM**

## What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

## Applications of "Embedded - Microcontrollers"

| Details | |
|---|---|
| Product Status | Active |
| Core Processor | PIC |
| Core Size | 8-Bit |
| Speed | 32MHz |
| Connectivity | I²C, LINbus, SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, POR, PWM, WDT |
| Number of I/O | 18 |
| Program Memory Size | 7KB (4K x 14) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 512 x 8 |
| Voltage - Supply (Vcc/Vdd) | 2.3V ~ 5.5V |
| Data Converters | A/D 12x10b; D/A 1x8b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 20-UFQFN Exposed Pad |
| Supplier Device Package | 20-UQFN (4x4) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic16f1618-i-gz |

## 1.1 Register and Bit Naming Conventions

### 1.1.1 REGISTER NAMES

When there are multiple instances of the same peripheral in a device, the peripheral control registers will be depicted as the concatenation of a peripheral identifier, peripheral instance, and control identifier. The control registers section will show just one instance of all the register names with an 'x' in the place of the peripheral instance number. This naming convention may also be applied to peripherals when there is only one instance of that peripheral in the device to maintain compatibility with other devices in the family that contain more than one.

### 1.1.2 BIT NAMES

There are two variants for bit names:

- Short name: Bit function abbreviation
- Long name: Peripheral abbreviation + short name

#### 1.1.2.1 Short Bit Names

Short bit names are an abbreviation for the bit function. For example, some peripherals are enabled with the EN bit. The bit names shown in the registers are the short name variant.

Short bit names are useful when accessing bits in C programs. The general format for accessing bits by the short name is *RegisterName*bits.*ShortName*. For example, the enable bit, EN, in the COG1CON0 register can be set in C programs with the instruction `COG1CON0bits.EN = 1`.

Short names are generally not useful in assembly programs because the same name may be used by different peripherals in different bit positions. When this occurs, during the include file generation, all instances of that short bit name are appended with an underscore plus the name of the register in which the bit resides to avoid naming contentions.

#### 1.1.2.2 Long Bit Names

Long bit names are constructed by adding a peripheral abbreviation prefix to the short name. The prefix is unique to the peripheral, thereby making every long bit name unique. The long bit name for the COG1 enable bit is the COG1 prefix, G1, appended with the enable bit short name, EN, resulting in the unique bit name G1EN.

Long bit names are useful in both C and assembly programs. For example, in C the COG1CON0 enable bit can be set with the `G1EN = 1` instruction. In assembly, this bit can be set with the `BSF COG1CON0,G1EN` instruction.

#### 1.1.2.3 Bit Fields

Bit fields are two or more adjacent bits in the same register. Bit fields adhere only to the short bit naming convention. For example, the three Least Significant bits of the COG1CON0 register contain the mode control bits. The short name for this field is MD. There is no long bit name variant. Bit field access is only possible in C programs. The following example demonstrates a C program instruction for setting the COG1 to the Push-Pull mode:

```
COG1CON0bits.MD = 0x5;
```

Individual bits in a bit field can also be accessed with long and short bit names. Each bit is the field name appended with the number of the bit position within the field. For example, the Most Significant mode bit has the short bit name MD2 and the long bit name is G1MD2. The following two examples demonstrate assembly program sequences for setting the COG1 to Push-Pull mode:

Example 1:

```
MOVLW  ~(1<<G1MD1)
ANDWF  COG1CON0,F
MOVLW  1<<G1MD2 | 1<<G1MD0
IORWF  COG1CON0,F
```

Example 2:

```
BSF    COG1CON0,G1MD2
BCF    COG1CON0,G1MD1
BSF    COG1CON0,G1MD0
```

### 1.1.3 REGISTER AND BIT NAMING EXCEPTIONS

#### 1.1.3.1 Status, Interrupt, and Mirror Bits

Status, interrupt enables, interrupt flags, and mirror bits are contained in registers that span more than one peripheral. In these cases, the bit name shown is unique so there is no prefix or short name variant.

#### 1.1.3.2 Legacy Peripherals

There are some peripherals that do not strictly adhere to these naming conventions. Peripherals that have existed for many years and are present in almost every device are the exceptions. These exceptions were necessary to limit the adverse impact of the new conventions on legacy code. Peripherals that do adhere to the new convention will include a table in the registers section indicating the long name prefix for each peripheral instance. Peripherals that fall into the exception category will not have this table. These peripherals include, but are not limited to, the following:

- EUSART
- MSSP

**TABLE 3-14: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)**

| Addr | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOR | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Banks 29** | | | | | | | | | | | |
| E8Ch to E8Fh | — | Unimplemented | | | | | | | | — | — |
| E90h | RA0PPS | — | — | — | RA0PPS<4:0> | | | | | ---0 0000 | ---0 0000 |
| E91h | RA1PPS | — | — | — | RA1PPS<4:0> | | | | | ---0 0000 | ---0 0000 |
| E92h | RA2PPS | — | — | — | RA2PPS<4:0> | | | | | ---0 0000 | ---0 0000 |
| E93h | — | Unimplemented | | | | | | | | — | — |
| E94h | RA4PPS | — | — | — | RA4PPS<4:0> | | | | | ---0 0000 | ---0 0000 |
| E95h | RA5PPS | — | — | — | RA5PPS<4:0> | | | | | ---0 0000 | ---0 0000 |
| E96h to E9Bh | — | Unimplemented | | | | | | | | — | — |
| E9Ch | RB4PPS(4) | — | — | — | RB4PPS<4:0> | | | | | ---0 0000 | ---0 0000 |
| E9Dh | RB5PPS(4) | — | — | — | RB5PPS<4:0> | | | | | ---0 0000 | ---0 0000 |
| E9Eh | RB6PPS(4) | — | — | — | RB6PPS<4:0> | | | | | ---0 0000 | ---0 0000 |
| E9Fh | RB7PPS(4) | — | — | — | RB7PPS<4:0> | | | | | ---0 0000 | ---0 0000 |
| EA0h | RC0PPS | — | — | — | RC0PPS<4:0> | | | | | ---0 0000 | ---0 0000 |
| EA1h | RC1PPS | — | — | — | RC1PPS<4:0> | | | | | ---0 0000 | ---0 0000 |
| EA2h | RC2PPS | — | — | — | RC2PPS<4:0> | | | | | ---0 0000 | ---0 0000 |
| EA3h | RC3PPS | — | — | — | RC3PPS<4:0> | | | | | ---0 0000 | ---0 0000 |
| EA4h | RC4PPS | — | — | — | RC4PPS<4:0> | | | | | ---0 0000 | ---0 0000 |
| EA5h | RC5PPS | — | — | — | RC5PPS<4:0> | | | | | ---0 0000 | ---0 0000 |
| EA6h | RC6PPS(4) | — | — | — | RC6PPS<4:0> | | | | | ---0 0000 | ---0 0000 |
| EA7h | RC7PPS(4) | — | — | — | RC7PPS<4:0> | | | | | ---0 0000 | ---0 0000 |
| EA8h to EEFh | — | Unimplemented | | | | | | | | — | — |

**Legend:** x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, r = reserved. Shaded locations are unimplemented, read as '0'.

**Note 1:** PIC16F1614/8 only.
**2:** Unimplemented, read as '1'.
**3:** PIC16(L)F1614 only.
**4:** PIC16(L)F1618 only.

**REGISTER 4-3: CONFIG3: CONFIGURATION WORD 3**

| R/P-0 | R/P-0 | R/P-1 | R/P-1 | R/P-1 | R/P-1 |
|---|---|---|---|---|---|
| WDTCCS<2:0> | | | WDTCWS<2:0> | | |
| bit 13 | | | | | bit 8 |

| U-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 |
|---|---|---|---|---|---|---|---|
| — | WDTE<1:0> | | WDTCPS<4:0> | | | | |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | P = Programmable bit | U = Unimplemented bit, read as ‘1’ |
| ‘0’ = Bit is cleared | ‘1’ = Bit is set | -n = Value when blank or after Bulk Erase |

bit 13-11 **WDTCCS<2:0>:** WDT Configuration Clock Select bits
111 =Software Control; WDT clock selected by CS<2:0>
110 =Reserved
•
•
•
010 =Reserved
001 =WDT reference clock is MFINTOSC, 31.25 kHz (default value)
000 =WDT reference clock is LFINTOSC, 31.00 kHz output

bit 10-8 **WDTCWS<2:0>:** WDT Configuration Window Select bits.

| WDTCWS <2:0> | WINDOW at POR | | | Software control of WINDOW? | Keyed access required? | |
|---|---|---|---|---|---|---|
| | Value | Window delay Percent of time | Window opening Percent of time | | | |
| 111 | 111 | n/a | 100 | Yes | No | Default fuse = 111 |
| 110 | 111 | n/a | 100 | No | Yes | |
| 101 | 101 | 25 | 75 | | | |
| 100 | 100 | 37.5 | 62.5 | | | |
| 011 | 011 | 50 | 50 | | | |
| 010 | 010 | 62.5 | 37.5 | | | |
| 001 | 001 | 75 | 25 | | | |
| 000 | 000 | 87.5 | 12.5(1) | | | |

bit 7 **Unimplemented:** Read as ‘1’

bit 6-5 **WDTE<1:0>:** Watchdog Timer Enable bits
11 =WDT enabled in all modes, the SEN bit in the WDTCON0 register is ignored
10 =WDT enabled while running and disabled in Sleep
01 =WDT controlled by the SEN bit in the WDTCON0 register
00 = WDT disabled

## 5.3 Clock Switching

The system clock source can be switched between external and internal clock sources via software using the System Clock Select (SCS) bits of the OSCCON register. The following clock sources can be selected using the SCS bits:

- Default system oscillator determined by FOSC bits in Configuration Words
- Internal Oscillator Block (INTOSC)

### 5.3.1 SYSTEM CLOCK SELECT (SCS) BITS

The System Clock Select (SCS) bits of the OSCCON register selects the system clock source that is used for the CPU and peripherals.

- When the SCS bits of the OSCCON register = 00, the system clock source is determined by value of the FOSC<1:0> bits in the Configuration Words.
- When the SCS bits of the OSCCON register = 1x, the system clock source is chosen by the internal oscillator frequency selected by the IRCF<3:0> bits of the OSCCON register. After a Reset, the SCS bits of the OSCCON register are always cleared.

When switching between clock sources, a delay is required to allow the new clock to stabilize. These oscillator delays are shown in Table 5-1.

**TABLE 5-1: OSCILLATOR SWITCHING DELAYS**

| Switch From | Switch To | Frequency | Oscillator Delay |
|---|---|---|---|
| Sleep | LFINTOSC(1)<br>MFINTOSC(1)<br>HFINTOSC(1) | 31 kHz<br>31.25 kHz-500 kHz<br>31.25 kHz-16 MHz | Oscillator Warm-up Delay (Tiosc st) |
| Sleep/POR | EC(1) | DC – 32 MHz | 2 cycles |
| LFINTOSC | EC(1) | DC – 32 MHz | 1 cycle of each |
| Any clock source | MFINTOSC(1)<br>HFINTOSC(1) | 31.25 kHz-500 kHz<br>31.25 kHz-16 MHz | 2 μs (approx.) |
| Any clock source | LFINTOSC(1) | 31 kHz | 1 cycle of each |
| PLL inactive | PLL active | 16-32 MHz | 2 ms (approx.) |

**Note 1:** PLL inactive.

## 8.3 Register Definitions: Voltage Regulator Control

**REGISTER 8-1: VREGCON: VOLTAGE REGULATOR CONTROL REGISTER(1)**

| U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | R/W-0/0 | R/W-1/1 |
|---|---|---|---|---|---|---|---|
| — | — | — | — | — | — | VREGPM | Reserved |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | |

bit 7-2 **Unimplemented:** Read as '0'

bit 1 **VREGPM**: Voltage Regulator Power Mode Selection bit
- 1 = Low-Power Sleep mode enabled in Sleep(2)
  Draws lowest current in Sleep, slower wake-up
- 0 = Normal Power mode enabled in Sleep(2)
  Draws higher current in Sleep, faster wake-up

bit 0 **Reserved:** Read as '1'. Maintain this bit set.

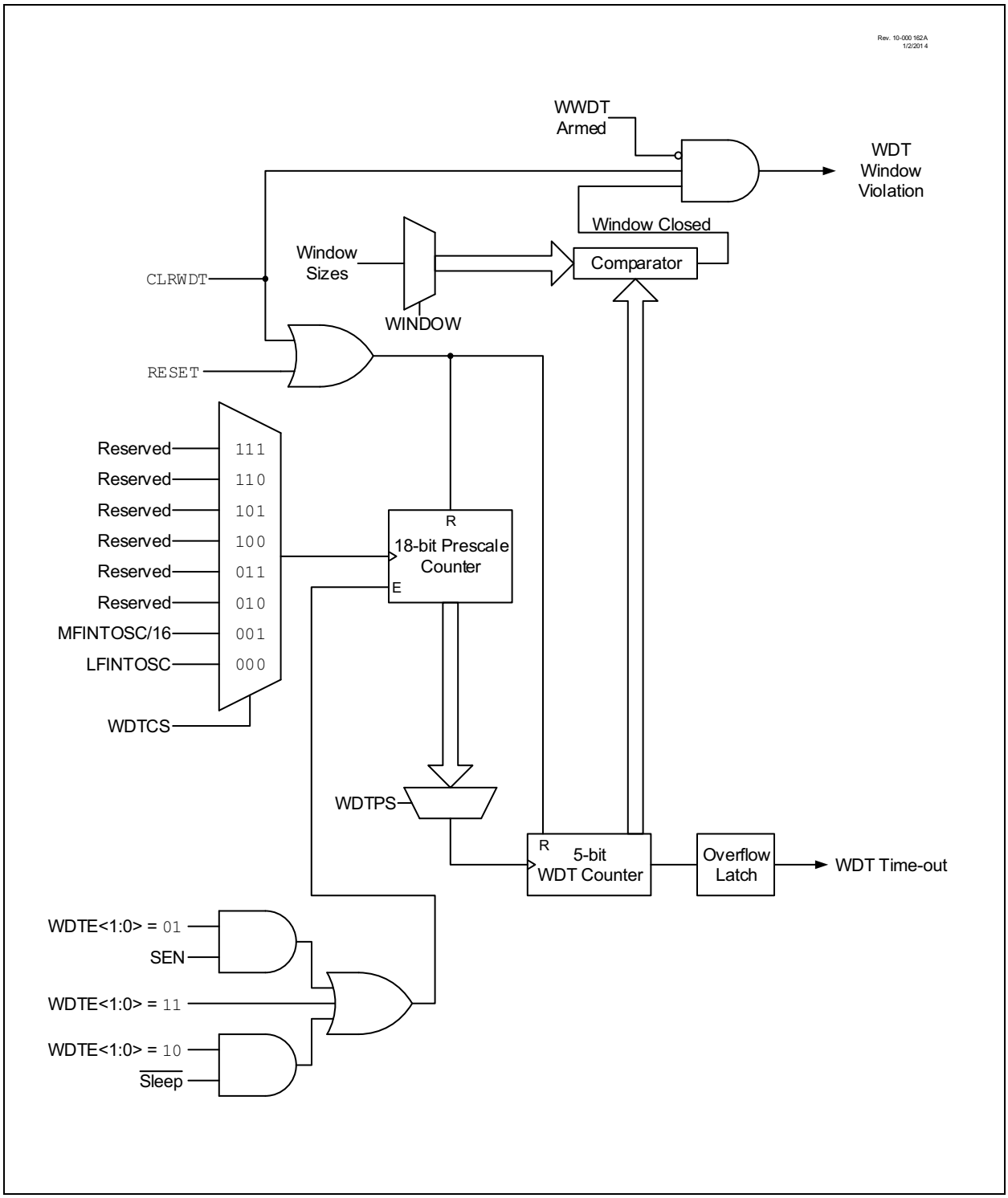
**Note 1:** PIC16F1614/8 only.
**2:** See **Section 35.0 "Electrical Specifications"**.

**TABLE 8-1: SUMMARY OF REGISTERS ASSOCIATED WITH POWER-DOWN MODE**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| INTCON | GIE | PEIE | TMR0IE | INTE | IOCIE | TMR0IF | INTF | IOCIF | 97 |
| IOCAF | — | — | IOCAF5 | IOCAF4 | IOCAF3 | IOCAF2 | IOCAF1 | IOCAF0 | 180 |
| IOCAN | — | — | IOCAN5 | IOCAN4 | IOCAN3 | IOCAN2 | IOCAN1 | IOCAN0 | 180 |
| IOCAP | — | — | IOCAP5 | IOCAP4 | IOCAP3 | IOCAP2 | IOCAP1 | IOCAP0 | 180 |
| IOCCP | IOCCP7(1) | IOCCP6(1) | IOCCP5 | IOCCP4 | IOCCP3 | IOCCP2 | IOCCP1 | IOCCP0 | 180 |
| IOCCN | IOCCN7(1) | IOCCN6(1) | IOCCN5 | IOCCN4 | IOCCN3 | IOCCN2 | IOCCN1 | IOCCN0 | 180 |
| IOCCF | IOCCF7(1) | IOCCF6(1) | IOCCF5 | IOCCF4 | IOCCF3 | IOCCF2 | IOCCF1 | IOCCF0 | 180 |
| PIE1 | TMR1GIE | ADIE | RCIE | TXIE | SSP1IE | CCP1IE | TMR2IE | TMR1IE | 98 |
| PIE2 | — | C2IE | C1IE | — | BCLIE | TMR6IE | TMR4IE | CCP2IE | 99 |
| PIE3 | — | — | CWGIE | ZCDIE | — | — | CLC2IE | CLC1IE | 100 |
| PIE4 | SCANIE | CRCIE | SMT2PWAIE | SMT2PRAIE | SMT2IE | SMT1PWAIE | SMT1PRAIE | SMT1IF | 101 |
| PIE5 | TMR3GIE | TMR3IE | TMR5GIE | TMR5IE | — | AT1IE | PID1EIE | PID1DIE | 102 |
| PIR1 | TMR1GIF | ADIF | RCIF | TXIF | SSP1IF | CCP1IF | TMR2IF | TMR1IF | 103 |
| PIR2 | — | C2IF | C1IF | — | BCLIF | TMR6IF | TMR4IF | CCP2IF | 104 |
| PIR3 | — | — | CWGIF | ZCDIF | — | — | CLC2IF | CLC1IF | 105 |
| PIR4 | SCANIF | CRCIF | SMT2PWAIF | SMT2PRAIF | SMT2IF | SMT1PWAIF | SMT1PRAIF | SMT1IF | 106 |
| PIR5 | TMR3GIF | TMR3IF | TMR5GIF | TMR5IF | — | AT1IF | PID1EIF | PID1DIF | 107 |
| STATUS | — | — | — | $\overline{TO}$ | $\overline{PD}$ | Z | DC | C | 25 |
| WDTCON0 | — | — | WDTPS<4:0> | | | | | SEN | 116 |

**Legend:** — = unimplemented, read as '0'. Shaded cells are not used in Power-Down mode.
**Note 1:** PIC16(L)F1618 only.

**FIGURE 9-1: WATCHDOG TIMER BLOCK DIAGRAM**



## 12.0 I/O PORTS

Each port has six standard registers for its operation. These registers are:

- TRISx registers (data direction)
- PORTx registers (reads the levels on the pins of the device)
- LATx registers (output latch)
- INLVLx (input level control)
- ODCONx registers (open-drain)
- SLRCONx registers (slew rate)

Some ports may have one or more of the following additional registers. These registers are:

- ANSELx (analog select)
- WPUx (weak pull-up)

In general, when a peripheral is enabled on a port pin, that pin cannot be used as a general purpose output. However, the pin can still be read.

**TABLE 12-1: PORT AVAILABILITY PER DEVICE**

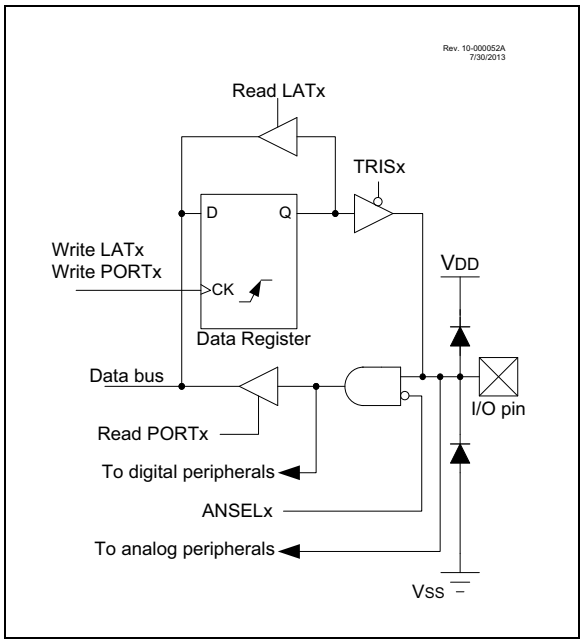
| Device | PORTA | PORTB | PORTC |
|---|---|---|---|
| PIC16(L)F1618 | • | • | • |
| PIC16(L)F1614 | • | | • |

The Data Latch (LATx registers) is useful for read-modify-write operations on the value that the I/O pins are driving.

A write operation to the LATx register has the same effect as a write to the corresponding PORTx register. A read of the LATx register reads of the values held in the I/O PORT latches, while a read of the PORTx register reads the actual I/O pin value.

Ports that support analog inputs have an associated ANSELx register. When an ANSEL bit is set, the digital input buffer associated with that bit is disabled. Disabling the input buffer prevents analog signal levels on the pin between a logic high and low from causing excessive current in the logic input circuitry. A simplified model of a generic I/O port, without the interfaces to other peripherals, is shown in Figure 12-1.

**FIGURE 12-1: GENERIC I/O PORT OPERATION**



**REGISTER 12-13: WPUB: WEAK PULL-UP PORTB REGISTER(1),(2)**

| R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | U-0 | U-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| WPUB7 | WPUB6 | WPUB5 | WPUB4 | — | — | — | — |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| ‘1’ = Bit is set | ‘0’ = Bit is cleared | |

bit 7-4 **WPUB<7:4>**: Weak Pull-up Register bits
1 = Pull-up enabled
0 = Pull-up disabled

bit 3-0 **Unimplemented:** Read as ‘0’

**Note 1:** Global $\overline{\text{WPUEN}}$ bit of the OPTION_REG register must be cleared for individual pull-ups to be enabled.
**2:** The weak pull-up device is automatically disabled if the pin is configured as an output.

**REGISTER 12-14: ODCONB: PORTB OPEN-DRAIN CONTROL REGISTER**

| R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | U-0 | U-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| ODB7 | ODB6 | ODB5 | ODB4 | — | — | — | — |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| ‘1’ = Bit is set | ‘0’ = Bit is cleared | |

bit 7-4 **ODB<7:4>**: PORTB Open-Drain Enable bits
For RB<7:4> pins, respectively
1 = Port pin operates as open-drain drive (sink current only)
0 = Port pin operates as standard push-pull drive (source and sink current)

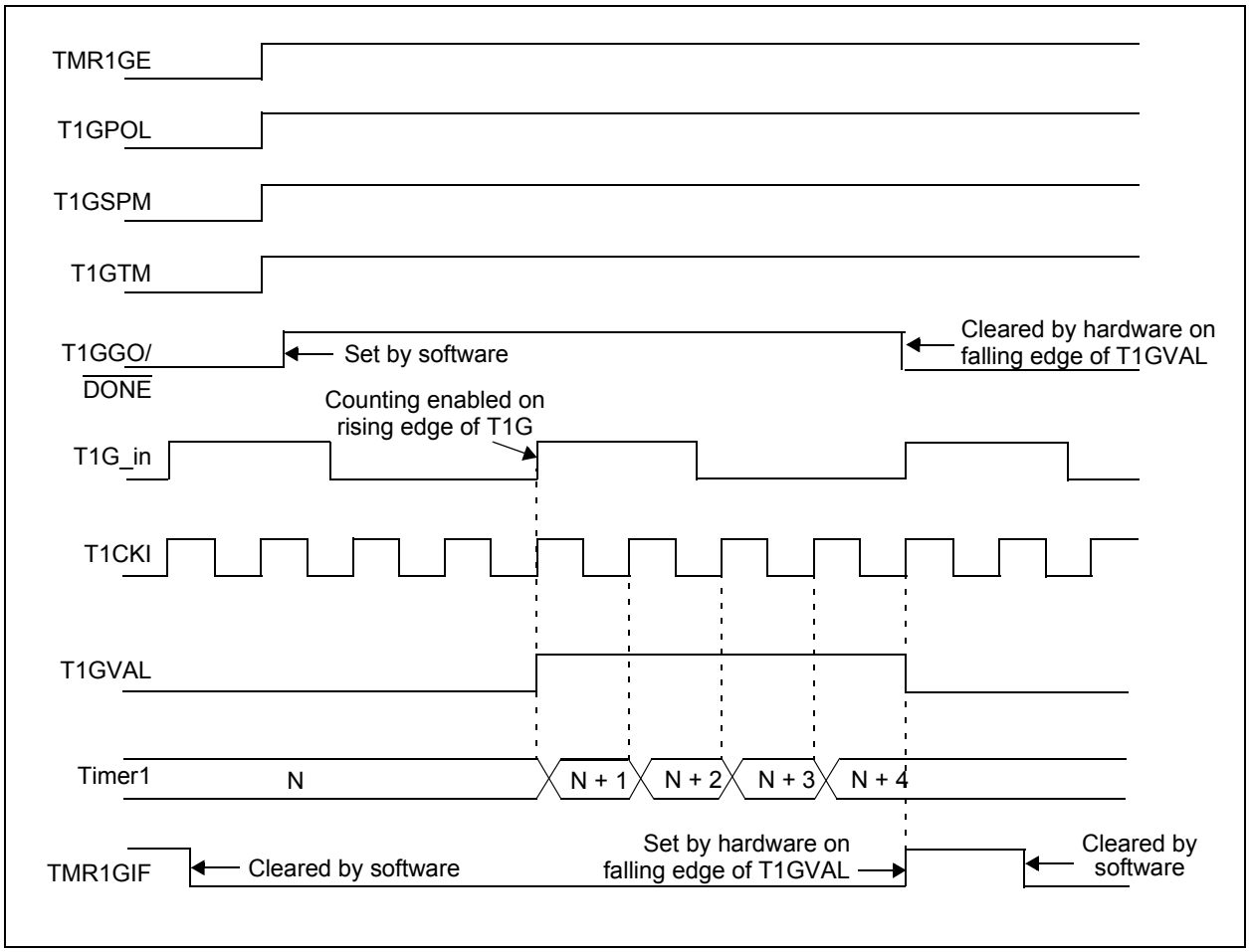
bit 3-0 **Unimplemented:** Read as ‘0’

**TABLE 16-2: SUMMARY OF REGISTERS ASSOCIATED WITH THE TEMPERATURE INDICATOR**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on page |
|---|---|---|---|---|---|---|---|---|---|
| FVRCON | FVREN | FVRRDY | TSEN | TSRNG | CDAFVR<1:0> | | ADFVR<1:0> | | 118 |

**Legend:** Shaded cells are unused by the temperature indicator module.

**FIGURE 22-6: TIMER1 GATE SINGLE-PULSE AND TOGGLE COMBINED MODE**



### 23.5.6 EDGE-TRIGGERED ONE-SHOT MODE

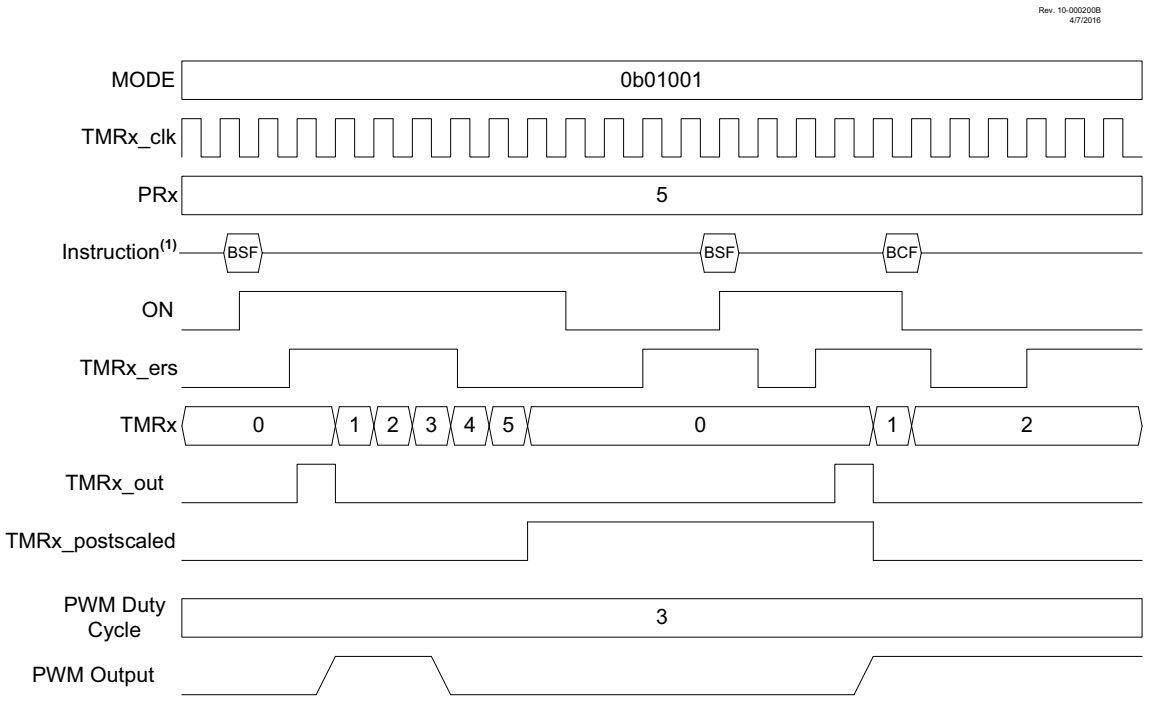
The Edge-Triggered One-Shot modes start the timer on an edge from the external signal input, after the ON bit is set, and clear the ON bit when the timer matches the PRx period value. The following edges will start the timer:

- Rising edge (MODE<4:0> = 01001)
- Falling edge (MODE<4:0> = 01010)
- Rising or Falling edge (MODE<4:0> = 01011)

If the timer is halted by clearing the ON bit then another TMRx_ers edge is required after the ON bit is set to resume counting. Figure 23-9 illustrates operation in the rising edge One-Shot mode.

When Edge-Triggered One-Shot mode is used in conjunction with the CCP then the edge-trigger will activate the PWM drive and the PWM drive will deactivate when the timer matches the CCPRx pulse width value and stay deactivated when the timer halts at the PRx period count match.

**FIGURE 23-9: EDGE-TRIGGERED ONE-SHOT MODE TIMING DIAGRAM (MODE = 01001)**




Note 1: BSF and BCF represent Bit-Set File and Bit-Clear File instructions executed by the CPU to set or clear the ON bit of TxCON. CPU execution is asynchronous to the timer clock input.

### 23.5.8 LEVEL RESET, EDGE-TRIGGERED HARDWARE LIMIT ONE-SHOT MODES

In Level -Triggered One-Shot mode the timer count is reset on the external signal level and starts counting on the rising/falling edge of the transition from Reset level to the active level while the ON bit is set. Reset levels are selected as follows:

- Low Reset level (MODE<4:0> = 01110)
- High Reset level (MODE<4:0> = 01111)

When the timer count matches the PRx period count, the timer is reset and the ON bit is cleared. When the ON bit is cleared by either a PRx match or by software control a new external signal edge is required after the ON bit is set to start the counter.

When Level-Triggered Reset One-Shot mode is used in conjunction with the CCP PWM operation the PWM drive goes active with the external signal edge that starts the timer. The PWM drive goes inactive when the timer count equals the CCPRx pulse width count. The PWM drive does not go active when the timer count clears at the PRx period count match.

## 23.6 Timer2 Operation During Sleep

When PSYNC = 1, Timer2 cannot be operated while the processor is in Sleep mode. The contents of the TMR2 and T2PR registers will remain unchanged while processor is in Sleep mode.

When PSYNC = 0, Timer2 will operate in Sleep as long as the clock source selected is also still running. Selecting the LFINTOSC, MFINTOSC, or HFINTOSC oscillator as the timer clock source will keep the selected oscillator running during Sleep.

## 24.4 I²C MODE OPERATION

All MSSP I²C communication is byte oriented and shifted out MSb first. Six SFR registers and two interrupt flags interface the module with the PIC® microcontroller and user software. Two pins, SDA and SCL, are exercised by the module to communicate with other external I²C devices.

### 24.4.1 BYTE FORMAT

All communication in I²C is done in 9-bit segments. A byte is sent from a master to a slave or vice-versa, followed by an Acknowledge bit sent back. After the eighth falling edge of the SCL line, the device outputting data on the SDA changes that pin to an input and reads in an acknowledge value on the next clock pulse.

The clock signal, SCL, is provided by the master. Data is valid to change while the SCL signal is low, and sampled on the rising edge of the clock. Changes on the SDA line while the SCL line is high define special conditions on the bus, explained below.

### 24.4.2 DEFINITION OF I²C TERMINOLOGY

There is language and terminology in the description of I²C communication that have definitions specific to I²C. That word usage is defined below and may be used in the rest of this document without explanation. This table was adapted from the Philips I²C specification.

### 24.4.3 SDA AND SCL PINS

Selection of any I²C mode with the SSPEN bit set, forces the SCL and SDA pins to be open-drain. These pins should be set by the user to inputs by setting the appropriate TRIS bits.

**Note 1:** Data is tied to output zero when an I²C mode is enabled.

**2:** Any device pin can be selected for SDA and SCL functions with the PPS peripheral. These functions are bidirectional. The SDA input is selected with the SSPDATPPS registers. The SCL input is selected with the SSPCLKPPS registers. Outputs are selected with the RxyPPS registers. It is the user's responsibility to make the selections so that both the input and the output for each function is on the same pin.

### 24.4.4 SDA HOLD TIME

The hold time of the SDA pin is selected by the SDAHT bit of the SSPxCON3 register. Hold time is the time SDA is held valid after the falling edge of SCL. Setting the SDAHT bit selects a longer 300 ns minimum hold time and may help on buses with large capacitance.

**TABLE 24-2: I²C BUS TERMS**

| TERM | Description |
|---|---|
| Transmitter | The device which shifts data out onto the bus. |
| Receiver | The device which shifts data in from the bus. |
| Master | The device that initiates a transfer, generates clock signals and terminates a transfer. |
| Slave | The device addressed by the master. |
| Multi-master | A bus with more than one device that can initiate data transfers. |
| Arbitration | Procedure to ensure that only one master at a time controls the bus. Winning arbitration ensures that the message is not corrupted. |
| Synchronization | Procedure to synchronize the clocks of two or more devices on the bus. |
| Idle | No master is controlling the bus, and both SDA and SCL lines are high. |
| Active | Any time one or more master devices are controlling the bus. |
| Addressed Slave | Slave device that has received a matching address and is actively being clocked by a master. |
| Matching Address | Address byte that is clocked into a slave that matches the value stored in SSPxADD. |
| Write Request | Slave receives a matching address with R/$\overline{W}$ bit clear, and is ready to clock in data. |
| Read Request | Master sends an address byte with the R/$\overline{W}$ bit set, indicating that it wishes to clock data out of the Slave. This data is the next and all following bytes until a Restart or Stop. |
| Clock Stretching | When a device on the bus holds SCL low to stall communication. |
| Bus Collision | Any time the SDA line is sampled low by the module while it is outputting and expected high state. |

#### 24.6.13.3 Bus Collision During a Stop Condition

Bus collision occurs during a Stop condition if:

a) After the SDA pin has been deasserted and allowed to float high, SDA is sampled low after the BRG has timed out (Case 1).

b) After the SCL pin is deasserted, SCL is sampled low before SDA goes high (Case 2).

The Stop condition begins with SDA asserted low. When SDA is sampled low, the SCL pin is allowed to float. When the pin is sampled high (clock arbitration), the Baud Rate Generator is loaded with SSPxADD and counts down to zero. After the BRG times out, SDA is sampled. If SDA is sampled low, a bus collision has occurred. This is due to another master attempting to drive a data '0' (Figure 24-38). If the SCL pin is sampled low before SDA is allowed to float high, a bus collision occurs. This is another case of another master attempting to drive a data '0' (Figure 24-39).

**FIGURE 24-38: BUS COLLISION DURING A STOP CONDITION (CASE 1)**

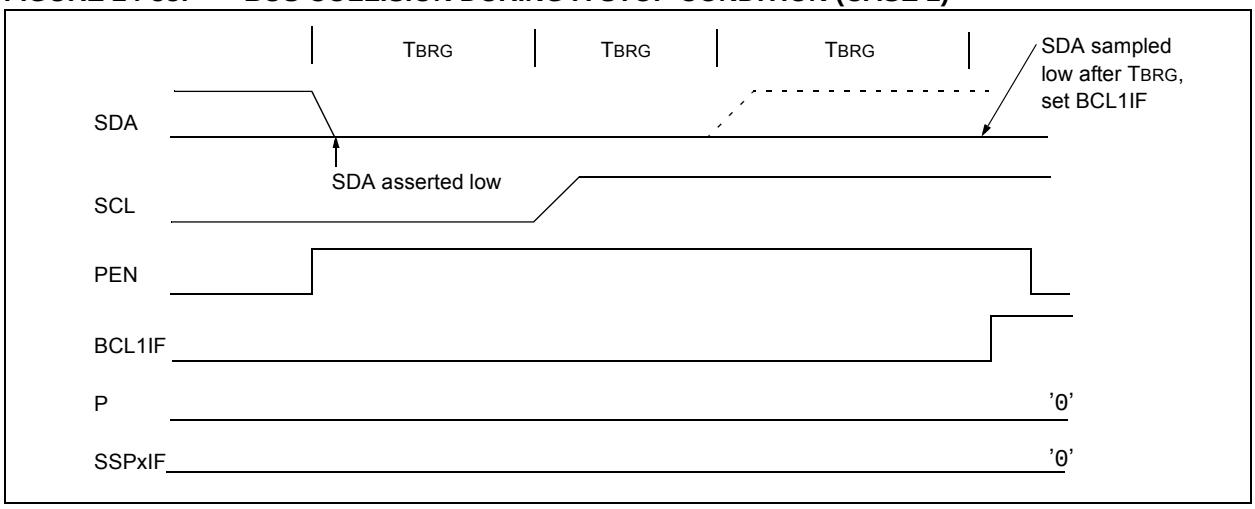



**FIGURE 24-39: BUS COLLISION DURING A STOP CONDITION (CASE 2)**

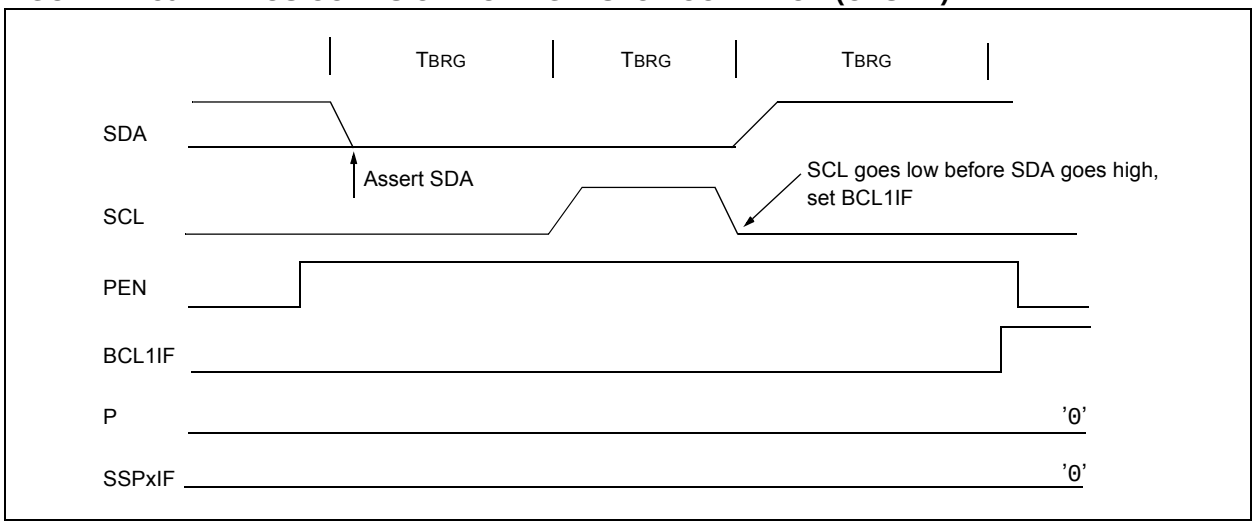
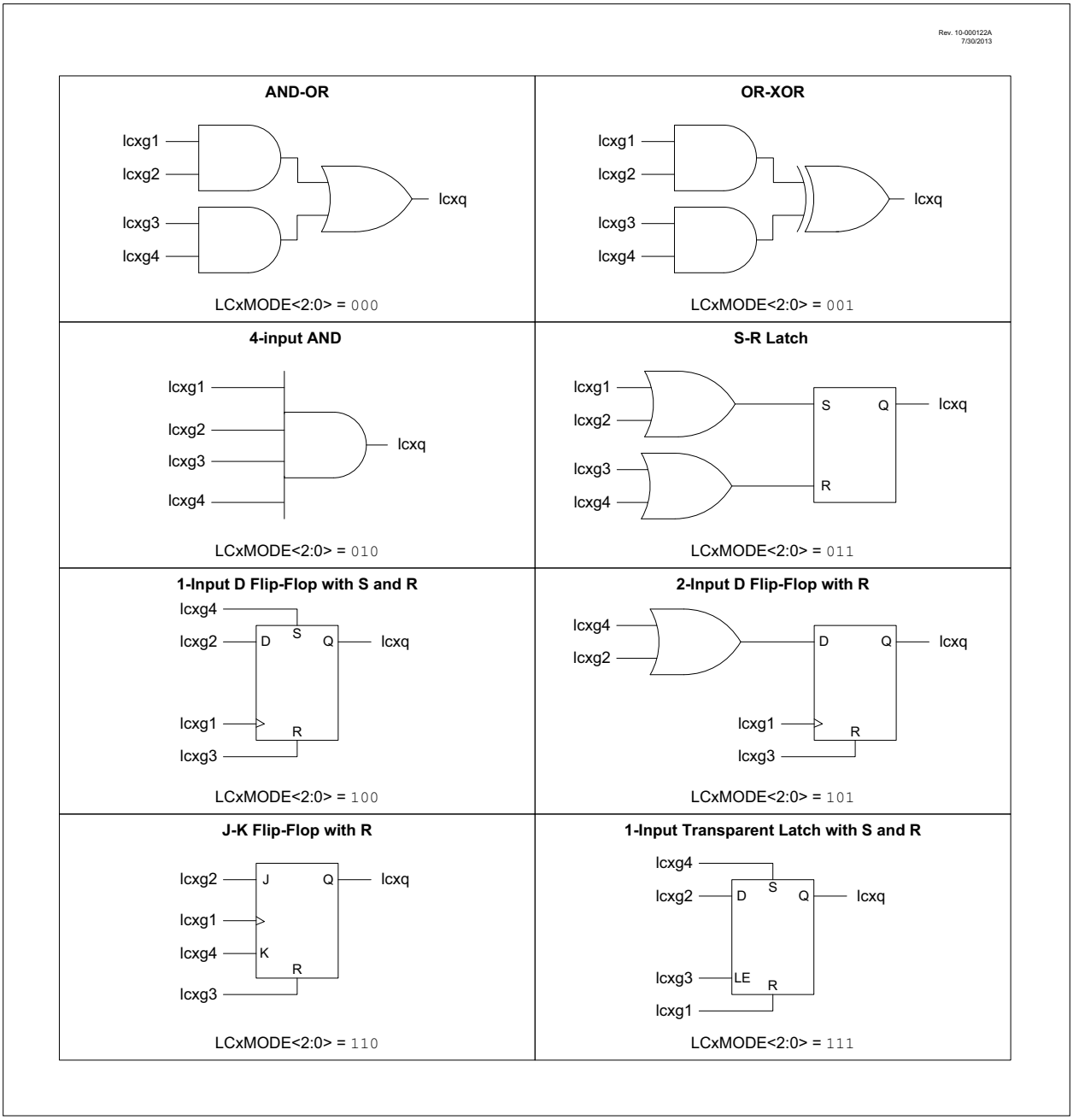


**FIGURE 29-3: PROGRAMMABLE LOGIC FUNCTIONS**



**REGISTER 31-16: ATxIR1: ANGULAR TIMER INTERRUPT FLAG 1 REGISTER**

| U-0 | U-0 | U-0 | U-0 | U-0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| — | — | — | — | — | CC3IF | CC2IF | CC1IF |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | q = Value depends on condition |

bit 7-3 **Unimplemented:** Read as '0'

bit 2 **CC3IF:** Capture/Compare Interrupt 3 Flag bit
If CC3MODE = 1 (Capture)
1 = Capture interrupt 3 has occurred; captured phase value is in ATxCC3
0 = Capture interrupt 3 has not occurred, or has been cleared
If CC3MODE = 0 (Compare)
1 = Compare interrupt 3 has occurred
0 = Compare interrupt 3 has not occurred, or has been cleared

bit 1 **CC2IF:** Capture/Compare Interrupt 2 Flag bit
If CC2MODE = 1 (Capture)
1 = Capture interrupt 2 has occurred; captured phase value is in ATxCC2
0 = Capture interrupt 2 has not occurred, or has been cleared
If CC2MODE = 0 (Compare)
1 = Compare interrupt 2 has occurred
0 = Compare interrupt 2 has not occurred, or has been cleared

bit 0 **CC1IF:** Capture/Compare Interrupt 1 Flag bit
If CC1MODE = 1 (Capture)
1 = Capture interrupt 1 has occurred; captured phase value is in ATxCC1
0 = Capture interrupt 1 has not occurred, or has been cleared
If CC1MODE = 0 (Compare)
1 = Compare interrupt 1 has occurred
0 = Compare interrupt 1 has not occurred, or has been cleared

**REGISTER 32-12: PIDxOUTU: PID OUTPUT UPPER REGISTER**

| U-0 | U-0 | U-0 | U-0 | U-0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| — | — | — | — | — | OUT<34:32> | | |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| ‘1’ = Bit is set | ‘0’ = Bit is cleared | q = Value depends on condition |

bit 7-3 **Unimplemented:** Read as ‘0’

bit 2-0 **OUT<34:32>:** Bits <34:32> of OUT. OUT is the output value of the PID after completing the designated calculation on the specified inputs.

**REGISTER 32-13: PIDxOUTHH: PID OUTPUT HIGH HIGH REGISTER**

| R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| OUT<31:24> | | | | | | | |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| ‘1’ = Bit is set | ‘0’ = Bit is cleared | q = Value depends on condition |

bit 7-0 **OUT<31:24>:** Bits <31:24> of OUT. OUT is the output value of the PID after completing the designated calculation on the specified inputs.

**Note:** Unless otherwise noted, VIN = 5V, FOSC = 500 kHz, CIN = 0.1 µF, TA = 25°C.

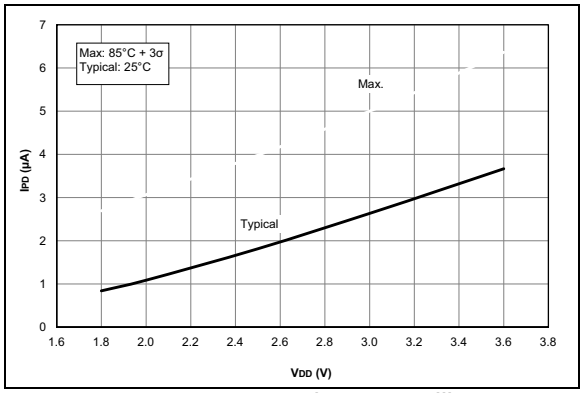



***FIGURE 36-31:*** *IPD, Timer1 Oscillator, FOSC = 32 kHz, PIC16LF1614/8 Only.*

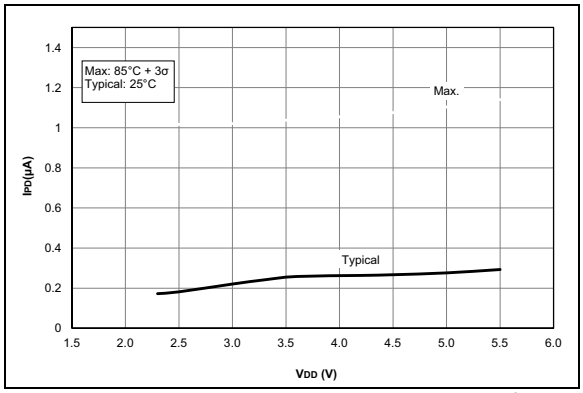



***FIGURE 36-34:*** *IPD, ADC Non-Converting, PIC16F1614/8 Only.*

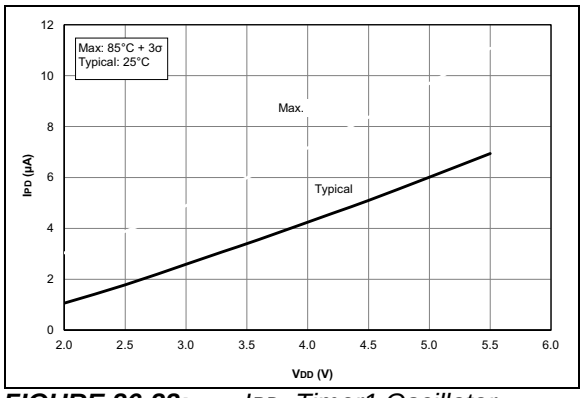



***FIGURE 36-32:*** *IPD, Timer1 Oscillator, FOSC = 32 kHz, PIC16F1614/8 Only.*

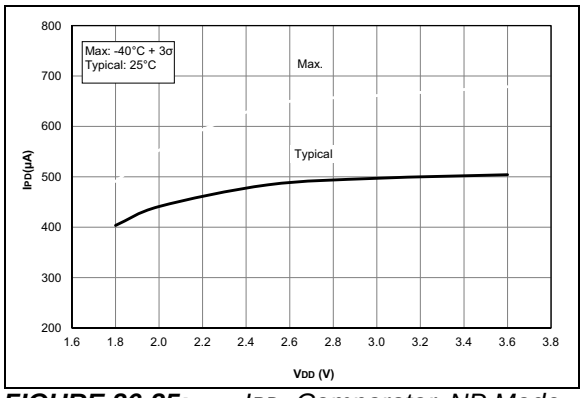



***FIGURE 36-35:*** *IPD, Comparator, NP Mode (CxSP = 1), PIC16LF1614/8 Only.*

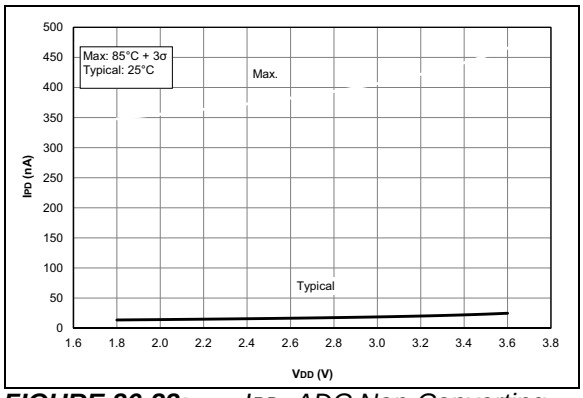



***FIGURE 36-33:*** *IPD, ADC Non-Converting, PIC16LF1614/8 Only.*

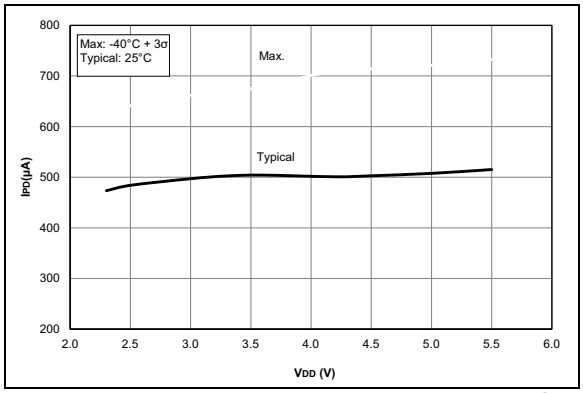
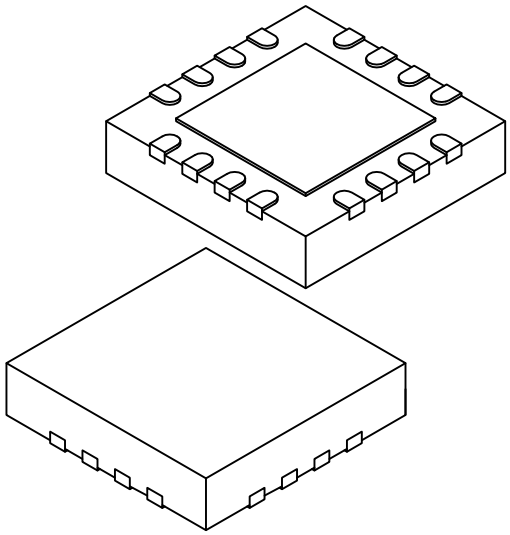



***FIGURE 36-36:*** *IPD, Comparator, NP Mode (CxSP = 1), PIC16F1614/8 Only.*

### 16-Lead Plastic Quad Flat, No Lead Package (ML) - 4x4x0.9mm Body [QFN]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



| Units | | MILLIMETERS | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Number of Pins | N | | 16 | |
| Pitch | e | | 0.65 BSC | |
| Overall Height | A | 0.80 | 0.90 | 1.00 |
| Standoff | A1 | 0.00 | 0.02 | 0.05 |
| Contact Thickness | A3 | | 0.20 REF | |
| Overall Width | E | | 4.00 BSC | |
| Exposed Pad Width | E2 | 2.50 | 2.65 | 2.80 |
| Overall Length | D | | 4.00 BSC | |
| Exposed Pad Length | D2 | 2.50 | 2.65 | 2.80 |
| Contact Width | b | 0.25 | 0.30 | 0.35 |
| Contact Length | L | 0.30 | 0.40 | 0.50 |
| Contact-to-Exposed Pad | K | 0.20 | - | - |

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated
3. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-127D Sheet 2 of 2