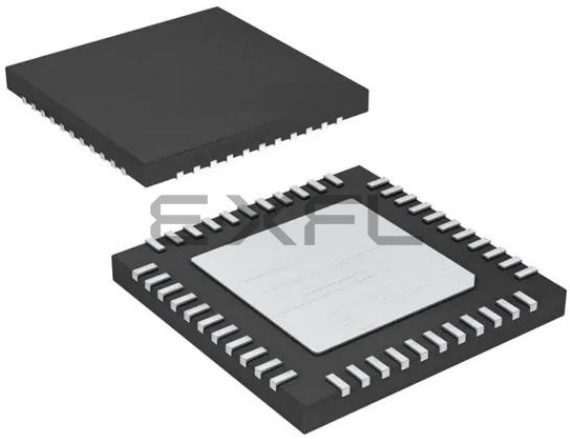


**Welcome to E-XFL.COM**

## 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 | 64MHz |
| Connectivity | I²C, LINbus, SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, LVD, POR, PWM, WDT |
| Number of I/O | 36 |
| Program Memory Size | 64KB (32K x 16) |
| Program Memory Type | FLASH |
| EEPROM Size | 1K x 8 |
| RAM Size | 3.6K x 8 |
| Voltage - Supply (Vcc/Vdd) | 1.8V ~ 3.6V |
| Data Converters | A/D 35x10b; D/A 1x5b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 44-VQFN Exposed Pad |
| Supplier Device Package | 44-QFN (8x8) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic18lf46k40-i-ml |

## 3.0 DEVICE CONFIGURATION

Device configuration consists of Configuration Words, Code Protection, Device ID and Rev ID.

### 3.1 Configuration Words

There are six Configuration Word bits that allow the user to setup the device with several choices of oscillators, Resets and memory protection options. These are implemented as Configuration Word 1 through Configuration Word 6 at 300000h through 30000Bh.

| | |
|---|---|
| **Note:** | The $\overline{\text{DEBUG}}$ bit in Configuration Words is managed automatically by device development tools including debuggers and programmers. For normal device operation, this bit should be maintained as a '1'. |

**REGISTER 3-9: Configuration Word 5L (30 0008h): Code Protection**

| U-1 | U-1 | U-1 | U-1 | U-1 | U-1 | R/W-1 | R/W-1 |
|---|---|---|---|---|---|---|---|
| — | — | — | — | — | — | $\overline{CPD}$ | $\overline{CP}$ |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '1' | |
| -n = Value for blank device | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

bit 7-2 **Unimplemented:** Read as '1'

bit 1 **$\overline{CPD}$:** Data NVM Memory Code Protection bit
- 1 = Data NVM code protection disabled
- 0 = Data NVM code protection enabled

bit 0 **$\overline{CP}$:** User NVM Program Memory Code Protection bit
- 1 = User NVM code protection disabled
- 0 = User NVM code protection enabled

**REGISTER 3-10: Configuration Word 6L (30 000Ah): Memory Read Protection**

| R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |
|---|---|---|---|---|---|---|---|
| EBTR7 | EBTR6 | EBTR5 | EBTR4 | EBTR3 | EBTR2 | EBTR1 | EBTR0 |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '1' | |
| -n = Value for blank device | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

bit 7-0 **EBTR<7:0>:** Table Read Protection bits(1)
- 1 = Corresponding Memory Block NOT protected from table reads executed in other blocks
- 0 = Corresponding Memory Block protected from table reads executed in other blocks

**Note 1:** Refer to Table 10-2 for details on implementation of the individual EBTR bits.

**REGISTER 3-11: Configuration Word 6H (30 000Bh): Memory Read Protection**

| U-1 | U-1 | U-1 | U-1 | U-1 | U-1 | R/W-1 | U-1 |
|---|---|---|---|---|---|---|---|
| — | — | — | — | — | — | EBTRB | — |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '1' | |
| -n = Value for blank device | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

bit 7-2 **Unimplemented:** Read as '1'

bit 1 **EBTRB:** Table Read Protection bit
- 1 = Memory Boot Block NOT protected from table reads executed in other blocks
- 0 = Memory Boot Block protected from table reads executed in other blocks

bit 0 **Unimplemented:** Read as '1'

### 10.4.2 ACCESS BANK

While the use of the BSR with an embedded 8-bit address allows users to address the entire range of data memory, it also means that the user must always ensure that the correct bank is selected. Otherwise, data may be read from or written to the wrong location. This can be disastrous if a GPR is the intended target of an operation, but an SFR is written to instead. Verifying and/or changing the BSR for each read or write to data memory can become very inefficient.

To streamline access for the most commonly used data memory locations, the data memory is configured with an Access Bank, which allows users to access a mapped block of memory without specifying a BSR. The Access Bank consists of the first 96 bytes of memory (00h-5Fh) in Bank 0 and the last 160 bytes of memory (60h-FFh) in Block 15. The lower half is known as the “Access RAM” and is composed of GPRs. This upper half is also where the device’s SFRs are mapped. These two areas are mapped contiguously in the Access Bank and can be addressed in a linear fashion by an 8-bit address (Figure 10-4).

The Access Bank is used by core PIC18 instructions that include the Access RAM bit (the ‘a’ parameter in the instruction). When ‘a’ is equal to ‘1’, the instruction uses the BSR and the 8-bit address included in the opcode for the data memory address. When ‘a’ is ‘0’, however, the instruction is forced to use the Access Bank address map; the current value of the BSR is ignored entirely.

Using this “forced” addressing allows the instruction to operate on a data address in a single cycle, without updating the BSR first. For 8-bit addresses of 60h and above, this means that users can evaluate and operate on SFRs more efficiently. The Access RAM below 60h is a good place for data values that the user might need to access rapidly, such as immediate computational results or common program variables. Access RAM also allows for faster and more code efficient context saving and switching of variables.

The mapping of the Access Bank is slightly different when the extended instruction set is enabled (XINST Configuration bit = 1). This is discussed in more detail in **Section 10.7.3 “Mapping the Access Bank in Indexed Literal Offset Mode”**.

### 10.4.3 GENERAL PURPOSE REGISTER FILE

PIC18 devices may have banked memory in the GPR area. This is data RAM, which is available for use by all instructions. GPRs start at the bottom of Bank 0 (address 000h) and grow upwards towards the bottom of the SFR area. GPRs are not initialized by a Power-on Reset and are unchanged on all other Resets.

### 10.4.4 SPECIAL FUNCTION REGISTERS

The Special Function Registers (SFRs) are registers used by the CPU and peripheral modules for controlling the desired operation of the device. These registers are implemented as static RAM. SFRs start at the top of data memory (FFFh) and extend downward to occupy the top portion of Bank 15 (F38h to FFFh). A list of these registers is given in Table 10-3 and Table 10-4.

The SFRs can be classified into two sets: those associated with the “core” device functionality (ALU, Resets and interrupts) and those related to the peripheral functions. The Reset and Interrupt registers are described in their respective chapters, while the ALU’s STATUS register is described later in this section. Registers related to the operation of a peripheral feature are described in the chapter for that peripheral.

The SFRs are typically distributed among the peripherals whose functions they control. Unused SFR locations are unimplemented and read as ‘0’s.

## 14.8 Register Definitions: Interrupt Control

**REGISTER 14-1: INTCON: INTERRUPT CONTROL REGISTER**

| R/W-0/0 | R/W-0/0 | R/W-0/0 | U-0 | U-0 | R/W-1/1 | R/W-1/1 | R/W-1/1 |
|---|---|---|---|---|---|---|---|
| GIE/GIEH | PEIE/GIEL | IPEN | — | — | INT2EDG | INT1EDG | INT0EDG |
| bit 7 | | | | | | | bit 0 |

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

bit 7 **GIE/GIEH:** Global Interrupt Enable bit

If IPEN = 1:
- 1 = Enables all unmasked interrupts and cleared by hardware for high-priority interrupts only
- 0 = Disables all interrupts

If IPEN = 0:
- 1 = Enables all unmasked interrupts and cleared by hardware for all interrupts
- 0 = Disables all interrupts

bit 6 **PEIE/GIEL:** Peripheral Interrupt Enable bit

If IPEN = 1:
- 1 = Enables all low-priority interrupts and cleared by hardware for low-priority interrupts only
- 0 = Disables all low-priority interrupts

If IPEN = 0:
- 1 = Enables all unmasked peripheral interrupts
- 0 = Disables all peripheral interrupts

bit 5 **IPEN:** Interrupt Priority Enable bit
- 1 = Enable priority levels on interrupts
- 0 = Disable priority levels on interrupts

bit 4-3 **Unimplemented**: Read as '0'

bit 2 **INT2EDG:** External Interrupt 2 Edge Select bit
- 1 = Interrupt on rising edge of INT2 pin
- 0 = Interrupt on falling edge of INT2 pin

bit 1 **INT1EDG:** External Interrupt 1 Edge Select bit
- 1 = Interrupt on rising edge of INT1 pin
- 0 = Interrupt on falling edge of INT1 pin

bit 0 **INT0EDG:** External Interrupt 0 Edge Select bit
- 1 = Interrupt on rising edge of INT0 pin
- 0 = Interrupt on falling edge of INT0 pin

**Note:** Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the global enable bit. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt. This feature allows for software polling.

**REGISTER 14-22: IPR4: PERIPHERAL INTERRUPT PRIORITY REGISTER 4**

| U-0 | U-0 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 |
|---|---|---|---|---|---|---|---|
| — | — | TMR6IP | TMR5IP | TMR4IP | TMR3IP | TMR2IP | TMR1IP |
| bit 7 | | | | | | | bit 0 |

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

bit 7-6 **Unimplemented:** Read as '0'

bit 5 **TMR6IP:** TMR6 to PR6 Match Interrupt Priority bit
1 = High priority
0 = Low priority

bit 4 **TMR5IP:** TMR5 Overflow Interrupt Priority bit
1 = High priority
0 = Low priority

bit 3 **TMR4IP:** TMR4 to PR4 Match Interrupt Priority bit
1 = High priority
0 = Low priority

bit 2 **TMR3IP:** TMR3 Overflow Interrupt Priority bit
1 = High priority
0 = Low priority

bit 1 **TMR2IP:** TMR2 to PR2 Match Interrupt Priority bit
1 = High priority
0 = Low priority

bit 0 **TMR1IP:** TMR1 Overflow Interrupt Priority bit
1 = High priority
0 = Low priority

**REGISTER 18-2: T0CON1: TIMER0 CONTROL REGISTER 1**

| 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 |
|---|---|---|---|---|---|---|---|
| T0CS<2:0> | | | T0ASYNC | T0CKPS<3:0> | | | |
| bit 7 | | | | | | | bit 0 |

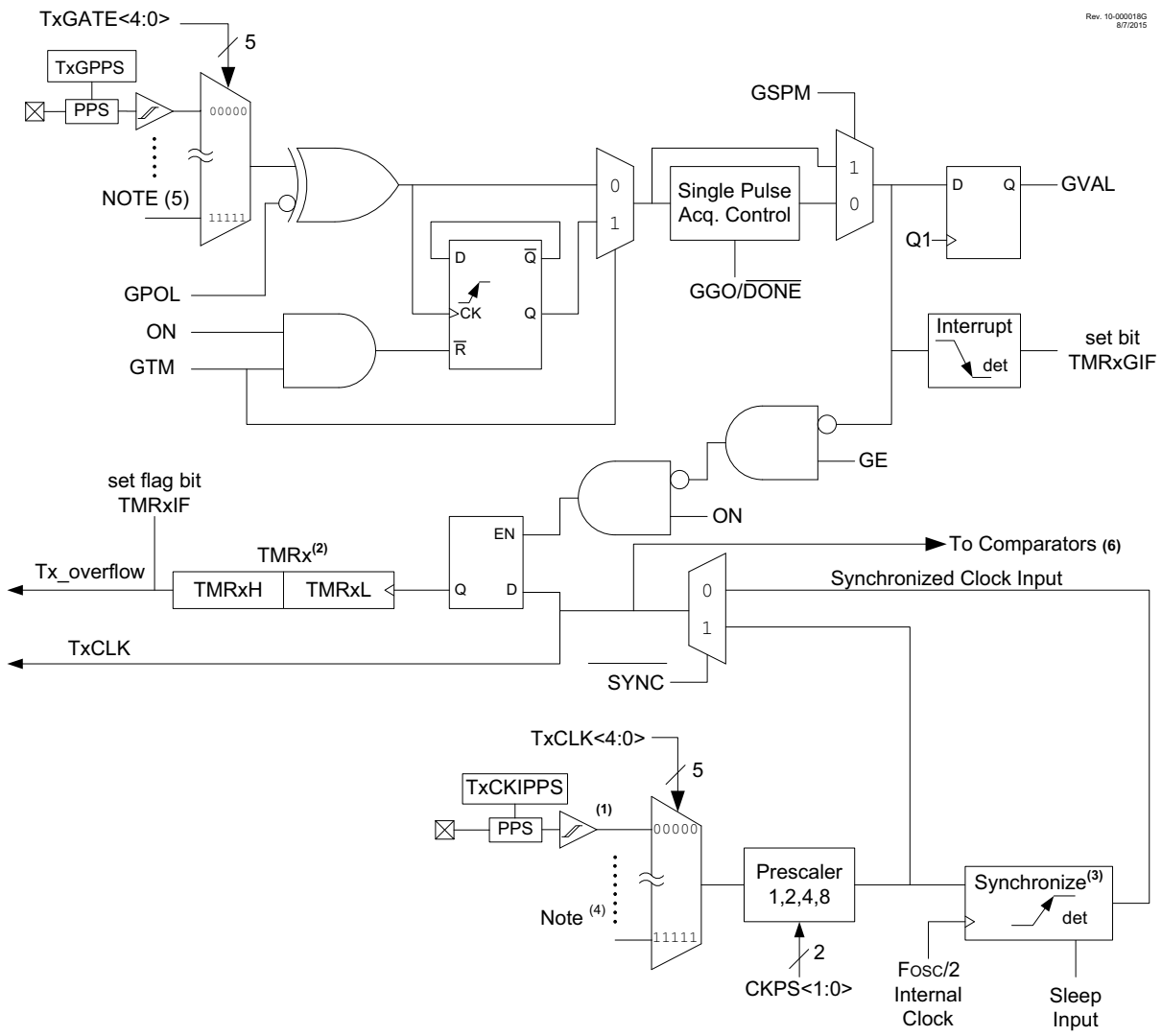
| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| ‘1’ = Bit is set | ‘0’ = Bit is cleared | |

bit 7-5 **T0CS<2:0>:** Timer0 Clock Source Select bits
111 = Reserved
110 = Reserved
101 = SOSC
100 = LFINTOSC
011 = HFINTOSC
010 = FOSC/4
001 = Pin selected by T0CKIPPS (Inverted)
000 = Pin selected by T0CKIPPS (Non-inverted)

bit 4 **T0ASYNC:** TMR0 Input Asynchronization Enable bit
1 = The input to the TMR0 counter is not synchronized to system clocks
0 = The input to the TMR0 counter is synchronized to FOSC/4

bit 3-0 **T0CKPS<3:0>:** Prescaler Rate Select bit
1111 = 1:32768
1110 = 1:16384
1101 = 1:8192
1100 = 1:4096
1011 = 1:2048
1010 = 1:1024
1001 = 1:512
1000 = 1:256
0111 = 1:128
0110 = 1:64
0101 = 1:32
0100 = 1:16
0011 = 1:8
0010 = 1:4
0001 = 1:2
0000 = 1:1

**FIGURE 19-1: TIMER1/3/5 BLOCK DIAGRAM**




**Note 1:** ST Buffer is high speed type when using TxCKIPPS.

**2:** TMRx register increments on rising edge.

**3:** Synchronize does not operate while in Sleep.

**4:** See Register 19-3 for clock source selections.

**5:** See Register 19-4 for gate source selection.

**6:** Synchronized comparator output should not be used in conjunction with synchronized input clock.

**REGISTER 19-3: TMRxCLK: TIMERx CLOCK REGISTER**

| U-0 | U-0 | U-0 | U-0 | R/W-0/u | R/W-0/u | R/W-0/u | R/W-0/u |
|---|---|---|---|---|---|---|---|
| — | — | — | — | CS<3:0> | | | |
| 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 | u = unchanged |

bit 7-4 **Unimplemented:** Read as ‘0’

bit 3-0 **CS<3:0>:** Timerx Clock Source Selection bits

| CS | Timer1 | Timer3 | Timer5 |
|---|---|---|---|
| | **Clock Source** | **Clock Source** | **Clock Source** |
| 1111-1100 | Reserved | Reserved | Reserved |
| 1011 | TMR5 overflow | TMR5 overflow | Reserved |
| 1010 | TMR3 overflow | Reserved | TMR3 overflow |
| 1001 | Reserved | TMR1 overflow | TMR1 overflow |
| 1000 | TMR0 overflow | TMR0 overflow | TMR0 overflow |
| 0111 | CLKREF | CLKREF | CLKREF |
| 0110 | SOSC | SOSC | SOSC |
| 0101 | MFINTOSC (500 kHz) | MFINTOSC (500 kHz) | MFINTOSC (500 kHz) |
| 0100 | LFINTOSC | LFINTOSC | LFINTOSC |
| 0011 | HFINTOSC | HFINTOSC | HFINTOSC |
| 0010 | Fosc | Fosc | Fosc |
| 0001 | Fosc/4 | Fosc/4 | Fosc/4 |
| 0000 | T1CKIPPS | T3CKIPPS | T5CKIPPS |

## 20.6 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.

### 21.4.1 CCPx PIN CONFIGURATION

The software must configure the CCPx pin as an output by clearing the associated TRIS bit and defining the appropriate output pin through the RxyPPS registers. See **Section 17.0 "Peripheral Pin Select (PPS) Module"** for more details.

The CCP output can also be used as an input for other peripherals.

| | |
|---|---|
| **Note:** | Clearing the CCPxCON register will force the CCPx compare output latch to the default low level. This is not the PORT I/O data latch. |

### 21.4.2 TIMER1 MODE RESOURCE

In Compare mode, Timer1 must be running in either Timer mode or Synchronized Counter mode. The compare operation may not work in Asynchronous Counter mode.

See **Section 19.0 "Timer1/3/5 Module with Gate Control"** for more information on configuring Timer1.

| | |
|---|---|
| **Note:** | Clocking Timer1 from the system clock (FOSC) should not be used in Compare mode. In order for Compare mode to recognize the trigger event on the CCPx pin, TImer1 must be clocked from the instruction clock (FOSC/4) or from an external clock source. |

### 21.4.3 AUTO-CONVERSION TRIGGER

All CCPx modes set the CCP interrupt flag (CCPxIF). When this flag is set and a match occurs, an auto-conversion trigger can take place if the CCP module is selected as the conversion trigger source.

Refer to **Section 31.2.5 "Auto-Conversion Trigger"** for more information.

| | |
|---|---|
| **Note:** | Removing the match condition by changing the contents of the CCPRxH and CCPRxL register pair, between the clock edge that generates the Auto-conversion Trigger and the clock edge that generates the Timer1 Reset, will preclude the Reset from occurring |

### 21.4.4 COMPARE DURING SLEEP

Since FOSC is shut down during Sleep mode, the Compare mode will not function properly during Sleep, unless the timer is running. The device will wake on interrupt (if enabled).

## 21.5 PWM Overview

Pulse-Width Modulation (PWM) is a scheme that provides power to a load by switching quickly between fully on and fully off states. The PWM signal resembles a square wave where the high portion of the signal is considered the ON state and the low portion of the signal is considered the OFF state. The high portion, also known as the pulse width, can vary in time and is defined in steps. A larger number of steps applied, which lengthens the pulse width, also supplies more power to the load. Lowering the number of steps applied, which shortens the pulse width, supplies less power. The PWM period is defined as the duration of one complete cycle or the total amount of on and off time combined.

PWM resolution defines the maximum number of steps that can be present in a single PWM period. A higher resolution allows for more precise control of the pulse-width time and in turn the power that is applied to the load.

The term duty cycle describes the proportion of the on time to the off time and is expressed in percentages, where 0% is fully off and 100% is fully on. A lower duty cycle corresponds to less power applied and a higher duty cycle corresponds to more power applied.

Figure 21-3 shows a typical waveform of the PWM signal.

### 21.5.1 STANDARD PWM OPERATION

The standard PWM function described in this section is available and identical for all CCP modules.

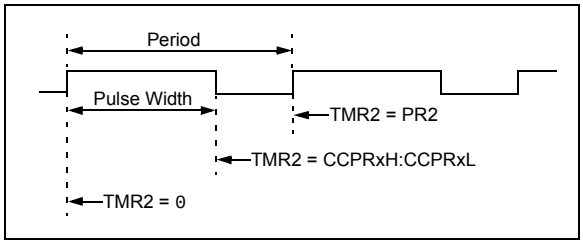
The standard PWM mode generates a Pulse-Width Modulation (PWM) signal on the CCPx pin with up to ten bits of resolution. The period, duty cycle, and resolution are controlled by the following registers:

- PR2 registers
- T2CON registers
- CCPRxL and CCPRxH registers
- CCPxCON registers

It is required to have FOSC/4 as the clock input to TMR2/4/6 for correct PWM operation. Figure 21-4 shows a simplified block diagram of PWM operation.

| | |
|---|---|
| **Note:** | The corresponding TRIS bit must be cleared to enable the PWM output on the CCPx pin. |

**FIGURE 21-3: CCP PWM OUTPUT SIGNAL**



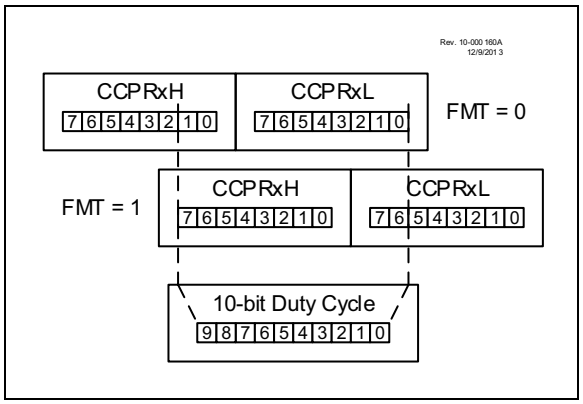
### 21.5.5 PWM DUTY CYCLE

The PWM duty cycle is specified by writing a 10-bit value to the CCPRxH:CCPRxL register pair. The alignment of the 10-bit value is determined by the FMT bit of the CCPxCON register (see Figure 21-5). The CCPRxH:CCPRxL register pair can be written to at any time; however the duty cycle value is not latched into the 10-bit buffer until after a match between PR2 and TMR2.

Equation 21-2 is used to calculate the PWM pulse width.

Equation 21-3 is used to calculate the PWM duty cycle ratio.

**FIGURE 21-5: PWM 10-BIT ALIGNMENT**




**EQUATION 21-2: PULSE WIDTH**

$$Pulse\ Width = (CCPRxH{:}CCPRxL\ register\ pair) \bullet T_{OSC} \bullet (TMR2\ Prescale\ Value)$$

**EQUATION 21-3: DUTY CYCLE RATIO**

$$Duty\ Cycle\ Ratio = \frac{(CCPRxH{:}CCPRxL\ register\ pair)}{4(PR2 + 1)}$$

CCPRxH:CCPRxL register pair are used to double buffer the PWM duty cycle. This double buffering is essential for glitchless PWM operation.

The 8-bit timer TMR2 register is concatenated with either the 2-bit internal system clock (FOSC), or two bits of the prescaler, to create the 10-bit time base. The system clock is used if the Timer2 prescaler is set to 1:1.

When the 10-bit time base matches the CCPRxH:CCPRxL register pair, then the CCPx pin is cleared (see Figure 21-4).

### 21.5.6 PWM RESOLUTION

The resolution determines the number of available duty cycles for a given period. For example, a 10-bit resolution will result in 1024 discrete duty cycles, whereas an 8-bit resolution will result in 256 discrete duty cycles.

The maximum PWM resolution is ten bits when PR2 is 255. The resolution is a function of the PR2 register value as shown by Equation 21-4.

**EQUATION 21-4: PWM RESOLUTION**

$$Resolution = \frac{log[4(PR2 + 1)]}{log(2)}\ bits$$

**Note:** If the pulse-width value is greater than the period the assigned PWM pin(s) will remain unchanged.

**FIGURE 24-14: CWG SHUTDOWN BLOCK DIAGRAM**

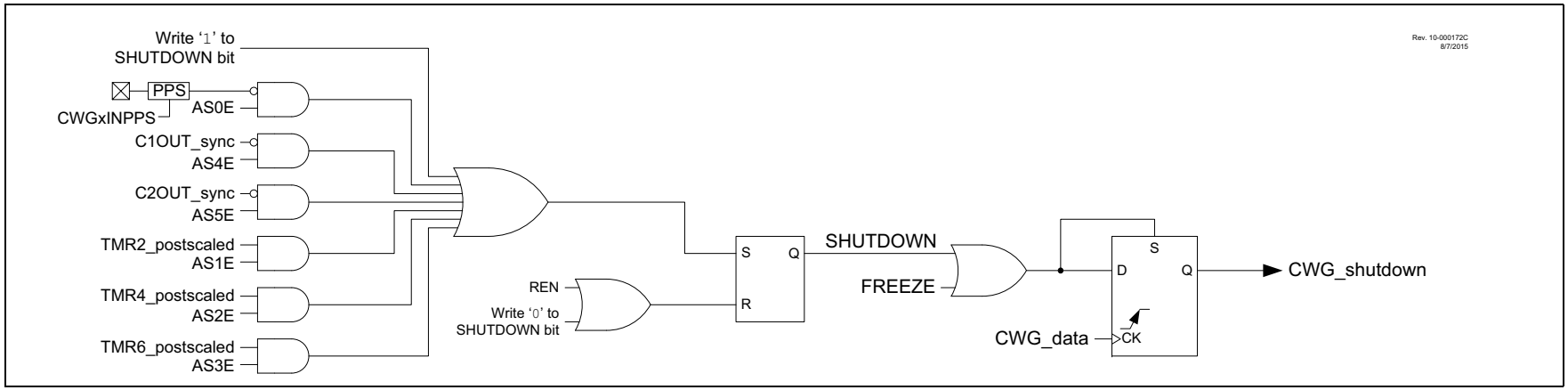



**FIGURE 24-15: SHUTDOWN FUNCTIONALITY, AUTO-RESTART DISABLED (REN = 0, LSAC = 01, LSBD = 01)**

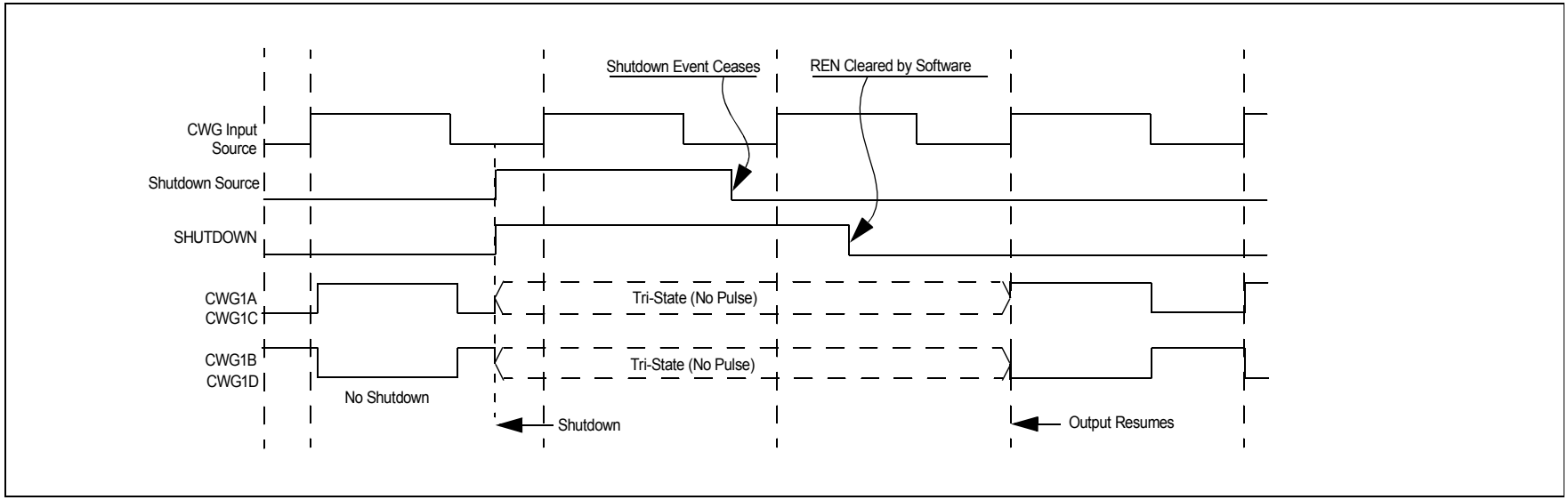


**FIGURE 25-5: Carrier Low Synchronization (MDSHSYNC = 0, MDCLSYNC = 1)**

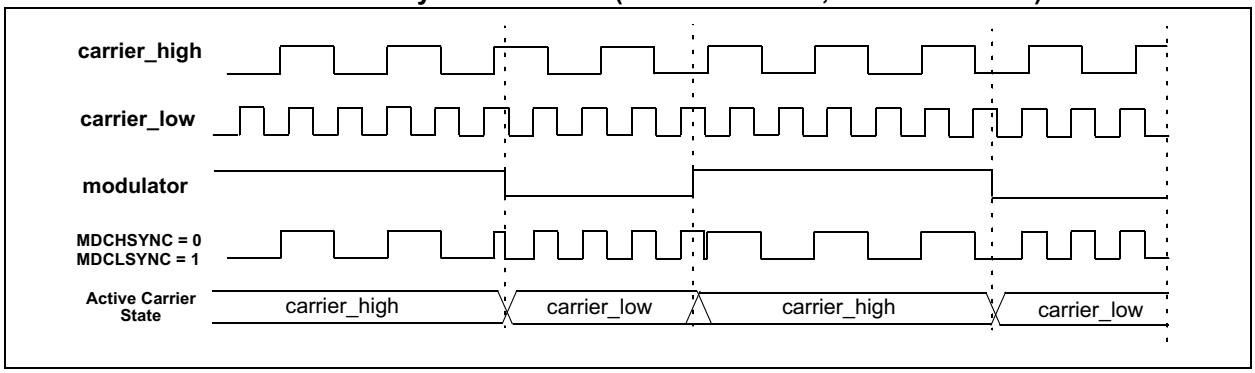
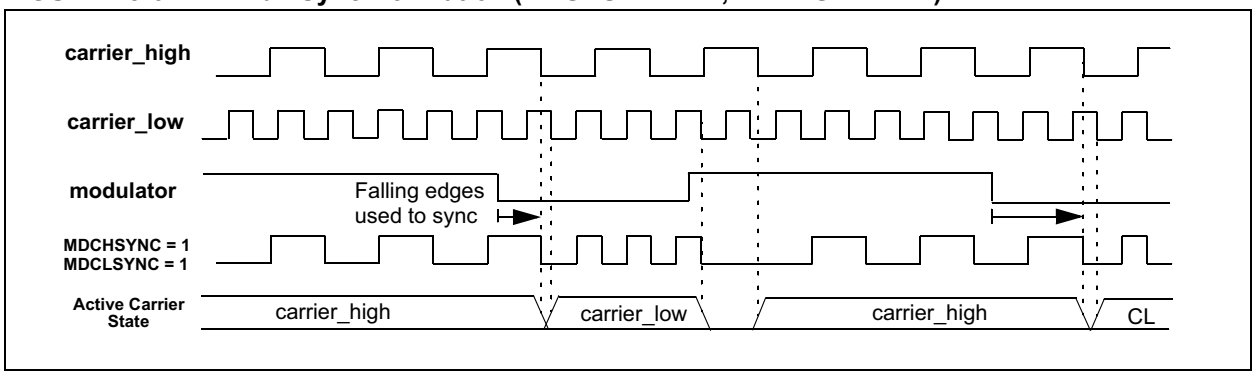



**FIGURE 25-6: Full Synchronization (MDSHSYNC = 1, MDCLSYNC = 1)**



## 26.8 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.

### 26.8.1 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.

### 26.8.2 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.

### 26.8.3 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 SSPxDATPPS registers. The SCL input is selected with the SSPxCLKPPS 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.

### 26.8.4 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 26-2: 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 hold 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. |

### 26.10.7 I²C Master Mode Reception

Master mode reception (Figure 26-29) is enabled by programming the Receive Enable bit, RCEN bit of the SSP1CON2 register.

| **Note:** | The MSSP module must be in an Idle state before the RCEN bit is set or the RCEN bit will be disregarded. |
|---|---|

The Baud Rate Generator begins counting and on each rollover, the state of the SCL pin changes (high-to-low/low-to-high) and data is shifted into the SSPSR. After the falling edge of the eighth clock, the receive enable flag is automatically cleared, the contents of the SSPSR are loaded into the SSPxBUF, the BF flag bit is set, the SSPxIF flag bit is set and the Baud Rate Generator is suspended from counting, holding SCL low. The MSSP is now in Idle state awaiting the next command. When the buffer is read by the CPU, the BF flag bit is automatically cleared. The user can then send an Acknowledge bit at the end of reception by setting the Acknowledge Sequence Enable, ACKEN bit of the SSPxCON2 register.

#### 26.10.7.1 BF Status Flag

In receive operation, the BF bit is set when an address or data byte is loaded into SSPxBUF from SSPSR. It is cleared when the SSPxBUF register is read.

#### 26.10.7.2 SSPOV Status Flag

In receive operation, the SSPOV bit is set when eight bits are received into the SSPSR and the BF flag bit is already set from a previous reception.

#### 26.10.7.3 WCOL Status Flag

If the user writes the SSPxBUF when a receive is already in progress (i.e., SSPSR is still shifting in a data byte), the WCOL bit is set and the contents of the buffer are unchanged (the write does not occur).

#### 26.10.7.4 Typical Receive Sequence:

1. The user generates a Start condition by setting the SEN bit of the SSPxCON2 register.
2. SSPxIF is set by hardware on completion of the Start.
3. SSPxIF is cleared by software.
4. User writes SSPxBUF with the slave address to transmit and the R/$\overline{W}$ bit set.
5. Address is shifted out the SDA pin until all eight bits are transmitted. Transmission begins as soon as SSPxBUF is written to.
6. The MSSP module shifts in the $\overline{ACK}$ bit from the slave device and writes its value into the ACKSTAT bit of the SSPxCON2 register.
7. The MSSP module generates an interrupt at the end of the ninth clock cycle by setting the SSPxIF bit.
8. User sets the RCEN bit of the SSPxCON2 register and the master clocks in a byte from the slave.
9. After the eighth falling edge of SCL, SSPxIF and BF are set.
10. Master clears SSPxIF and reads the received byte from SSPUF, clears BF.
11. Master sets the $\overline{ACK}$ value sent to slave in the ACKDT bit of the SSPxCON2 register and initiates the $\overline{ACK}$ by setting the ACKEN bit.
12. Master's $\overline{ACK}$ is clocked out to the slave and SSPxIF is set.
13. User clears SSPxIF.
14. Steps 8-13 are repeated for each received byte from the slave.
15. Master sends a not $\overline{ACK}$ or Stop to end communication.

## 32.9 Comparator Response Time

The comparator output is indeterminate for a period of time after the change of an input source or the selection of a new reference voltage. This period is referred to as the response time. The response time of the comparator differs from the settling time of the voltage reference. Therefore, both of these times must be considered when determining the total response time to a comparator input change. See the Comparator and Voltage Reference Specifications in Table 37-15 and Table 37-17 for more details.

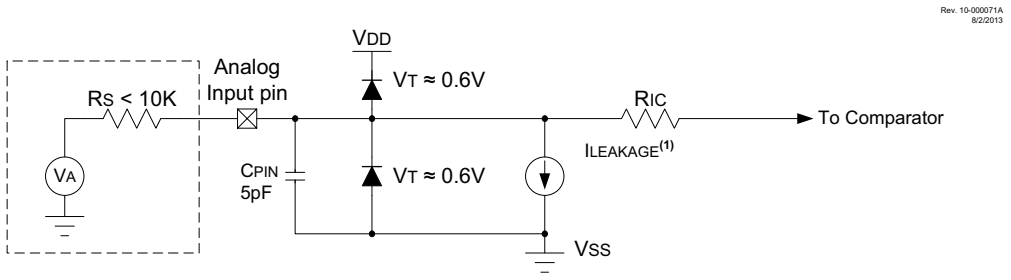
## 32.10 Analog Input Connection Considerations

A simplified circuit for an analog input is shown in Figure 32-3. Since the analog input pins share their connection with a digital input, they have reverse biased ESD protection diodes to $V_{DD}$ and $V_{SS}$. The analog input, therefore, must be between $V_{SS}$ and $V_{DD}$. If the input voltage deviates from this range by more than 0.6V in either direction, one of the diodes is forward biased and a latch-up may occur.

A maximum source impedance of 10 kΩ is recommended for the analog sources. Also, any external component connected to an analog input pin, such as a capacitor or a Zener diode, should have very little leakage current to minimize inaccuracies introduced.

| | |
|---|---|
| **Note 1:** | When reading a PORT register, all pins configured as analog inputs will read as a '0'. Pins configured as digital inputs will convert as an analog input, according to the input specification. |
| **2:** | Analog levels on any pin defined as a digital input, may cause the input buffer to consume more current than is specified. |

**FIGURE 32-3: ANALOG INPUT MODEL**




**Legend:** $C_{PIN}$ = Input Capacitance
$I_{LEAKAGE}$ = Leakage Current at the pin due to various junctions
$R_{IC}$ = Interconnect Resistance
$R_S$ = Source Impedance
$V_A$ = Analog Voltage
$V_T$ = Threshold Voltage

**Note 1:** See **Section 37.0 "Electrical Specifications"**.

**TABLE 37-3: POWER-DOWN CURRENT ($I_{PD}$)(1,2)**

| PIC18LF26/45/46K40 | | | | Standard Operating Conditions (unless otherwise stated) | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| PIC18F26/45/46K40 | | | | Standard Operating Conditions (unless otherwise stated)<br>VREGPM = 1 | | | | | |
| Param. No. | Symbol | Device Characteristics | Min. | Typ.† | Max. +85°C | Max. +125°C | Units | Conditions | |
| | | | | | | | | $V_{DD}$ | Note |
| D200 | $I_{PD}$ | $I_{PD}$ Base | — | 0.05 | 2 | 9 | µA | 3.0V | |
| D200<br>D200A | $I_{PD}$ | $I_{PD}$ Base | — | 0.4 | 4 | 12 | µA | 3.0V | |
| | | | — | 20 | — | — | µA | 3.0V | VREGPM = 0 |
| D201 | $I_{PD\_WDT}$ | Low-Frequency Internal Oscillator/ WDT | — | 0.4 | 3 | 10 | µA | 3.0V | |
| D201 | $I_{PD\_WDT}$ | Low-Frequency Internal Oscillator/ WDT | — | 0.6 | 5 | 13 | µA | 3.0V | |
| D202 | $I_{PD\_SOSC}$ | Secondary Oscillator ($S_{OSC}$) | — | 0.6 | 5 | 13 | µA | 3.0V | |
| D202 | $I_{PD\_SOSC}$ | Secondary Oscillator ($S_{OSC}$) | — | 0.8 | 8.5 | 15 | µA | 3.0V | |
| D203 | $I_{PD\_FVR}$ | FVR | — | 31 | — | — | µA | 3.0V | FVRCON = 0X81 or 0x84 |
| D203 | $I_{PD\_FVR}$ | FVR | — | 32 | — | — | µA | 3.0V | FVRCON = 0X81 or 0x84 |
| D204 | $I_{PD\_BOR}$ | Brown-out Reset (BOR) | — | 9 | 14 | 18 | µA | 3.0V | |
| D204 | $I_{PD\_BOR}$ | Brown-out Reset (BOR) | — | 14 | 19 | 21 | µA | 3.0V | |
| D205 | $I_{PD\_LPBOR}$ | Low-Power Brown-out Reset (LPBOR) | — | 0.5 | — | — | µA | 3.0V | |
| D205 | $I_{PD\_LPBOR}$ | Low-Power Brown-out Reset (LPBOR) | — | 0.7 | — | — | µA | 3.0V | |
| D206 | $I_{PD\_HLVD}$ | High/Low Voltage Detect (HLVD) | — | 31 | — | — | µA | 3.0V | |
| D206 | $I_{PD\_HLVD}$ | High/Low Voltage Detect (HLVD) | — | 32 | — | — | µA | 3.0V | |
| D207 | $I_{PD\_ADCA}$ | ADC - Active | — | 250 | — | — | µA | 3.0V | ADC is converting (4) |
| D207 | $I_{PD\_ADCA}$ | ADC - Active | — | 280 | — | — | µA | 3.0V | ADC is converting (4) |
| D208 | $I_{PD\_CMP}$ | Comparator | — | 25 | 38 | 40 | µA | 3.0V | |
| D208 | $I_{PD\_CMP}$ | Comparator | — | 28 | 50 | 60 | µA | 3.0V | |

† Data in "Typ." column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** The peripheral current is the sum of the base $I_{DD}$ and the additional current consumed when this peripheral is enabled. The peripheral Δ current can be determined by subtracting the base $I_{DD}$ or $I_{PD}$ current from this limit. Max. values should be used when calculating total current consumption.

**2:** The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode with all I/O pins in high-impedance state and tied to $V_{SS}$.

**3:** All peripheral currents listed are on a per-peripheral basis if more than one instance of a peripheral is available.

**4:** ADC clock source is FRC.

**TABLE 37-5: MEMORY PROGRAMMING SPECIFICATIONS**

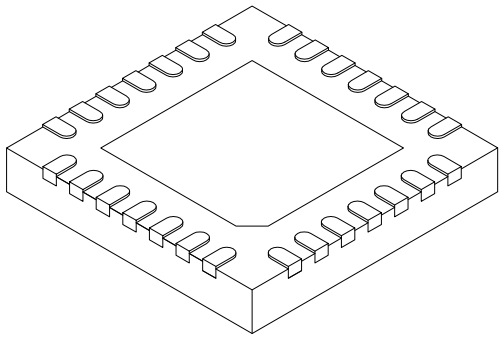
| Standard Operating Conditions (unless otherwise stated) | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym.** | **Characteristic** | **Min.** | **Typ†** | **Max.** | **Units** | **Conditions** |
| **Data EEPROM Memory Specifications** | | | | | | | |
| MEM20 | $E_D$ | DataEE Byte Endurance | 100k | — | — | E/W | -40°C ≤ TA ≤ +85°C |
| MEM21 | $T_{D\_RET}$ | Characteristic Retention | — | 40 | — | Year | Provided no other specifications are violated |
| MEM22 | $N_{D\_REF}$ | Total Erase/Write Cycles before Refresh | 1M<br>500k | 10M<br>— | —<br>— | E/W | -40°C ≤ TA ≤ +60°C<br>-40°C ≤ TA ≤ +85°C |
| MEM23 | $V_{D\_RW}$ | VDD for Read or Erase/Write operation | VDDMIN | — | VDDMAX | V | |
| MEM24 | $T_{D\_BEW}$ | Byte Erase and Write Cycle Time | — | 4.0 | 5.0 | ms | |
| **Program Flash Memory Specifications** | | | | | | | |
| MEM30 | $E_P$ | Flash Memory Cell Endurance | 10k | — | — | E/W | -40°C ≤ TA ≤ +85°C (**Note 1**) |
| MEM32 | $T_{P\_RET}$ | Characteristic Retention | — | 40 | — | Year | Provided no other specifications are violated |
| MEM33 | $V_{P\_RD}$ | VDD for Read operation | VDDMIN | — | VDDMAX | V | |
| MEM34 | $V_{P\_REW}$ | VDD for Row Erase or Write operation | VDDMIN | — | VDDMAX | V | |
| MEM35 | $T_{P\_REW}$ | Self-Timed Row Erase or Self-Timed Write | — | 2.0 | 2.5 | ms | |

† Data in “Typ” column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** Flash Memory Cell Endurance for the Flash memory is defined as: One Row Erase operation and one Self-Timed Write.

**28-Lead Plastic Quad Flat, No Lead Package (ML) - 6x6 mm Body [QFN]**
**With 0.55 mm Terminal Length**

| **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 | | 28 | |
| Pitch | e | | 0.65 BSC | |
| Overall Height | A | 0.80 | 0.90 | 1.00 |
| Standoff | A1 | 0.00 | 0.02 | 0.05 |
| Terminal Thickness | A3 | | 0.20 REF | |
| Overall Width | E | | 6.00 BSC | |
| Exposed Pad Width | E2 | 3.65 | 3.70 | 4.20 |
| Overall Length | D | | 6.00 BSC | |
| Exposed Pad Length | D2 | 3.65 | 3.70 | 4.20 |
| Terminal Width | b | 0.23 | 0.30 | 0.35 |
| Terminal Length | L | 0.50 | 0.55 | 0.70 |
| Terminal-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 C04-105C Sheet 2 of 2

## APPENDIX B: DEVICE DIFFERENCES

The differences between the devices listed in this data sheet are shown in Table B-1.

**TABLE B-1: DEVICE DIFFERENCES**

| Features[1] | PIC18(L)F26K40 | PIC18(L)F45K40 | PIC18(L)F46K40 |
|---|---|---|---|
| Program Memory (Bytes) | 65536 | 32768 | 65536 |
| SRAM (Bytes) | 3720 | 2048 | 3720 |
| EEPROM (Bytes) | 1024 | 256 | 1024 |
| Interrupt Sources | 36 | 36 | 36 |
| I/O Ports | Ports A, B, C, (E) | Ports A, B, C, D, E | Ports A, B, C, D, E |
| Capture/Compare/PWM Modules (CCP) | 2 | 2 | 2 |
| 10-bit Analog-to-Digital Module | 4 internal<br>24 external | 4 internal<br>35 external | 4 internal<br>35 external |
| Packages | 28-pin SPDIP<br>28-pin SOIC<br>28-pin SSOP<br>28-pin QFN<br>28-pin UQFN | 40-pin PDIP<br>40-pin UQFN<br>44-pin TQFP<br>44-pin QFN | 40-pin PDIP<br>40-pin UQFN<br>44-pin TQFP<br>44-pin QFN |

**Note 1:** PIC18F2x/4xK40: operating voltage, 2.3V-5.5V. PIC18LF2x/4xK40: operating voltage, 1.8V-3.6V.