**Welcome to E-XFL.COM**

## What is "Embedded - Microcontrollers"?

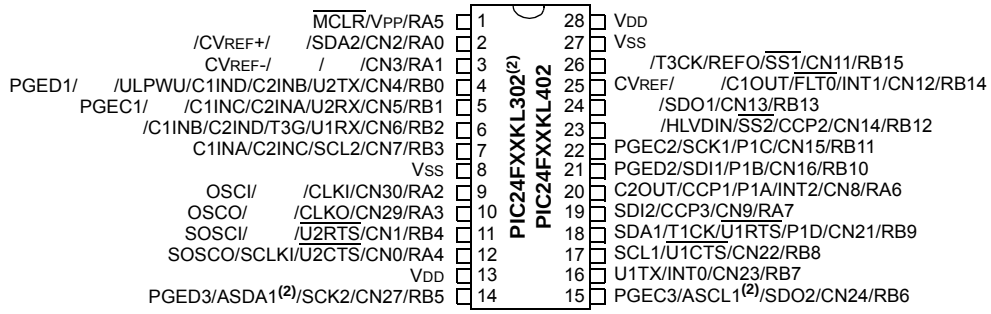
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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"

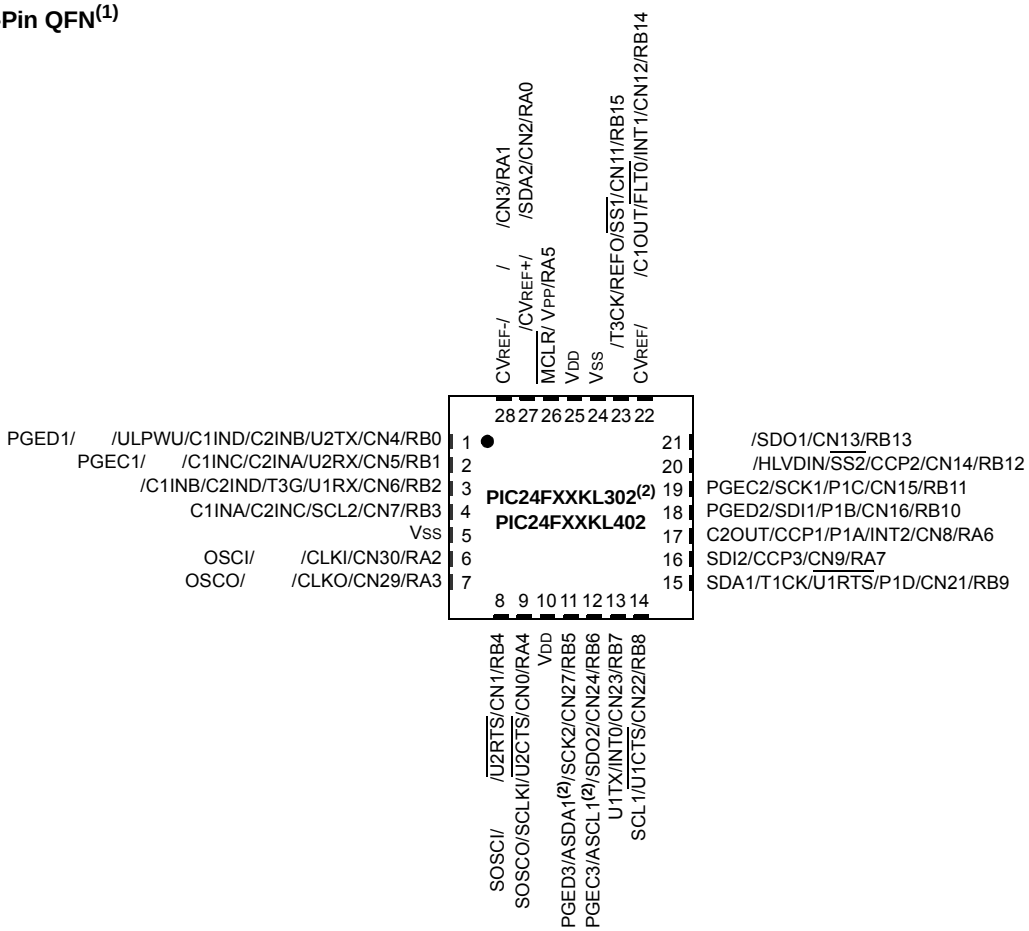
| Details | |
|---|---|
| Product Status | Active |
| Core Processor | PIC |
| Core Size | 16-Bit |
| Speed | 32MHz |
| Connectivity | I²C, IrDA, LINbus, SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, HLVD, POR, PWM, WDT |
| Number of I/O | 18 |
| Program Memory Size | 16KB (5.5K x 24) |
| Program Memory Type | FLASH |
| EEPROM Size | 512 x 8 |
| RAM Size | 1K x 8 |
| Voltage - Supply (Vcc/Vdd) | 1.8V ~ 3.6V |
| Data Converters | A/D 12x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 125°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 20-SOIC (0.295", 7.50mm Width) |
| Supplier Device Package | 20-SOIC |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic24f16kl401-e-so |

## Pin Diagrams: PIC24FXXKL302/402

### 28-Pin SPDIP/SSOP/SOIC(1)




### 28-Pin QFN(1)




**Contact your Microchip sales team for Chip Scale Package (CSP) availability.**

**Note 1:** Analog features (indicated in ) are not available on PIC24FXXKL302 devices.

**2:** Alternate location for $I^2C$™ functionality of MSSP1, as determined by the I2C1SEL Configuration bit.

## Table of Contents

### 3.3.2 DIVIDER

The divide block supports 32-bit/16-bit and 16-bit/16-bit signed and unsigned integer divide operations with the following data sizes:

1. 32-bit signed/16-bit signed divide
2. 32-bit unsigned/16-bit unsigned divide
3. 16-bit signed/16-bit signed divide
4. 16-bit unsigned/16-bit unsigned divide

The quotient for all divide instructions ends up in W0 and the remainder in W1. Sixteen-bit signed and unsigned DIV instructions can specify any W register for both the 16-bit divisor (Wn), and any W register (aligned) pair (W(m + 1):Wm) for the 32-bit dividend. The divide algorithm takes one cycle per bit of divisor, so both 32-bit/16-bit and 16-bit/16-bit instructions take the same number of cycles to execute.

### 3.3.3 MULTI-BIT SHIFT SUPPORT

The PIC24F ALU supports both single bit and single-cycle, multi-bit arithmetic and logic shifts. Multi-bit shifts are implemented using a shifter block, capable of performing up to a 15-bit arithmetic right shift, or up to a 15-bit left shift, in a single cycle. All multi-bit shift instructions only support Register Direct Addressing for both the operand source and result destination.

A full summary of instructions that use the shift operation is provided in Table 3-2.

**TABLE 3-2: INSTRUCTIONS THAT USE THE SINGLE AND MULTI-BIT SHIFT OPERATION**

| Instruction | Description |
|---|---|
| ASR | Arithmetic shift right source register by one or more bits. |
| SL | Shift left source register by one or more bits. |
| LSR | Logical shift right source register by one or more bits. |

### 4.3.3 READING DATA FROM PROGRAM MEMORY USING PROGRAM SPACE VISIBILITY

The upper 32 Kbytes of data space may optionally be mapped into a 16K word page of the program space. This provides transparent access of stored constant data from the data space without the need to use special instructions (i.e., TBLRDL/H).

Program space access through the data space occurs if the MSb of the data space EA is '1' and PSV is enabled by setting the PSV bit in the CPU Control (CORCON<2>) register. The location of the program memory space to be mapped into the data space is determined by the Program Space Visibility Page Address (PSVPAG) register. This 8-bit register defines any one of 256 possible pages of 16K words in program space. In effect, PSVPAG functions as the upper 8 bits of the program memory address, with 15 bits of the EA functioning as the lower bits.

By incrementing the PC by 2 for each program memory word, the lower 15 bits of data space addresses directly map to the lower 15 bits in the corresponding program space addresses.

Data reads from this area add an additional cycle to the instruction being executed, since two program memory fetches are required.

Although each data space address, 8000h and higher, maps directly into a corresponding program memory address (see Figure 4-7), only the lower 16 bits of the 24-bit program word are used to contain the data. The upper 8 bits of any program space location, used as data, should be programmed with '1111 1111' or '0000 0000' to force a NOP. This prevents possible issues should the area of code ever be accidentally executed.

| **Note:** | PSV access is temporarily disabled during Table Reads/Writes. |
|---|---|

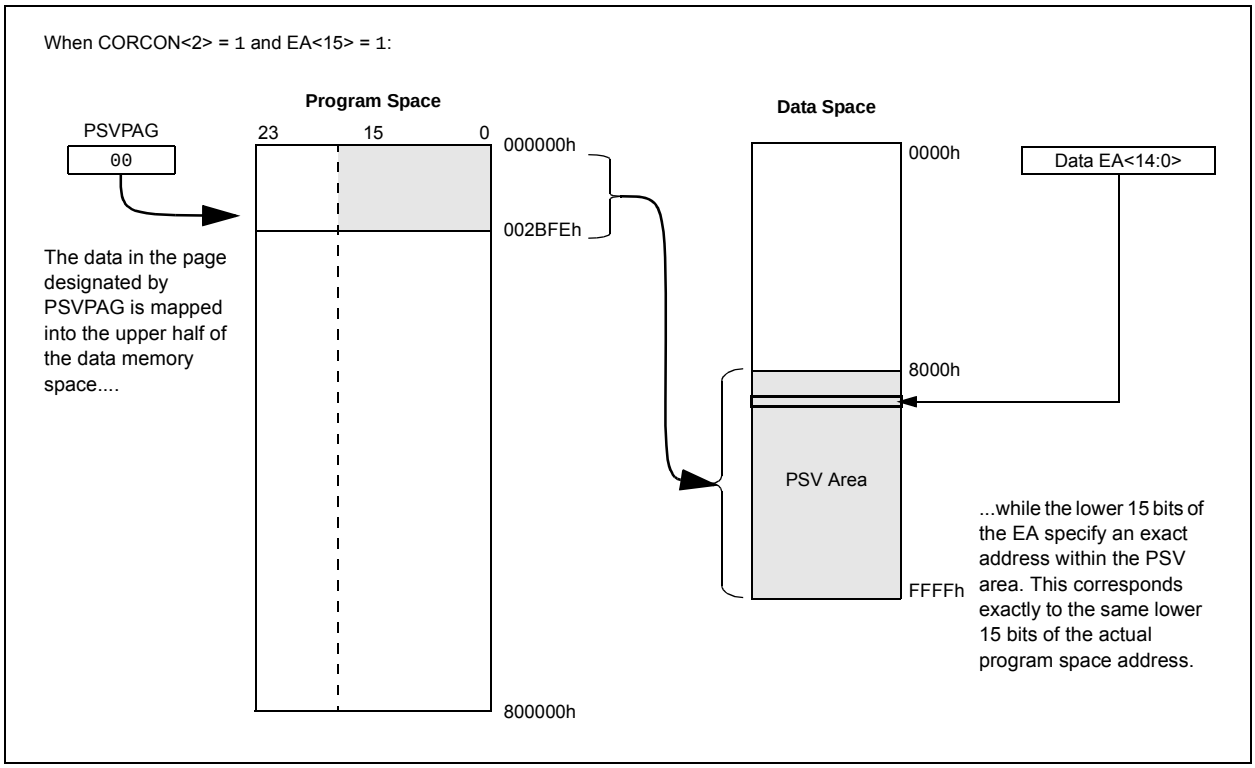
For operations that use PSV and are executed outside of a REPEAT loop, the MOV and MOV.D instructions will require one instruction cycle, in addition to the specified execution time. All other instructions will require two instruction cycles in addition to the specified execution time.

For operations that use PSV, which are executed inside a REPEAT loop, there will be some instances that require two instruction cycles, in addition to the specified execution time of the instruction:

- Execution in the first iteration
- Execution in the last iteration
- Execution prior to exiting the loop due to an interrupt
- Execution upon re-entering the loop after an interrupt is serviced

Any other iteration of the REPEAT loop will allow the instruction accessing data, using PSV, to execute in a single cycle.

**FIGURE 4-7: PROGRAM SPACE VISIBILITY OPERATION**



## 8.3 Interrupt Control and Status Registers

Depending on the particular device, the PIC24F16KL402 family of devices implements up to 28 registers for the interrupt controller:

- INTCON1
- INTCON2
- IFS0 through IFS5
- IEC0 through IEC5
- IPC0 through IPC7, ICP9, IPC12, ICP16, ICP18 and IPC20
- INTTREG

Global interrupt control functions are controlled from INTCON1 and INTCON2. INTCON1 contains the Interrupt Nesting Disable (NSTDIS) bit, as well as the control and status flags for the processor trap sources. The INTCON2 register controls the external interrupt request signal behavior and the use of the AIV table.

The IFSx registers maintain all of the interrupt request flags. Each source of interrupt has a status bit, which is set by the respective peripherals or external signal, and is cleared via software.

The IECx registers maintain all of the interrupt enable bits. These control bits are used to individually enable interrupts from the peripherals or external signals.

The IPCx registers are used to set the Interrupt Priority Level for each source of interrupt. Each user interrupt source can be assigned to one of eight priority levels.

The INTTREG register contains the associated interrupt vector number and the new CPU Interrupt Priority Level, which are latched into the Vector Number (VECNUM<6:0>) and the Interrupt Level (ILR<3:0>) bit fields in the INTTREG register. The new Interrupt Priority Level is the priority of the pending interrupt.

The interrupt sources are assigned to the IFSx, IECx and IPCx registers in the same sequence listed in Table 8-2. For example, the INT0 (External Interrupt 0) is depicted as having a vector number and a natural order priority of 0. The INT0IF status bit is found in IFS0<0>, the INT0IE enable bit in IEC0<0> and the INT0IP<2:0> priority bits are in the first position of IPC0 (IPC0<2:0>).

Although they are not specifically part of the interrupt control hardware, two of the CPU control registers contain bits that control interrupt functionality. The ALU STATUS Register (SR) contains the IPL<2:0> bits (SR<7:5>). These indicate the current CPU Interrupt Priority Level. The user may change the current CPU priority level by writing to the IPL bits.

The CORCON register contains the IPL3 bit, which together with the IPL<2:0> bits, also indicates the current CPU priority level. IPL3 is a read-only bit so that the trap events cannot be masked by the user's software.

All interrupt registers are described in Register 8-3 through Register 8-30, in the following sections.

**REGISTER 8-12: IEC1: INTERRUPT ENABLE CONTROL REGISTER 1**

| R/W-0 | R/W-0 | R/W-0 | U-0 | R/W-0 | U-0 | R/W-0 | U-0 |
|---|---|---|---|---|---|---|---|
| U2TXIE(1) | U2RXIE(1) | INT2IE | — | T4IE(1) | — | CCP3IE(1) | — |
| bit 15 | | | | | | | bit 8 |

| U-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | — | INT1IE | CNIE | CMIE | BCL1IE | SSP1IE |
| 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 **U2TXIE:** UART2 Transmitter Interrupt Enable bit(1)
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 14 **U2RXIE:** UART2 Receiver Interrupt Enable bit(1)
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 13 **INT2IE:** External Interrupt 2 Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 12 **Unimplemented:** Read as '0'

bit 11 **T4IE:** Timer4 Interrupt Enable bit(1)
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 10 **Unimplemented:** Read as '0'

bit 9 **CCP3IE:** Capture/Compare/PWM3 Interrupt Enable bit(1)
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 8-5 **Unimplemented:** Read as '0'

bit 4 **INT1IE:** External Interrupt 1 Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 3 **CNIE:** Input Change Notification Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 2 **CMIE:** Comparator Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 1 **BCL1IE:** MSSP1 $I^2C$™ Bus Collision Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 0 **SSP1IE:** MSSP1 SPI/$I^2C$ Event Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

**Note 1:** These bits are unimplemented on PIC24FXXKL10X and PIC24FXXKL20X devices.

The following code sequence for a clock switch is recommended:

1. Disable interrupts during the OSCCON register unlock and write sequence.
2. Execute the unlock sequence for the OSCCON high byte by writing 78h and 9Ah to OSCCON<15:8>, in two back-to-back instructions.
3. Write the new oscillator source to the NOSCx bits in the instruction immediately following the unlock sequence.
4. Execute the unlock sequence for the OSCCON low byte by writing 46h and 57h to OSCCON<7:0>, in two back-to-back instructions.
5. Set the OSWEN bit in the instruction immediately following the unlock sequence.
6. Continue to execute code that is not clock-sensitive (optional).
7. Invoke an appropriate amount of software delay (cycle counting) to allow the selected oscillator and/or PLL to start and stabilize.
8. Check to see if OSWEN is '0'. If it is, the switch was successful. If OSWEN is still set, then check the LOCK bit to determine the cause of failure.

The core sequence for unlocking the OSCCON register and initiating a clock switch is shown in Example 9-1.

**EXAMPLE 9-1: BASIC CODE SEQUENCE FOR CLOCK SWITCHING**

```
;Place the new oscillator selection in W0
;OSCCONH (high byte) Unlock Sequence
MOV        #OSCCONH, w1
MOV        #0x78, w2
MOV        #0x9A, w3
MOV.b      w2, [w1]
MOV.b      w3, [w1]
;Set new oscillator selection
MOV.b      WREG, OSCCONH
;OSCCONL (low byte) unlock sequence
MOV        #OSCCONL, w1
MOV        #0x46, w2
MOV        #0x57, w3
MOV.b      w2, [w1]
MOV.b      w3, [w1]
;Start oscillator switch operation
BSET       OSCCON,#0
```

## 9.5 Reference Clock Output

In addition to the CLKO output (FOSC/2) available in certain oscillator modes, the device clock in the PIC24F16KL402 family devices can also be configured to provide a reference clock output signal to a port pin. This feature is available in all oscillator configurations and allows the user to select a greater range of clock submultiples to drive external devices in the application.

This reference clock output is controlled by the REFOCON register (Register 9-4). Setting the ROEN bit (REFOCON<15>) makes the clock signal available on the REFO pin. The RODIV bits (REFOCON<11:8>) enable the selection of 16 different clock divider options.

The ROSSLP and ROSEL bits (REFOCON<13:12>) control the availability of the reference output during Sleep mode. The ROSEL bit determines if the oscillator on OSC1 and OSC2, or the current system clock source, is used for the reference clock output. The ROSSLP bit determines if the reference source is available on REFO when the device is in Sleep mode.

To use the reference clock output in Sleep mode, both the ROSSLP and ROSEL bits must be set. The device clock must also be configured for one of the primary modes (EC, HS or XT). Therefore, if the ROSEL bit is also not set, the oscillator on OSC1 and OSC2 will be powered down when the device enters Sleep mode. Clearing the ROSEL bit allows the reference output frequency to change as the system clock changes during any clock switches.

## 10.3 Ultra Low-Power Wake-up

The Ultra Low-Power Wake-up (ULPWU) on pin, RB0, allows a slow falling voltage to generate an interrupt without excess current consumption. This feature provides a low-power technique for periodically waking up the device from Sleep mode.

To use this feature:

1. Charge the capacitor on RB0 by configuring the RB0 pin to an output and setting it to '1'.
2. Stop charging the capacitor by configuring RB0 as an input.
3. Discharge the capacitor by setting the ULPEN and ULPSINK bits in the ULPWCON register.
4. Configure Sleep mode.
5. Enter Sleep mode.

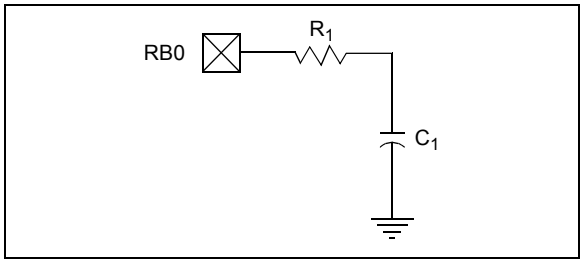
The time-out is dependent on the discharge time of the RC circuit on RB0. When the voltage on RB0 drops below $V_{IL}$, the device wakes up and executes the next instruction.

When the ULPWU module wakes the device from Sleep mode, the ULPWUIF bit (IFS5<0>) is set. Software can check this bit upon wake-up to determine the wake-up source.

See Example 10-2 for initializing the ULPWU module.

A series resistor, between RB0 and the external capacitor, provides overcurrent protection for the RB0/AN2/ULPWU pin and enables software calibration of the time-out (see Figure 10-1).

**FIGURE 10-1: SERIES RESISTOR**




A timer can be used to measure the charge time and discharge time of the capacitor. The charge time can then be adjusted to provide the desired delay in Sleep. This technique compensates for the affects of temperature, voltage and component accuracy. The peripheral can also be configured as a simple, programmable Low-Voltage Detect (LVD) or temperature sensor.

**EXAMPLE 10-2: ULTRA LOW-POWER WAKE-UP INITIALIZATION**

```
//*****************************************************************************
// 1. Charge the capacitor on RB0
//*****************************************************************************
    TRISBbits.TRISB0 = 0;
    LATBbits.LATB0 = 1;
    for(i = 0; i < 10000; i++) Nop();
//*****************************************************************************
//2. Stop Charging the capacitor on RB0
//*****************************************************************************
    TRISBbits.TRISB0 = 1;
//*****************************************************************************
//3. Enable ULPWU Interrupt
//*****************************************************************************
IFS5bits.ULPWUIF = 0;
IEC5bits.ULPWUIE = 1;
IPC20bits.ULPWUIP = 0x7;
//*****************************************************************************
//4. Enable the Ultra Low Power Wakeup module and allow capacitor discharge
//*****************************************************************************
    ULPWCONbits.ULPEN = 1;
    ULPWCONbits.ULPSINK = 1;
//*****************************************************************************
//5. Enter Sleep Mode
//*****************************************************************************
    Sleep();
//for Sleep, execution will resume here
```

## 14.0 TIMER3 MODULE

| | |
|---|---|
| **Note:** | This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information on Timers, refer to the *"dsPIC33/PIC24 Family Reference Manual"*, **"Timers"** (DS39704). |

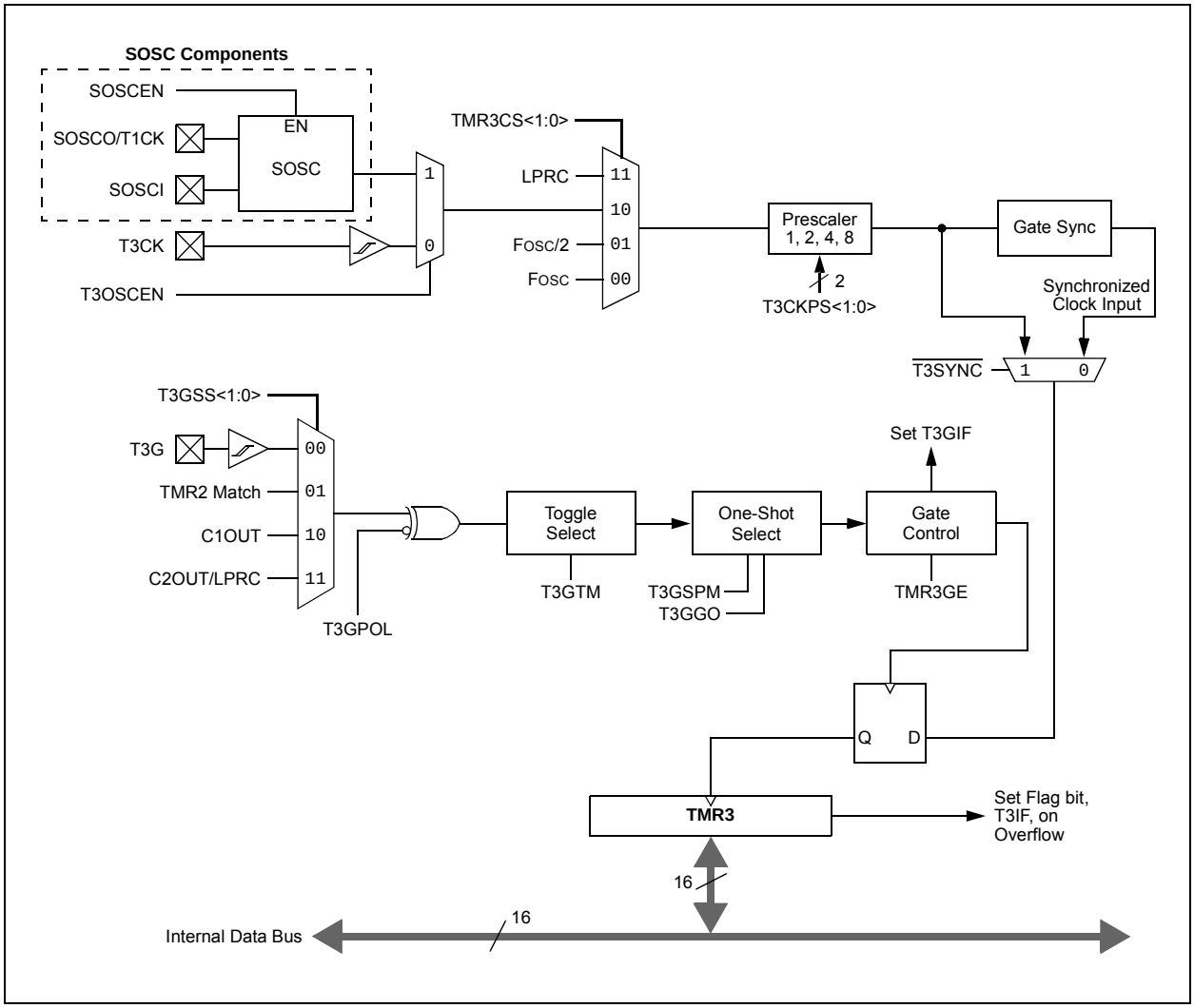
The Timer3 timer/counter modules incorporate these features:

- Software-selectable operation as a 16-bit timer or counter
- One 16-bit readable and writable Timer Value register
- Selectable clock source (internal or external) with device clock, SOSC or LPRC oscillator options
- Interrupt-on-overflow
- Multiple timer gating options, including:
  - User-selectable gate sources and polarity
  - Gate/toggle operation
  - Single Pulse (One-Shot) mode
- Module Reset on ECCP Special Event Trigger

The Timer3 module is controlled through the T3CON register (Register 14-1). A simplified block diagram of the Timer3 module is shown in Figure 14-1.

The FOSC clock source should not be used with the ECCP capture/compare features. If the timer will be used with the capture or compare features, always select one of the other timer clocking options.

**FIGURE 14-1: TIMER3 BLOCK DIAGRAM**



## 16.0 CAPTURE/COMPARE/PWM (CCP) AND ENHANCED CCP MODULES

**Note:** This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information on the Capture/Compare/PWM module, refer to the *"dsPIC33/PIC24 Family Reference Manual"*.

Depending on the particular device, PIC24F16KL402 family devices include up to three CCP and/or ECCP modules. Key features of all CCP modules include:

- 16-bit input capture for a range of edge events
- 16-bit output compare with multiple output options
- Single-output Pulse-Width Modulation (PWM) with up to 10 bits of resolution
- User-selectable time base from any available timer
- Special Event Trigger on capture and compare events to automatically trigger a range of peripherals

ECCP modules also include these features:

- Operation in Half-Bridge and Full-Bridge (Forward and Reverse) modes
- Pulse steering control across any or all Enhanced PWM pins with user-configurable steering synchronization
- User-configurable external Fault detect with auto-shutdown and auto-restart

PIC24FXXKL40X/30X devices instantiate three CCP modules, one Enhanced (ECCP1) and two standard (CCP2 and CCP3). All other devices instantiate two standard CCP modules (CCP1 and CCP2).

### 16.1 Timer Selection

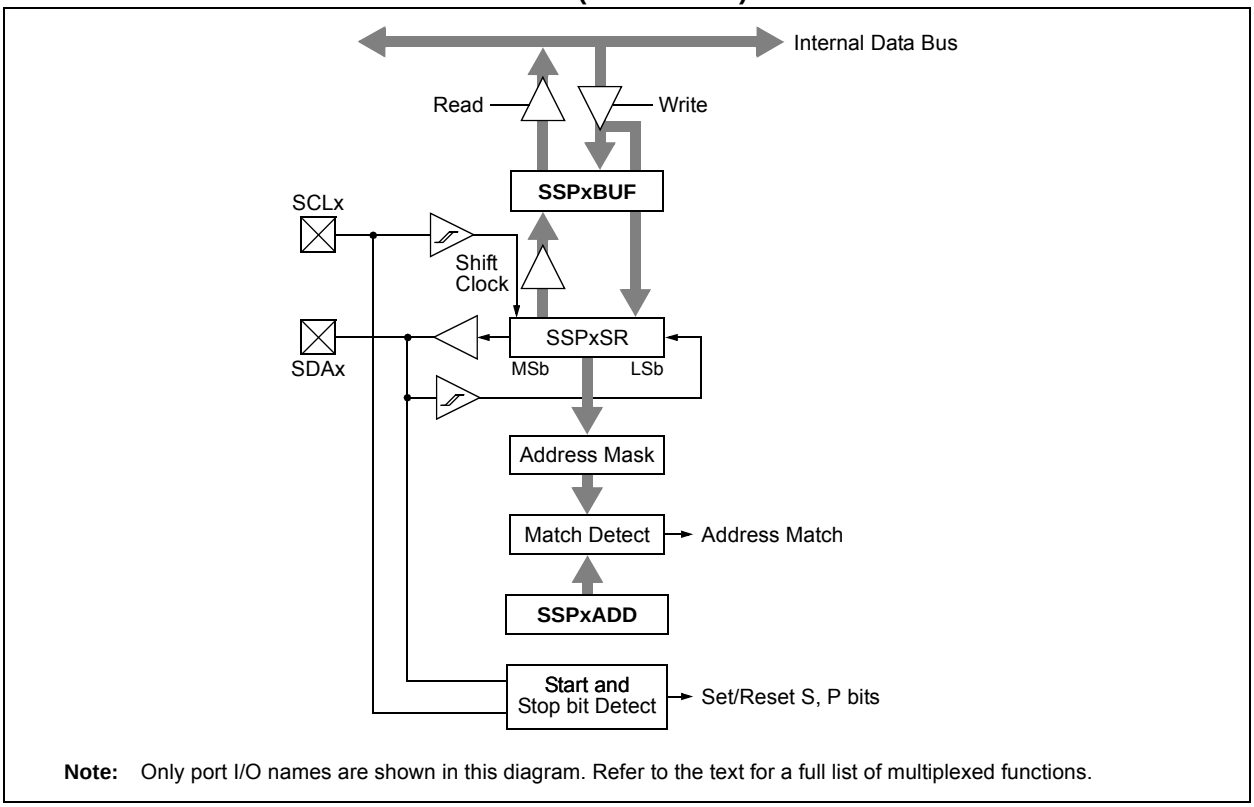
On all PIC24F16KL402 family devices, the CCP and ECCP modules use Timer3 as the time base for capture and compare operations. PWM and Enhanced PWM operations may use either Timer2 or Timer4. PWM time base selection is done through the CCPTMRS0 register (Register 16-6).

### 16.2 CCP I/O Pins

To configure I/O pins with a CCP function, the proper mode must be selected by setting the CCPxM<3:0> bits.

Where the Enhanced CCP module is available, it may have up to four PWM outputs depending on the selected operating mode. These outputs are designated, P1A through P1D. The outputs that are active depend on the ECCP operating mode selected. To configure I/O pins for Enhanced PWM operation, the proper PWM mode must be selected by setting the PM<1:0> and CCPxM<3:0> bits.

**FIGURE 17-3: MSSPx BLOCK DIAGRAM (I²C™ MODE)**




**Note:** Only port I/O names are shown in this diagram. Refer to the text for a full list of multiplexed functions.

**FIGURE 17-4: MSSPx BLOCK DIAGRAM (I²C™ MASTER MODE)**

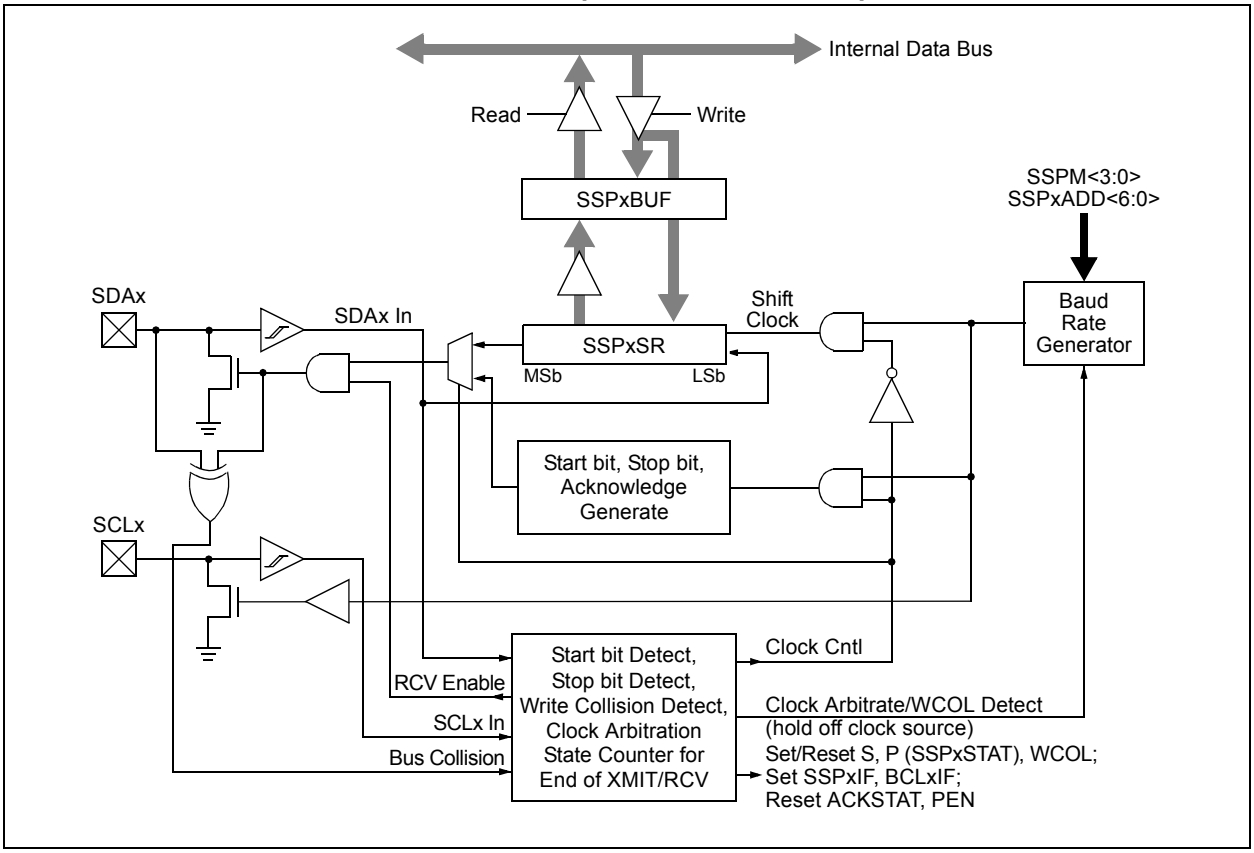
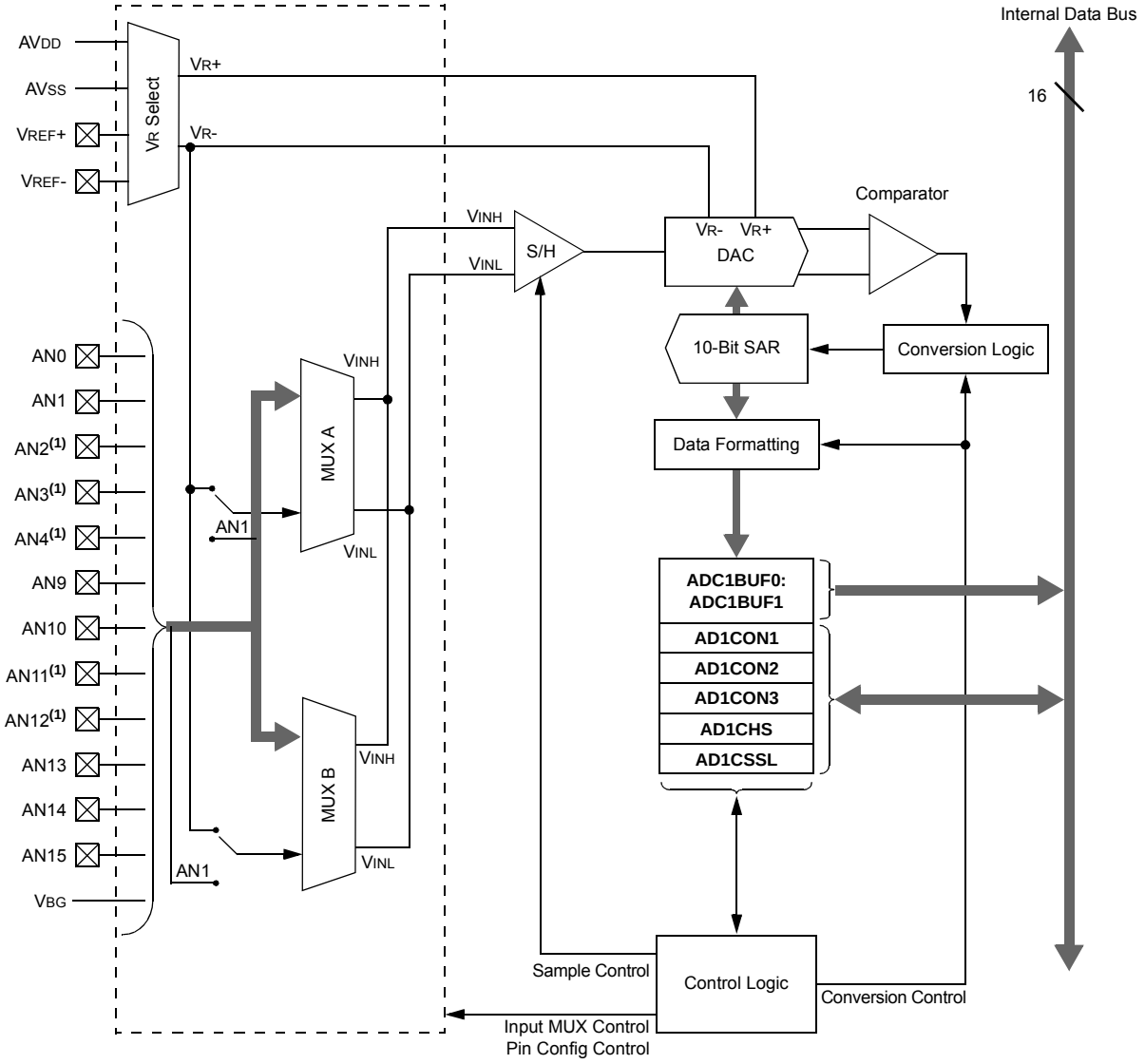


**FIGURE 19-1: 10-BIT HIGH-SPEED A/D CONVERTER BLOCK DIAGRAM**




**Note 1:** Unimplemented in 14-pin devices.

**REGISTER 23-1: FBS: BOOT SEGMENT CONFIGURATION REGISTER**

| U-0 | U-0 | U-0 | U-0 | R/C-1(1) | R/C-1(1) | R/C-1(1) | R/C-1(1) |
|---|---|---|---|---|---|---|---|
| — | — | — | — | BSS2 | BSS1 | BSS0 | BWRP |
| bit 7 | | | | | | | bit 0 |

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

bit 7-4 **Unimplemented:** Read as '0'

bit 3-1 **BSS<2:0>:** Boot Segment Program Flash Code Protection bits(1)
- 111 = No Boot Segment; all program memory space is General Segment
- 110 = Standard security Boot Segment starts at 0200h, ends at 0AFEh
- 101 = Standard security Boot Segment starts at 0200h, ends at 15FEh(2)
- 100 = Reserved
- 011 = Reserved
- 010 = High-security Boot Segment starts at 0200h, ends at 0AFEh
- 001 = High-security Boot Segment starts at 0200h, ends at 15FEh(2)
- 000 = Reserved

bit 0 **BWRP:** Boot Segment Program Flash Write Protection bit(1)
- 1 = Boot Segment may be written
- 0 = Boot Segment is write-protected

**Note 1:** Code protection bits can only be programmed by clearing them. They can be reset to their default factory state ('1'), but only by performing a bulk erase and reprogramming the entire device.

**2:** This selection is available only on PIC24F16KL40X devices.

**REGISTER 23-2: FGS: GENERAL SEGMENT CONFIGURATION REGISTER**

| U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | R/C-1(1) | R/C-1(1) |
|---|---|---|---|---|---|---|---|
| — | — | — | — | — | — | GSS0 | GWRP |
| bit 7 | | | | | | | bit 0 |

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

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

bit 1 **GSS0:** General Segment Code Flash Code Protection bit(1)
- 1 = No protection
- 0 = Standard security is enabled

bit 0 **GWRP:** General Segment Code Flash Write Protection bit(1)
- 1 = General Segment may be written
- 0 = General Segment is write-protected

**Note 1:** Code protection bits can only be programmed by clearing them. They can be reset to their default factory state ('1'), but only by performing a bulk erase and reprogramming the entire device.

## 23.3 Unique ID

A read-only Unique ID value is stored at addresses, 800802h through 800808h. This factory programmed value is unique to each microcontroller produced in the PIC24F16KL402 family. To access this region, use Table Read instructions or Program Space Visibility.

To ensure a globally Unique ID across other Microchip microcontroller families, the “Unique ID” value should be further concatenated with the family and Device ID values stored at address, FF0000h.

**REGISTER 23-8: DEVID: DEVICE ID REGISTER**

| U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| — | — | — | — | — | — | — | — |
| bit 23 | | | | | | | bit 16 |

| R | R | R | R | R | R | R | R |
|---|---|---|---|---|---|---|---|
| FAMID7 | FAMID6 | FAMID5 | FAMID4 | FAMID3 | FAMID2 | FAMID1 | FAMID0 |
| bit 15 | | | | | | | bit 8 |

| R | R | R | R | R | R | R | R |
|---|---|---|---|---|---|---|---|
| DEV7 | DEV6 | DEV5 | DEV4 | DEV3 | DEV2 | DEV1 | DEV0 |
| 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 23-16 **Unimplemented:** Read as ‘0’

bit 15-8 **FAMID<7:0>:** Device Family Identifier bits
01001011 = PIC24F16KL402 family

bit 7-0 **DEV<7:0>:** Individual Device Identifier bits
00000001 = PIC24F04KL100
00000010 = PIC24F04KL101

00000101 = PIC24F08KL200
00000110 = PIC24F08KL201

00001010 = PIC24F08KL301
00000000 = PIC24F08KL302

00001110 = PIC24F08KL401
00000100 = PIC24F08KL402
00011110 = PIC24F16KL401
00010100 = PIC24F16KL402

**NOTES:**

**TABLE 26-6: DC CHARACTERISTICS: OPERATING CURRENT ($I_{DD}$)(2)**

| DC CHARACTERISTICS | | | Standard Operating Conditions: 1.8V to 3.6V<br>Operating temperature -40°C ≤ TA ≤ +85°C for Industrial<br>-40°C ≤ TA ≤ +125°C for Extended | | | |
|---|---|---|---|---|---|---|
| **Parameter No.** | **Typical(1)** | **Max** | **Units** | **Conditions** | | |
| **IDD Current** | | | | | | |
| DC20 | 0.154 | 0.350 | mA | 1.8V | +85V°C | 0.5 MIPS, FOSC = 1 MHz |
| | 0.301 | 0.630 | | 3.3V | | |
| | — | .500 | mA | 1.8V | +125°C | |
| | — | .800 | | 3.3V | | |
| DC22 | 0.300 | — | mA | 1.8V | +85°C | 1 MIPS, FOSC = 2 MHz |
| | 0.585 | — | | 3.3V | | |
| DC24 | 7.76 | 12.0 | mA | 3.3V | +85°C | 16 MIPS, FOSC = 32 MHz |
| | — | 18.0 | | 3.3V | +125°C | |
| DC26 | 1.44 | — | mA | 1.8V | +85°C | FRC (4 MIPS), FOSC = 8 MHz |
| | 2.71 | — | | 3.3V | | |
| DC30 | 4.00 | 28.0 | µA | 1.8V | +85°C | LPRC (15.5 KIPS), FOSC = 31 kHz |
| | 9.00 | 55.0 | | 3.3V | | |
| | — | 45.0 | µA | 1.8V | +125°C | |
| | — | 90.0 | | 3.3V | | |

**Note 1:** Data in the Typical column is at 3.3V, +25°C, unless otherwise stated.

**2:** IDD is measured with all peripherals disabled. All I/Os are configured as outputs and set low; PMDx bits are set to '1' and WDT, etc., are all disabled.

**TABLE 26-7: DC CHARACTERISTICS: IDLE CURRENT ($I_{IDLE}$)(2)**

| DC CHARACTERISTICS | | | Standard Operating Conditions: 1.8V to 3.6V<br>Operating temperature -40°C ≤ TA ≤ +85°C for Industrial<br>-40°C ≤ TA ≤ +125°C for Extended | | | |
|---|---|---|---|---|---|---|
| **Parameter No.** | **Typical(1)** | **Max** | **Units** | **Conditions** | | |
| **Idle Current (IIDLE)** | | | | | | |
| DC40 | 0.035 | 0.080 | mA | 1.8V | +85°C | 0.5 MIPS, FOSC = 1 MHz |
| | 0.077 | 0.150 | | 3.3V | | |
| | — | 0.160 | mA | 1.8V | +125°C | |
| | — | 0.300 | | 3.3V | | |
| DC42 | 0.076 | — | mA | 1.8V | +85°C | 1 MIPS, FOSC = 2 MHz |
| | 0.146 | — | | 3.3V | | |
| DC44 | 2.52 | 3.20 | mA | 3.3V | +85°C | 16 MIPS, FOSC = 32 MHz |
| | — | 5.00 | mA | 3.3V | +125°C | |
| DC46 | 0.45 | — | mA | 1.8V | +85°C | FRC (4 MIPS), FOSC = 8 MHz |
| | 0.76 | — | mA | 3.3V | | |
| DC50 | 0.87 | 18.0 | µA | 1.8V | +85°C | LPRC (15.5 KIPS), FOSC = 31 kHz |
| | 1.55 | 40.0 | µA | 3.3V | | |
| | — | 27.0 | µA | 1.8V | +125°C | |
| | — | 50.0 | µA | 3.3V | | |

**Note 1:** Data in the Typical column is at 3.3V, +25°C, unless otherwise stated.

**2:** IIDLE is measured with all I/Os configured as outputs and set low; PMDx bits are set to '1' and WDT, etc., are all disabled.

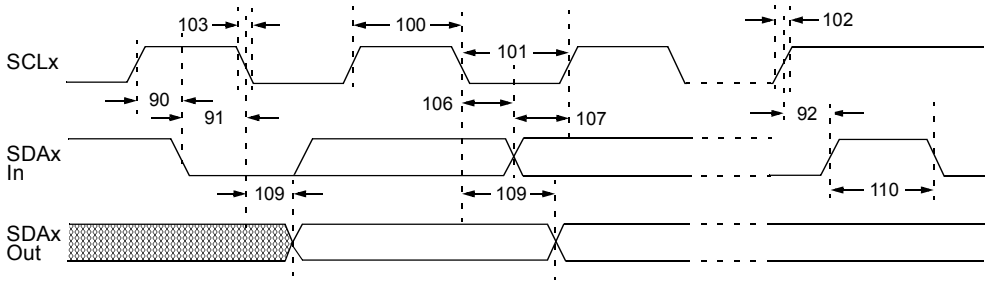
**TABLE 26-8: DC CHARACTERISTICS: POWER-DOWN CURRENT ($I_{PD}$)**

| DC CHARACTERISTICS | | | Standard Operating Conditions: 1.8V to 3.6V<br>Operating temperature -40°C ≤ TA ≤ +85°C for Industrial<br>-40°C ≤ TA ≤ +125°C for Extended | | | |
|---|---|---|---|---|---|---|
| **Parameter No.** | **Typical(1)** | **Max** | **Units** | **Conditions** | | |
| **Power-Down Current ($I_{PD}$)** | | | | | | |
| DC60 | 0.01 | 0.20 | µA | -40°C | 1.8V | Sleep Mode(2) |
| | 0.03 | 0.20 | µA | +25°C | | |
| | 0.06 | 0.87 | µA | +60°C | | |
| | 0.20 | 1.35 | µA | +85°C | | |
| | — | 8.00 | µA | +125ºC | | |
| | 0.01 | 0.54 | µA | -40°C | 3.3V | |
| | 0.03 | 0.54 | µA | +25°C | | |
| | 0.08 | 1.68 | µA | +60°C | | |
| | 0.25 | 2.45 | µA | +85°C | | |
| | — | 10.00 | µA | +125ºC | | |

**Note 1:** Data in the Typical column is at 3.3V, +25°C unless otherwise stated.

**2:** Base $I_{PD}$ is measured with all peripherals and clocks disabled. All I/Os are configured as outputs and set low; PMDx bits are set to '1' and WDT, etc., are all disabled

**FIGURE 26-14: MSSPx I²C™ BUS DATA TIMING**




**Note:** Refer to Figure 26-3 for load conditions.

**TABLE 26-34: I²C™ BUS DATA REQUIREMENTS (MASTER MODE)**

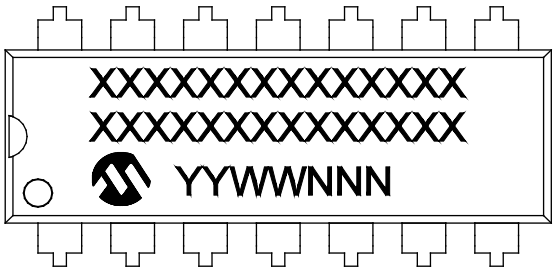
| Param. No. | Symbol | Characteristic | | Min | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| 100 | THIGH | Clock High Time | 100 kHz mode | 2(TOSC)(BRG + 1) | — | — | |
| | | | 400 kHz mode | 2(TOSC)(BRG + 1) | — | — | |
| 101 | TLOW | Clock Low Time | 100 kHz mode | 2(TOSC)(BRG + 1) | — | — | |
| | | | 400 kHz mode | 2(TOSC)(BRG + 1) | — | — | |
| 102 | TR | SDAx and SCLx Rise Time | 100 kHz mode | — | 1000 | ns | CB is specified to be from 10 to 400 pF |
| | | | 400 kHz mode | 20 + 0.1 CB | 300 | ns | |
| 103 | TF | SDAx and SCLx Fall Time | 100 kHz mode | — | 300 | ns | CB is specified to be from 10 to 400 pF |
| | | | 400 kHz mode | 20 + 0.1 CB | 300 | ns | |
| 90 | TSU:STA | Start Condition Setup Time | 100 kHz mode | 2(TOSC)(BRG + 1) | — | — | Only relevant for Repeated Start condition |
| | | | 400 kHz mode | 2(TOSC)(BRG + 1) | — | — | |
| 91 | THD:STA | Start Condition Hold Time | 100 kHz mode | 2(TOSC)(BRG + 1) | — | — | After this period, the first clock pulse is generated |
| | | | 400 kHz mode | 2(TOSC)(BRG + 1) | — | — | |
| 106 | THD:DAT | Data Input Hold Time | 100 kHz mode | 0 | — | ns | |
| | | | 400 kHz mode | 0 | 0.9 | μs | |
| 107 | TSU:DAT | Data Input Setup Time | 100 kHz mode | 250 | — | ns | **(Note 1)** |
| | | | 400 kHz mode | 100 | — | ns | |
| 92 | TSU:STO | Stop Condition Setup Time | 100 kHz mode | 2(TOSC)(BRG + 1) | — | — | |
| | | | 400 kHz mode | 2(TOSC)(BRG + 1) | — | — | |
| 109 | TAA | Output Valid from Clock | 100 kHz mode | — | 3500 | ns | |
| | | | 400 kHz mode | — | 1000 | ns | |
| 110 | TBUF | Bus Free Time | 100 kHz mode | 4.7 | — | μs | Time the bus must be free before a new transmission can start |
| | | | 400 kHz mode | 1.3 | — | μs | |
| D102 | CB | Bus Capacitive Loading | | — | 400 | pF | |

**Note 1:** A Fast mode I²C bus device can be used in a Standard mode I²C bus system, but Parameter 107 ≥ 250 ns must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCLx signal. If such a device does stretch the LOW period of the SCLx signal, it must output the next data bit to the SDAx line, Parameter 102 + Parameter 107 = 1000 + 250 = 1250 ns (for 100 kHz mode), before the SCLx line is released.

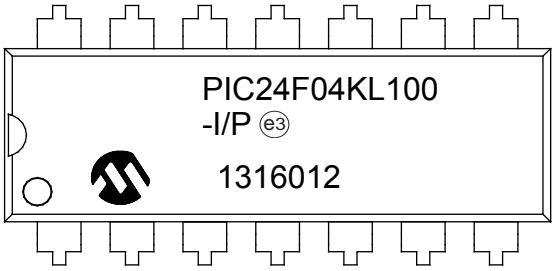
## 27.0 PACKAGING INFORMATION

### 27.1 Package Marking Information

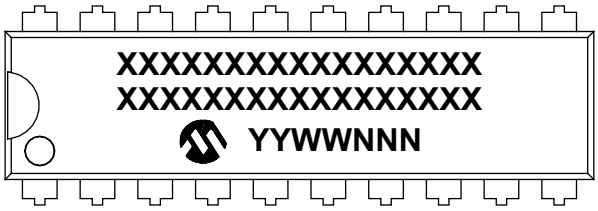
14-Lead PDIP (300 mil)



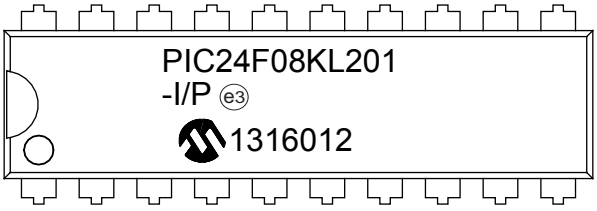

Example



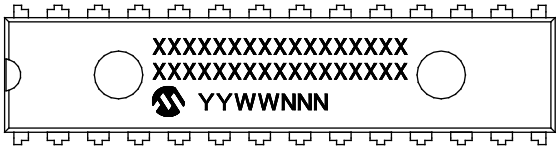

20-Lead PDIP (300 mil)



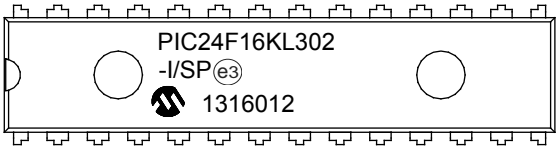

Example




28-Lead SPDIP (.300")




Example




| | | |
|---|---|---|
| **Legend:** | XX...X | Product-specific information |
| | Y | Year code (last digit of calendar year) |
| | YY | Year code (last 2 digits of calendar year) |
| | WW | Week code (week of January 1 is week '01') |
| | NNN | Alphanumeric traceability code |
| | (e3) | Pb-free JEDEC designator for Matte Tin (Sn) |
| | * | This package is Pb-free. The Pb-free JEDEC designator ((e3)) can be found on the outer packaging for this package. |

**Note:** In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

**NOTES:**