**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 | 23 |
| Program Memory Size | 32KB (11K x 24) |
| Program Memory Type | FLASH |
| EEPROM Size | 512 x 8 |
| RAM Size | 2K x 8 |
| Voltage - Supply (Vcc/Vdd) | 2V ~ 5.5V |
| Data Converters | A/D 13x12b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 125°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 28-SSOP (0.209", 5.30mm Width) |
| Supplier Device Package | 28-SSOP |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic24fv32ka302-e-ss |

**EXAMPLE 5-2: ERASING A PROGRAM MEMORY ROW – 'C' LANGUAGE CODE**

```
// C example using MPLAB C30

    int __attribute__ ((space(auto_psv))) progAddr = 0x1234;// Global variable located in Pgm Memory
    unsigned int offset;

//Set up pointer to the first memory location to be written

    TBLPAG = __builtin_tblpage(&progAddr);              // Initialize PM Page Boundary SFR
    offset = __builtin_tbloffset(&progAddr);            // Initialize lower word of address

    __builtin_tblwtl(offset, 0x0000);                   // Set base address of erase block
                                                        // with dummy latch write

    NVMCON = 0x4058;                                    // Initialize NVMCON

    asm("DISI #5");                                     // Block all interrupts for next 5
                                                        // instructions
    __builtin_write_NVM();                              // C30 function to perform unlock
                                                        // sequence and set WR
```

**EXAMPLE 5-3: LOADING THE WRITE BUFFERS – ASSEMBLY LANGUAGE CODE**

```
; Set up NVMCON for row programming operations
       MOV     #0x4004, W0                  ;
       MOV     W0, NVMCON                   ; Initialize NVMCON
; Set up a pointer to the first program memory location to be written
; program memory selected, and writes enabled
       MOV     #0x0000, W0                  ;
       MOV     W0, TBLPAG                   ; Initialize PM Page Boundary SFR
       MOV     #0x6000, W0                  ; An example program memory address
; Perform the TBLWT instructions to write the latches
; 0th_program_word
       MOV     #LOW_WORD_0, W2              ;
       MOV     #HIGH_BYTE_0, W3             ;
       TBLWTL  W2, [W0]                     ; Write PM low word into program latch
       TBLWTH  W3, [W0++]                   ; Write PM high byte into program latch
; 1st_program_word
       MOV     #LOW_WORD_1, W2              ;
       MOV     #HIGH_BYTE_1, W3             ;
       TBLWTL  W2, [W0]                     ; Write PM low word into program latch
       TBLWTH  W3, [W0++]                   ; Write PM high byte into program latch
;  2nd_program_word
       MOV     #LOW_WORD_2, W2              ;
       MOV     #HIGH_BYTE_2, W3             ;
       TBLWTL  W2, [W0]                     ; Write PM low word into program latch
       TBLWTH  W3, [W0++]                   ; Write PM high byte into program latch
       •
       •
       •
; 32nd_program_word
       MOV     #LOW_WORD_31, W2             ;
       MOV     #HIGH_BYTE_31, W3            ;
       TBLWTL  W2, [W0]                     ; Write PM low word into program latch
       TBLWTH  W3, [W0]                     ; Write PM high byte into program latch
```

**REGISTER 8-23: IPC6: INTERRUPT PRIORITY CONTROL REGISTER 6**

| U-0 | R/W-1 | R/W-0 | R/W-0 | U-0 | U-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| — | T4IP2 | T4IP1 | T4IP0 | — | — | — | — |
| bit 15 | | | | | | | bit 8 |

| U-0 | R/W-1 | R/W-0 | R/W-0 | U-0 | U-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| — | OC3IP2 | OC3IP1 | OC3IP0 | — | — | — | — |
| 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 **Unimplemented:** Read as '0'

bit 14-12 **T4IP<2:0>:** Timer4 Interrupt Priority bits
111 = Interrupt is Priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is Priority 1
000 = Interrupt source is disabled

bit 11-7 **Unimplemented:** Read as '0'

bit 6-4 **OC3IP<2:0>:** Output Compare Channel 3 Interrupt Priority bits
111 = Interrupt is Priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is Priority 1
000 = Interrupt source is disabled

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

**REGISTER 9-3: OSCTUN: FRC OSCILLATOR TUNE REGISTER**

| U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| — | — | — | — | — | — | — | — |
| bit 15 | | | | | | | bit 8 |

| U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | TUN5(1) | TUN4(1) | TUN3(1) | TUN2(1) | TUN1(1) | TUN0(1) |
| 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 15-6 **Unimplemented:** Read as '0'

bit 5-0 **TUN<5:0>:** FRC Oscillator Tuning bits(1)

011111 = Maximum frequency deviation
011110
•
•
•
000001
000000 = Center frequency, oscillator is running at factory calibrated frequency
111111
•
•
•
100001
100000 = Minimum frequency deviation

**Note 1:** Increments or decrements of TUN<5:0> may not change the FRC frequency in equal steps over the FRC tuning range and may not be monotonic.

## 12.0 TIMER1

| | |
|---|---|
| **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 *"PIC24F Family Reference Manual"*, **Section 14. "Timers"** (DS39704). |

The Timer1 module is a 16-bit timer which can serve as the time counter for the Real-Time Clock (RTC) or operate as a free-running, interval timer/counter. Timer1 can operate in three modes:

- 16-Bit Timer
- 16-Bit Synchronous Counter
- 16-Bit Asynchronous Counter

Timer1 also supports these features:

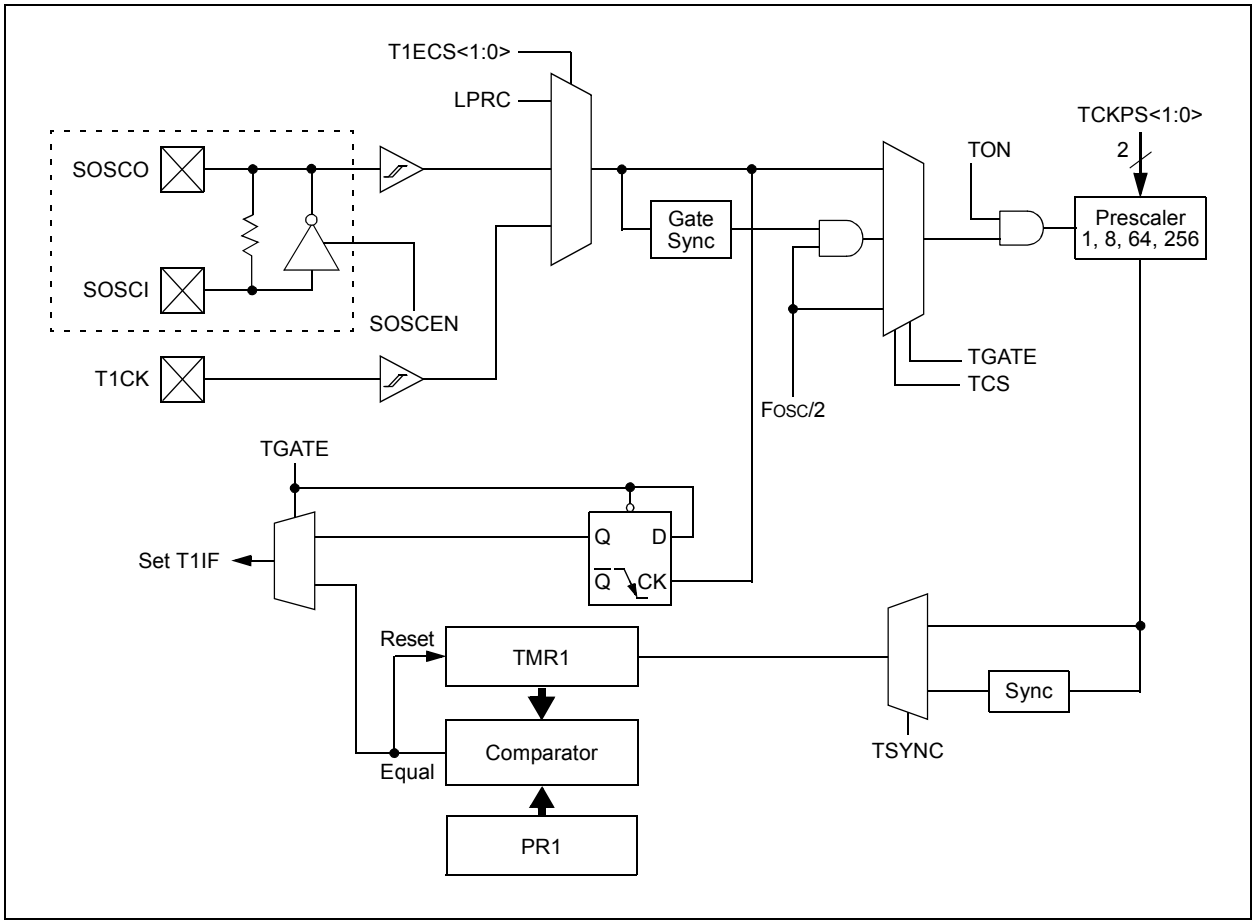
- Timer Gate Operation
- Selectable Prescaler Settings
- Timer Operation During CPU Idle and Sleep modes
- Interrupt on 16-Bit Period Register Match or Falling Edge of External Gate Signal

Figure 12-1 illustrates a block diagram of the 16-bit Timer1 module.

To configure Timer1 for operation:

1. Set the TON bit (= 1).
2. Select the timer prescaler ratio using the TCKPS<1:0> bits.
3. Set the Clock and Gating modes using the TCS and TGATE bits.
4. Set or clear the TSYNC bit to configure synchronous or asynchronous operation.
5. Load the timer period value into the PR1 register.
6. If interrupts are required, set the Timer1 Interrupt Enable bit, T1IE. Use the Timer1 Interrupt Priority bits, T1IP<2:0>, to set the interrupt priority.

**FIGURE 12-1: 16-BIT TIMER1 MODULE BLOCK DIAGRAM**



**REGISTER 13-2: TyCON: TIMER3 AND TIMER5 CONTROL REGISTER**

| R/W-0 | U-0 | R/W-0 | U-0 | U-0 | U-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| TON(1) | — | TSIDL(1) | — | — | — | — | — |
| bit 15 | | | | | | | bit 8 |

| U-0 | R/W-0 | R/W-0 | R/W-0 | U-0 | U-0 | R/W-0 | U-0 |
|---|---|---|---|---|---|---|---|
| — | TGATE(1) | TCKPS1(1) | TCKPS0(1) | — | — | TCS(1) | — |
| 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 **TON:** Timery On bit(1)
1 = Starts 16-bit Timery
0 = Stops 16-bit Timery

bit 14 **Unimplemented:** Read as '0'

bit 13 **TSIDL:** Timery Stop in Idle Mode bit(1)
1 = Discontinues module operation when device enters Idle mode
0 = Continues module operation in Idle mode

bit 12-7 **Unimplemented:** Read as '0'

bit 6 **TGATE:** Timery Gated Time Accumulation Enable bit(1)
When TCS = 1:
This bit is ignored.
When TCS = 0:
1 = Gated time accumulation is enabled
0 = Gated time accumulation is disabled

bit 5-4 **TCKPS<1:0>:** Timery Input Clock Prescale Select bits(1)
11 = 1:256
10 = 1:64
01 = 1:8
00 = 1:1

bit 3-2 **Unimplemented:** Read as '0'

bit 1 **TCS:** Timery Clock Source Select bit(1)
1 = External clock is from the T3CK pin (on the rising edge)
0 = Internal clock (FOSC/2)

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

**Note 1:** When 32-bit operation is enabled (TxCON<3> = 1), these bits have no effect on Timery operation. All timer functions are set through the TxCON register.

**REGISTER 14-1: ICxCON1: INPUT CAPTURE x CONTROL REGISTER 1**

| U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| — | — | ICSIDL | ICTSEL2 | ICTSEL1 | ICTSEL0 | — | — |
| bit 15 | | | | | | | bit 8 |

| U-0 | R/W-0 | R/W-0 | R-0, HSC | R-0, HSC | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | ICI1 | ICI0 | ICOV | ICBNE | ICM2 | ICM1 | ICM0 |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | HSC = Hardware Settable/Clearable bit | | |
|---|---|---|---|
| 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 15-14 **Unimplemented:** Read as '0'

bit 13 **ICSIDL:** Input Capture x Module Stop in Idle Control bit
1 = Input capture module halts in CPU Idle mode
0 = Input capture module continues to operate in CPU Idle mode

bit 12-10 **ICTSEL<2:0>:** Input Capture x Timer Select bits
111 = System clock (FOSC/2)
110 = Reserved
101 = Reserved
100 = Timer1
011 = Timer5
010 = Timer4
001 = Timer2
000 = Timer3

bit 9-7 **Unimplemented:** Read as '0'

bit 6-5 **ICI<1:0>:** Select Number of Captures per Interrupt bits
11 = Interrupt on every fourth capture event
10 = Interrupt on every third capture event
01 = Interrupt on every second capture event
00 = Interrupt on every capture event

bit 4 **ICOV:** Input Capture x Overflow Status Flag bit (read-only)
1 = Input capture overflow occurred
0 = No input capture overflow occurred

bit 3 **ICBNE:** Input Capture x Buffer Empty Status bit (read-only)
1 = Input capture buffer is not empty, at least one more capture value can be read
0 = Input capture buffer is empty

bit 2-0 **ICM<2:0>:** Input Capture Mode Select bits
111 = Interrupt mode: Input capture functions as an interrupt pin only when the device is in Sleep or Idle mode (rising edge detect only, all other control bits are not applicable)
110 = Unused (module disabled)
101 = Prescaler Capture mode: Capture on every 16th rising edge
100 = Prescaler Capture mode: Capture on every 4th rising edge
011 = Simple Capture mode: Capture on every rising edge
010 = Simple Capture mode: Capture on every falling edge
001 = Edge Detect Capture mode: Capture on every edge (rising and falling); ICI<1:0 bits do not control interrupt generation for this mode
000 = Input capture module is turned off

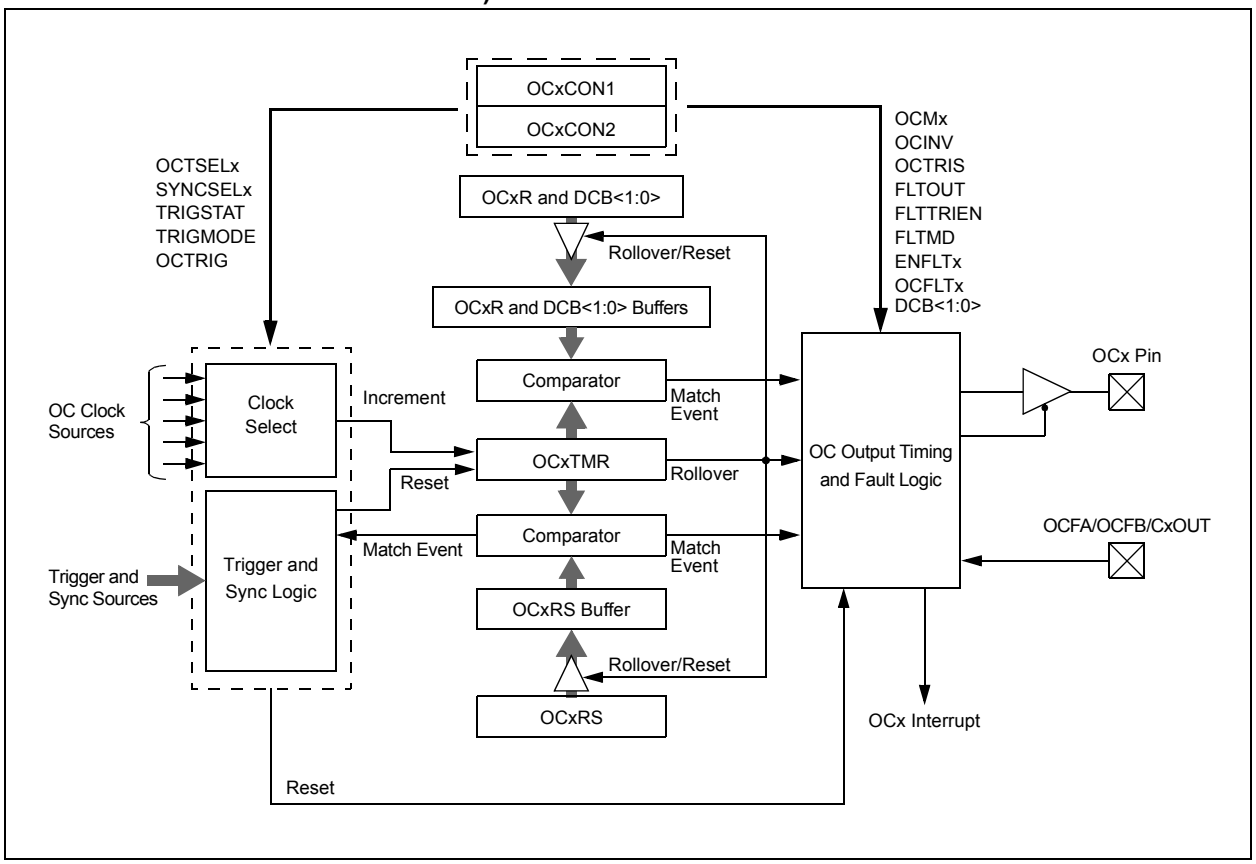
## 15.3 Pulse-Width Modulation (PWM) Mode

In PWM mode, the output compare module can be configured for edge-aligned or center-aligned pulse waveform generation. All PWM operations are double-buffered (buffer registers are internal to the module and are not mapped into SFR space).

To configure the output compare module for edge-aligned PWM operation:

1. Calculate the desired ON time and load it into the OCxR register.
2. Calculate the desired period and load it into the OCxRS register.
3. Select the current OCx as the synchronization source by writing 0x1F to SYNCSEL<4:0> (OCxCON2<4:0>) and '0' to OCTRIG (OCxCON2<7>).
4. Select a clock source by writing the OCTSEL2<2:0> (OCxCON<12:10>) bits.
5. Enable interrupts, if required, for the timer and output compare modules. The output compare interrupt is required for PWM Fault pin utilization.
6. Select the desired PWM mode in the OCM<2:0> (OCxCON1<2:0>) bits.
7. If a timer is selected as a clock source, set the TMRy prescale value and enable the time base by setting the TON (TxCON<15>) bit.

**FIGURE 15-2: OUTPUT COMPARE x BLOCK DIAGRAM (DOUBLE-BUFFERED, 16-BIT PWM MODE)**



## 16.0 SERIAL PERIPHERAL INTERFACE (SPI)

**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 Serial Peripheral Interface, refer to the *"PIC24F Family Reference Manual"*, **Section 23. "Serial Peripheral Interface (SPI)"** (DS39699).

The Serial Peripheral Interface (SPI) module is a synchronous serial interface useful for communicating with other peripheral or microcontroller devices. These peripheral devices may be serial data EEPROMs, shift registers, display drivers, A/D Converters, etc. The SPI module is compatible with Motorola® SPI and SIOP interfaces.

The module supports operation in two buffer modes. In Standard mode, data is shifted through a single serial buffer. In Enhanced Buffer mode, data is shifted through an 8-level FIFO buffer.

**Note:** Do not perform read-modify-write operations (such as bit-oriented instructions) on the SPI1BUF register in either Standard or Enhanced Buffer mode.

The module also supports a basic framed SPI protocol while operating in either Master or Slave mode. A total of four framed SPI configurations are supported.

The SPI serial interface consists of four pins:

- SDI1: Serial Data Input
- SDO1: Serial Data Output
- SCK1: Shift Clock Input or Output
- $\overline{SS1}$: Active-Low Slave Select or Frame Synchronization I/O Pulse

The SPI module can be configured to operate using 2, 3 or 4 pins. In the 3-pin mode, $\overline{SS1}$ is not used. In the 2-pin mode, both SDO1 and $\overline{SS1}$ are not used.

Block diagrams of the module, in Standard and Enhanced Buffer modes, are shown in Figure 16-1 and Figure 16-2.

The devices of the PIC24FV32KA304 family offer two SPI modules on a device.

**Note:** In this section, the SPI modules are referred to as SPIx. Special Function Registers (SFRs) will follow a similar notation. For example, SPI1CON1 or SPI1CON2 refers to the control register for the SPI1 module.

To set up the SPI1 module for the Standard Master mode of operation:

1. If using interrupts:
   a) Clear the SPI1IF bit in the IFS0 register.
   b) Set the SPI1IE bit in the IEC0 register.
   c) Write the respective SPI1IPx bits in the IPC2 register to set the interrupt priority.
2. Write the desired settings to the SPI1CON1 and SPI1CON2 registers with the MSTEN bit (SPI1CON1<5>) = 1.
3. Clear the SPIROV bit (SPI1STAT<6>).
4. Enable SPI operation by setting the SPIEN bit (SPI1STAT<15>).
5. Write the data to be transmitted to the SPI1BUF register. Transmission (and reception) will start as soon as data is written to the SPI1BUF register.

To set up the SPI1 module for the Standard Slave mode of operation:

1. Clear the SPI1BUF register.
2. If using interrupts:
   a) Clear the SPI1IF bit in the IFS0 register.
   b) Set the SPI1IE bit in the IEC0 register.
   c) Write the respective SPI1IPx bits in the IPC2 register to set the interrupt priority.
3. Write the desired settings to the SPI1CON1 and SPI1CON2 registers with the MSTEN bit (SPI1CON1<5>) = 0.
4. Clear the SMP bit.
5. If the CKE bit is set, then the SSEN bit (SPI1CON1<7>) must be set to enable the $\overline{SS1}$ pin.
6. Clear the SPIROV bit (SPI1STAT<6>).
7. Enable SPI operation by setting the SPIEN bit (SPI1STAT<15>).

**REGISTER 20-3: CRCXORL: CRC XOR POLYNOMIAL REGISTER, LOW BYTE**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| X15 | X14 | X13 | X12 | X11 | X10 | X9 | X8 |
| bit 15 | | | | | | | bit 8 |

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | U-0 |
|---|---|---|---|---|---|---|---|
| X7 | X6 | X5 | X4 | X3 | X2 | X1 | — |
| 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 15-1 **X<15:1>:** XOR of Polynomial Term $X^n$ Enable bits

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

**REGISTER 20-4: CRCXORH: CRC XOR POLYNOMIAL REGISTER, HIGH BYTE**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| X31 | X30 | X29 | X28 | X27 | X26 | X25 | X24 |
| bit 15 | | | | | | | bit 8 |

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| X23 | X22 | X21 | X20 | X19 | X18 | X17 | X16 |
| bit 7 | | | | | | | bit 0 |

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

bit 15-0 **X<31:16>:** XOR of Polynomial Term $X^n$ Enable bits

**REGISTER 22-1: AD1CON1: A/D CONTROL REGISTER 1**

| R/W-0 | U-0 | R/W-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| ADON | — | ADSIDL | — | — | MODE12 | FORM1 | FORM0 |
| bit 15 | | | | | | | bit 8 |

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | U-0 | R/W-0 | R/W-0, HSC | R/C-0, HSC |
|---|---|---|---|---|---|---|---|
| SSRC3 | SSRC2 | SSRC1 | SSRC0 | — | ASAM | SAMP | DONE |
| bit 7 | | | | | | | bit 0 |

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

bit 15 **ADON:** A/D Operating Mode bit
1 = A/D Converter module is operating
0 = A/D Converter is off

bit 14 **Unimplemented:** Read as '0'

bit 13 **ADSIDL:** A/D Stop in Idle Mode bit
1 = Discontinues module operation when device enters Idle mode
0 = Continues module operation in Idle mode

bit 12-11 **Unimplemented:** Read as '0'

bit 10 **MODE12:** 12-Bit Operation Mode bit
1 = 12-bit A/D operation
0 = 10-bit A/D operation

bit 9-8 **FORM<1:0>:** Data Output Format bits (see the following formats)
11 = Fractional result, signed, left-justified
10 = Absolute fractional result, unsigned, left-justified
01 = Decimal result, signed, right-justified
00 = Absolute decimal result, unsigned, right-justified

bit 7-4 **SSRC<3:0>:** Sample Clock Source Select bits
1111 = Not available; do not use
•
•
•
1000 = Not available; do not use
0111 = Internal counter ends sampling and starts conversion (auto-convert)
0110 = Not available; do not use
0101 = Timer1 event ends sampling and starts conversion
0100 = CTMU event ends sampling and starts conversion
0011 = Timer5 event ends sampling and starts conversion
0010 = Timer3 event ends sampling and starts conversion
0001 = INT0 event ends sampling and starts conversion
0000 = Clearing the SAMP bit in software ends sampling and begins conversion

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

bit 2 **ASAM:** A/D Sample Auto-Start bit
1 = Sampling begins immediately after the last conversion; SAMP bit is auto-set
0 = Sampling begins when the SAMP bit is manually set

bit 1 **SAMP:** A/D Sample Enable bit
1 = A/D Sample-and-Hold amplifiers are sampling
0 = A/D Sample-and-Hold amplifiers are holding

bit 0 **DONE:** A/D Conversion Status bit
1 = A/D conversion cycle has completed
0 = A/D conversion cycle has not started or is in progress

**REGISTER 22-2: AD1CON2: A/D CONTROL REGISTER 2 (CONTINUED)**

bit 0 **ALTS:** Alternate Input Sample Mode Select bit
1 = Uses channel input selects for Sample A on the first sample and Sample B on the next sample
0 = Always uses channel input selects for Sample A

**Note 1:** This is only applicable when the buffer is used in FIFO mode (BUFREGEN = 0). In addition, BUFS is only used when BUFM = 1.
**2:** The voltage reference setting will not be within the specification with VDD below 4.5V.
**3:** The voltage reference setting will not be within the specification with VDD below 2.3V.

**REGISTER 22-3: AD1CON3: A/D CONTROL REGISTER 3**

| R/W-0 | R-0 | r-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| ADRC | EXTSAM | r | SAMC4 | SAMC3 | SAMC2 | SAMC1 | SAMC0 |
| bit 15 | | | | | | | bit 8 |

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| ADCS7 | ADCS6 | ADCS5 | ADCS4 | ADCS3 | ADCS2 | ADCS1 | ADCS0 |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | r = Reserved bit | | |
|---|---|---|---|
| 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 **ADRC:** A/D Conversion Clock Source bit
1 = RC clock
0 = Clock is derived from the system clock

bit 14 **EXTSAM:** Extended Sampling Time bit
1 = A/D is still sampling after SAMP = 0
0 = A/D is finished sampling

bit 13 **Reserved:** Maintain as '0'

bit 12-8 **SAMC<4:0>:** Auto-Sample Time Select bits
11111 = 31 TAD
•
•
•
00001 = 1 TAD
00000 = 0 TAD

bit 7-0 **ADCS<7:0>:** A/D Conversion Clock Select bits
11111111-01000000 = Reserved
00111111 = 64 · TCY = TAD
•
•
•
00000001 = 2 · TCY = TAD
00000000 = TCY = TAD

**REGISTER 22-10: AD1CTMUENH: A/D CTMU ENABLE REGISTER (HIGH WORD)(1)**

| U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| — | — | — | — | — | — | — | — |
| bit 15 | | | | | | | bit 8 |

| U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | — | — | — | — | CTMEN17 | CTMEN16 |
| 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 15-2 **Unimplemented:** Read as '0'.

bit 1-0 **CTMEN<17:16>:** CTMU Enabled During Conversion bits

1 = CTMU is enabled and connected to the selected channel during conversion
0 = CTMU is not connected to this channel

**Note 1:** Unimplemented channels are read as '0'.

**REGISTER 22-11: AD1CTMUENL: A/D CTMU ENABLE REGISTER (LOW WORD)(1)**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| CTMEN15 | CTMEN14 | CTMEN13 | CTMEN12 | CTMUEN11 | CTMEN10 | CTMEN9 | CTMEN8 |
| bit 15 | | | | | | | bit 8 |

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| CTMEN7 | CTMEN6 | CTMEN5 | CTMEN4 | CTMEN3 | CTMEN2 | CTMEN1 | CTMEN0 |
| bit 7 | | | | | | | bit 0 |

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

bit 15-0 **CTMEN<15:0>:** CTMU Enabled During Conversion bits

1 = CTMU is enabled and connected to the selected channel during conversion
0 = CTMU is not connected to this channel

**Note 1:** Unimplemented channels are read as '0'.

**TABLE 28-2: INSTRUCTION SET OVERVIEW (CONTINUED)**

| Assembly Mnemonic | Assembly Syntax | | Description | # of Words | # of Cycles | Status Flags Affected |
|---|---|---|---|---|---|---|
| PWRSAV | PWRSAV | #lit1 | Go into Sleep or Idle mode | 1 | 1 | WDTO, Sleep |
| RCALL | RCALL | Expr | Relative Call | 1 | 2 | None |
| | RCALL | Wn | Computed Call | 1 | 2 | None |
| REPEAT | REPEAT | #lit14 | Repeat Next Instruction lit14 + 1 times | 1 | 1 | None |
| | REPEAT | Wn | Repeat Next Instruction (Wn) + 1 times | 1 | 1 | None |
| RESET | RESET | | Software Device Reset | 1 | 1 | None |
| RETFIE | RETFIE | | Return from Interrupt | 1 | 3 (2) | None |
| RETLW | RETLW | #lit10,Wn | Return with Literal in Wn | 1 | 3 (2) | None |
| RETURN | RETURN | | Return from Subroutine | 1 | 3 (2) | None |
| RLC | RLC | f | f = Rotate Left through Carry f | 1 | 1 | C, N, Z |
| | RLC | f,WREG | WREG = Rotate Left through Carry f | 1 | 1 | C, N, Z |
| | RLC | Ws,Wd | Wd = Rotate Left through Carry Ws | 1 | 1 | C, N, Z |
| RLNC | RLNC | f | f = Rotate Left (No Carry) f | 1 | 1 | N, Z |
| | RLNC | f,WREG | WREG = Rotate Left (No Carry) f | 1 | 1 | N, Z |
| | RLNC | Ws,Wd | Wd = Rotate Left (No Carry) Ws | 1 | 1 | N, Z |
| RRC | RRC | f | f = Rotate Right through Carry f | 1 | 1 | C, N, Z |
| | RRC | f,WREG | WREG = Rotate Right through Carry f | 1 | 1 | C, N, Z |
| | RRC | Ws,Wd | Wd = Rotate Right through Carry Ws | 1 | 1 | C, N, Z |
| RRNC | RRNC | f | f = Rotate Right (No Carry) f | 1 | 1 | N, Z |
| | RRNC | f,WREG | WREG = Rotate Right (No Carry) f | 1 | 1 | N, Z |
| | RRNC | Ws,Wd | Wd = Rotate Right (No Carry) Ws | 1 | 1 | N, Z |
| SE | SE | Ws,Wnd | Wnd = Sign-Extended Ws | 1 | 1 | C, N, Z |
| SETM | SETM | f | f = FFFFh | 1 | 1 | None |
| | SETM | WREG | WREG = FFFFh | 1 | 1 | None |
| | SETM | Ws | Ws = FFFFh | 1 | 1 | None |
| SL | SL | f | f = Left Shift f | 1 | 1 | C, N, OV, Z |
| | SL | f,WREG | WREG = Left Shift f | 1 | 1 | C, N, OV, Z |
| | SL | Ws,Wd | Wd = Left Shift Ws | 1 | 1 | C, N, OV, Z |
| | SL | Wb,Wns,Wnd | Wnd = Left Shift Wb by Wns | 1 | 1 | N, Z |
| | SL | Wb,#lit5,Wnd | Wnd = Left Shift Wb by lit5 | 1 | 1 | N, Z |
| SUB | SUB | f | f = f – WREG | 1 | 1 | C, DC, N, OV, Z |
| | SUB | f,WREG | WREG = f – WREG | 1 | 1 | C, DC, N, OV, Z |
| | SUB | #lit10,Wn | Wn = Wn – lit10 | 1 | 1 | C, DC, N, OV, Z |
| | SUB | Wb,Ws,Wd | Wd = Wb – Ws | 1 | 1 | C, DC, N, OV, Z |
| | SUB | Wb,#lit5,Wd | Wd = Wb – lit5 | 1 | 1 | C, DC, N, OV, Z |
| SUBB | SUBB | f | f = f – WREG – ($\overline{C}$) | 1 | 1 | C, DC, N, OV, Z |
| | SUBB | f,WREG | WREG = f – WREG – ($\overline{C}$) | 1 | 1 | C, DC, N, OV, Z |
| | SUBB | #lit10,Wn | Wn = Wn – lit10 – ($\overline{C}$) | 1 | 1 | C, DC, N, OV, Z |
| | SUBB | Wb,Ws,Wd | Wd = Wb – Ws – ($\overline{C}$) | 1 | 1 | C, DC, N, OV, Z |
| | SUBB | Wb,#lit5,Wd | Wd = Wb – lit5 – ($\overline{C}$) | 1 | 1 | C, DC, N, OV, Z |
| SUBR | SUBR | f | f = WREG – f | 1 | 1 | C, DC, N, OV, Z |
| | SUBR | f,WREG | WREG = WREG – f | 1 | 1 | C, DC, N, OV, Z |
| | SUBR | Wb,Ws,Wd | Wd = Ws – Wb | 1 | 1 | C, DC, N, OV, Z |
| | SUBR | Wb,#lit5,Wd | Wd = lit5 – Wb | 1 | 1 | C, DC, N, OV, Z |
| SUBBR | SUBBR | f | f = WREG – f – ($\overline{C}$) | 1 | 1 | C, DC, N, OV, Z |
| | SUBBR | f,WREG | WREG = WREG – f – ($\overline{C}$) | 1 | 1 | C, DC, N, OV, Z |
| | SUBBR | Wb,Ws,Wd | Wd = Ws – Wb – ($\overline{C}$) | 1 | 1 | C, DC, N, OV, Z |
| | SUBBR | Wb,#lit5,Wd | Wd = lit5 – Wb – ($\overline{C}$) | 1 | 1 | C, DC, N, OV, Z |
| SWAP | SWAP.b | Wn | Wn = Nibble Swap Wn | 1 | 1 | None |
| | SWAP | Wn | Wn = Byte Swap Wn | 1 | 1 | None |
| TBLRDH | TBLRDH | Ws,Wd | Read Prog<23:16> to Wd<7:0> | 1 | 2 | None |

**TABLE 28-2: INSTRUCTION SET OVERVIEW (CONTINUED)**

| Assembly Mnemonic | Assembly Syntax | | Description | # of Words | # of Cycles | Status Flags Affected |
|---|---|---|---|---|---|---|
| TBLRDL | TBLRDL | Ws,Wd | Read Prog<15:0> to Wd | 1 | 2 | None |
| TBLWTH | TBLWTH | Ws,Wd | Write Ws<7:0> to Prog<23:16> | 1 | 2 | None |
| TBLWTL | TBLWTL | Ws,Wd | Write Ws to Prog<15:0> | 1 | 2 | None |
| ULNK | ULNK | | Unlink Frame Pointer | 1 | 1 | None |
| XOR | XOR | f | f = f .XOR. WREG | 1 | 1 | N, Z |
| | XOR | f,WREG | WREG = f .XOR. WREG | 1 | 1 | N, Z |
| | XOR | #lit10,Wn | Wd = lit10 .XOR. Wd | 1 | 1 | N, Z |
| | XOR | Wb,Ws,Wd | Wd = Wb .XOR. Ws | 1 | 1 | N, Z |
| | XOR | Wb,#lit5,Wd | Wd = Wb .XOR. lit5 | 1 | 1 | N, Z |
| ZE | ZE | Ws,Wnd | Wnd = Zero-Extend Ws | 1 | 1 | C, Z, N |

## 29.0 ELECTRICAL CHARACTERISTICS

This section provides an overview of the PIC24FV32KA304 family electrical characteristics. Additional information will be provided in future revisions of this document as it becomes available.

Absolute maximum ratings for the PIC24FV32KA304 family devices are listed below. Exposure to these maximum rating conditions for extended periods may affect device reliability. Functional operation of the device at these, or any other conditions above the parameters indicated in the operation listings of this specification, is not implied.

### Absolute Maximum Ratings(†)

Ambient temperature under bias ..... -40°C to +135°C
Storage temperature ..... -65°C to +150°C
Voltage on $V_{DD}$ with respect to $V_{SS}$ (PIC24FVXXKA30X) ..... -0.3V to +6.5V
Voltage on $V_{DD}$ with respect to $V_{SS}$ (PIC24FXXKA30X) ..... -0.3V to +4.5V
Voltage on any combined analog and digital pin with respect to $V_{SS}$ ..... -0.3V to ($V_{DD}$ + 0.3V)
Voltage on any digital only pin with respect to $V_{SS}$ ..... -0.3V to ($V_{DD}$ + 0.3V)
Voltage on $\overline{MCLR}/V_{PP}$ pin with respect to $V_{SS}$ ..... -0.3V to +9.0V
Maximum current out of $V_{SS}$ pin ..... 300 mA
Maximum current into $V_{DD}$ pin(1) ..... 250 mA
Maximum output current sunk by any I/O pin ..... 25 mA
Maximum output current sourced by any I/O pin ..... 25 mA
Maximum current sunk by all ports ..... 200 mA
Maximum current sourced by all ports(1) ..... 200 mA

**Note 1:** Maximum allowable current is a function of the device maximum power dissipation (see Table 29-1).

† **NOTICE:** Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

**TABLE 29-7: DC CHARACTERISTICS: IDLE CURRENT (IIDLE)**

| DC CHARACTERISTICS | | Standard Operating Conditions: 1.8V to 3.6V PIC24F32KA3XX<br>2.0V to 5.5V PIC24FV32KA3XX<br>Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial<br>-40°C ≤ TA ≤ +125°C for Extended | | | | |
|---|---|---|---|---|---|---|
| **Parameter No.** | **Device** | **Typical** | **Max** | **Units** | **Conditions** | |
| **Idle Current (IIDLE)** | | | | | | |
| DC40 | PIC24FV32KA3XX | 120 | 200 | µA | 2.0V | 0.5 MIPS, FOSC = 1 MHz(1) |
| | | 160 | 430 | µA | 5.0V | |
| | PIC24F32KA3XX | 50 | 100 | µA | 1.8V | |
| | | 90 | 370 | µA | 3.3V | |
| DC42 | PIC24FV32KA3XX | 165 | — | µA | 2.0V | 1 MIPS, FOSC = 2 MHz(1) |
| | | 260 | — | µA | 5.0V | |
| | PIC24F32KA3XX | 95 | — | µA | 1.8V | |
| | | 180 | — | µA | 3.3V | |
| DC44 | PIC24FV32KA3XX | 3.1 | 6.5 | mA | 5.0V | 16 MIPS, FOSC = 32 MHz(1) |
| | PIC24F32KA3XX | 2.9 | 6.0 | mA | 3.3V | |
| DC46 | PIC24FV32KA3XX | 0.65 | — | mA | 2.0V | FRC (4 MIPS), FOSC = 8 MHz |
| | | 1.0 | — | mA | 5.0V | |
| | PIC24F32KA3XX | 0.55 | — | mA | 1.8V | |
| | | 1.0 | — | mA | 3.3V | |
| DC50 | PIC24FV32KA3XX | 60 | 200 | µA | 2.0V | LPRC (15.5 KIPS), FOSC = 31 kHz |
| | | 70 | 350 | µA | 5.0V | |
| | PIC24F32KA3XX | 2.2 | 18 | µA | 1.8V | |
| | | 4.0 | 60 | µA | 3.3V | |

**Legend:** Unshaded rows represent PIC24F32KA3XX devices and shaded rows represent PIC24FV32KA3XX devices.

**Note 1:** Oscillator is in External Clock mode (FOSCSEL<2:0> = 010, FOSC<1:0> = 00).

**FIGURE 30-7: TYPICAL AND MAXIMUM IDD vs. TEMPERATURE (FRC MODE)**

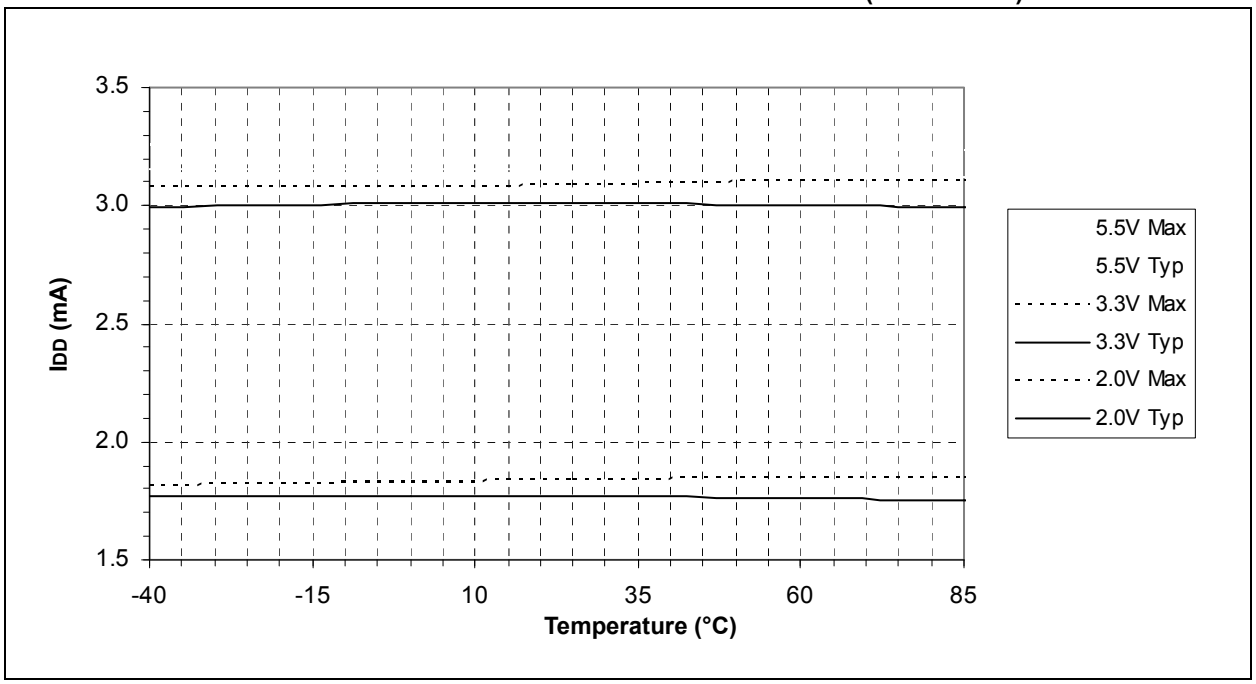



**FIGURE 30-8: TYPICAL AND MAXIMUM IIDLE vs. VDD (FRC MODE)**

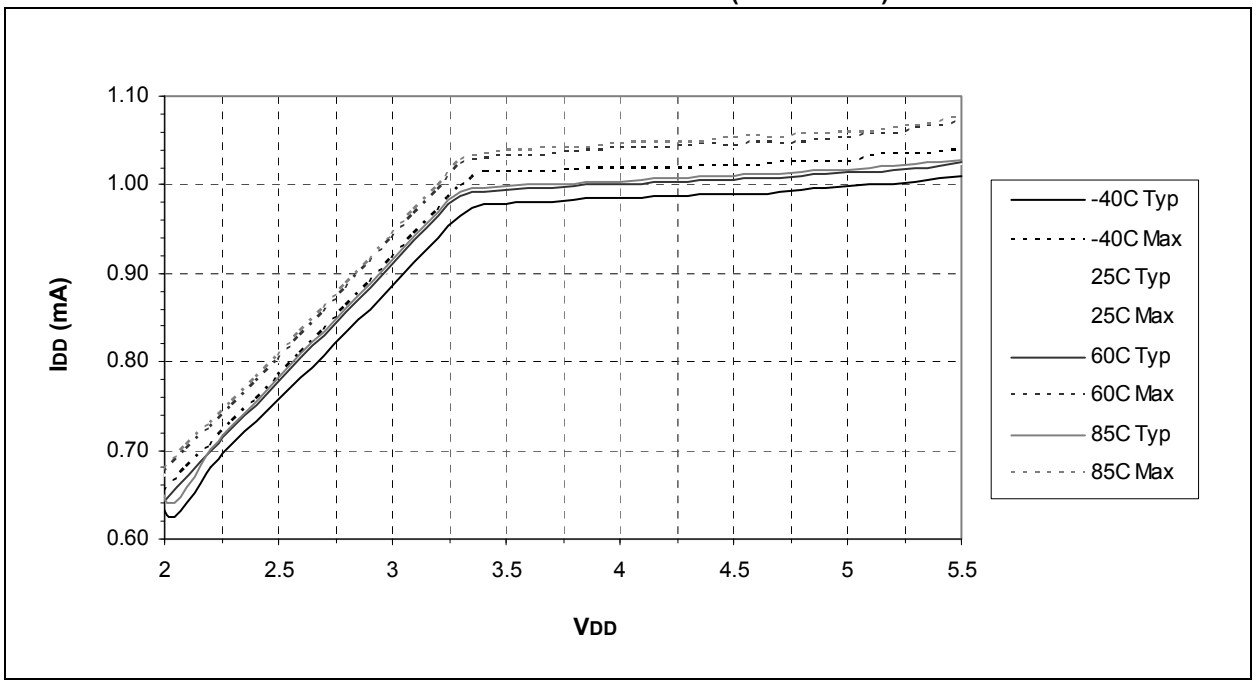


**FIGURE 30-32: TYPICAL BAND GAP VOLTAGE vs. VDD**

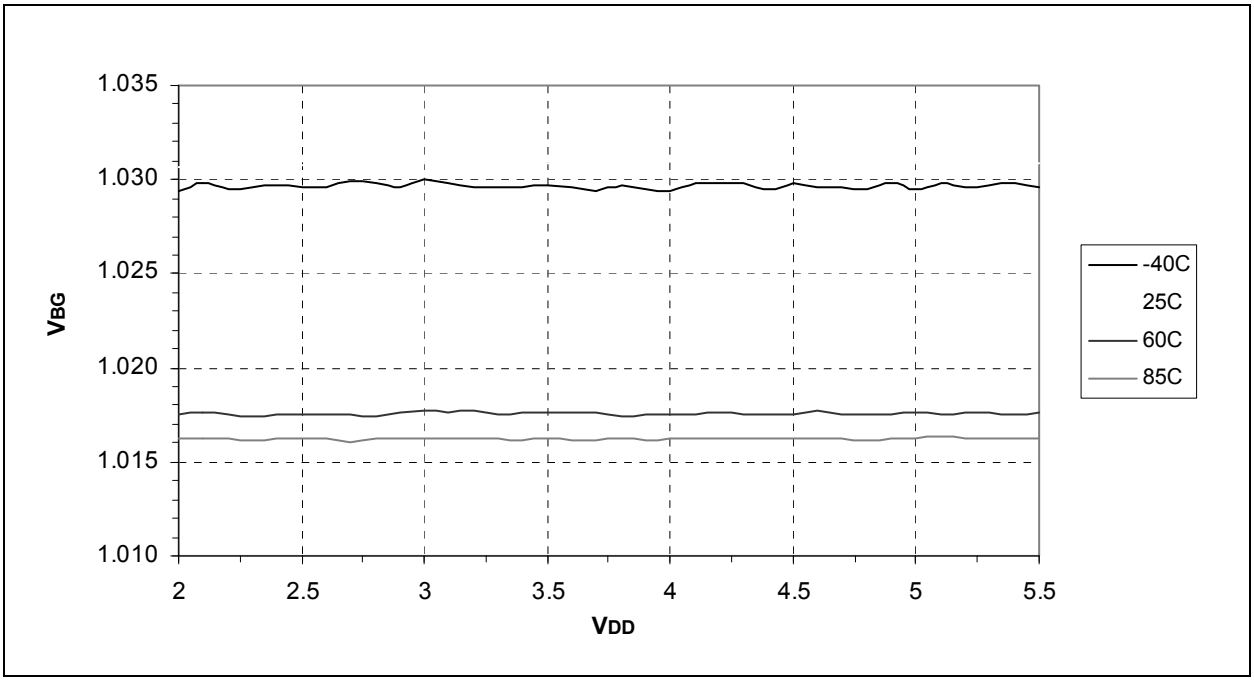
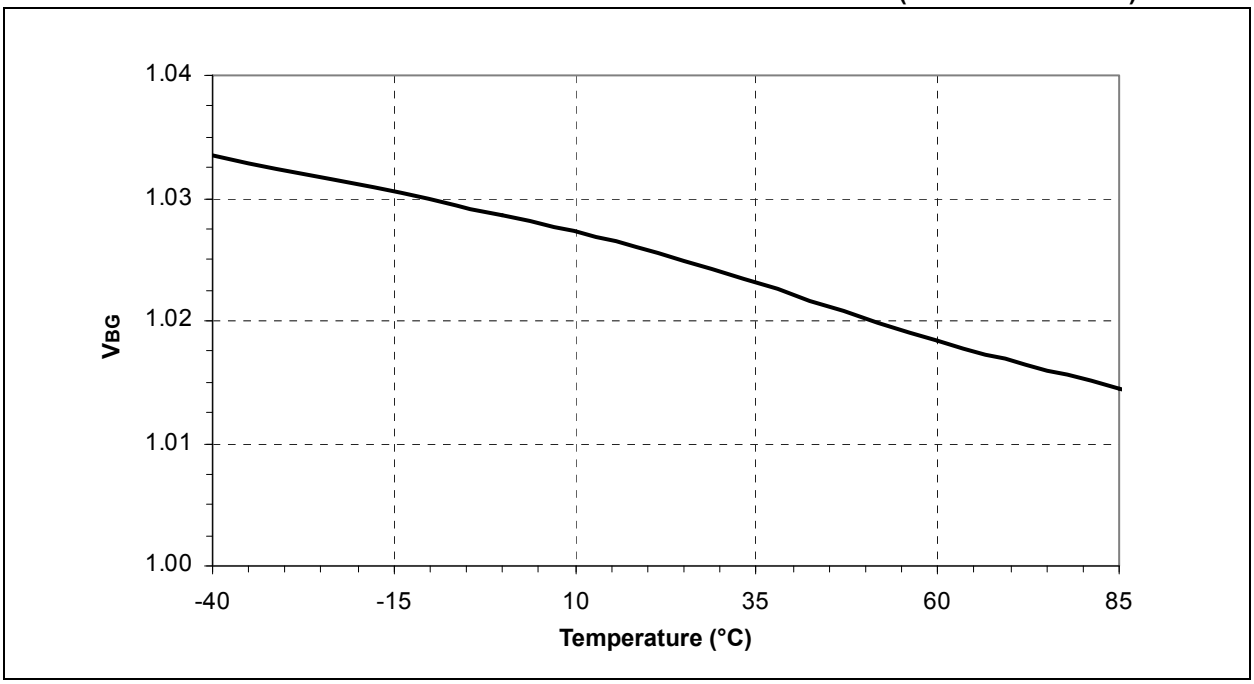
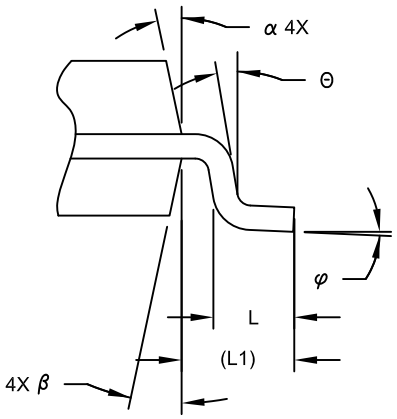



**FIGURE 30-33: TYPICAL BAND GAP VOLTAGE vs. TEMPERATURE (2.0V ≤ VDD ≤ 5.5V)**

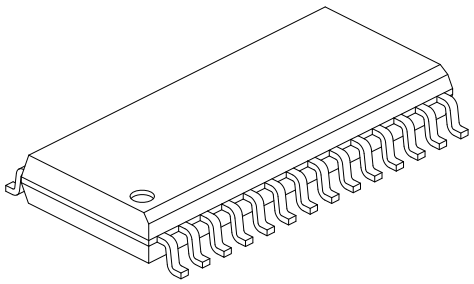


## 28-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

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




VIEW C



| Units | | MILLIMETERS | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Number of Pins | N | | 28 | |
| Pitch | e | | 1.27 BSC | |
| Overall Height | A | - | - | 2.65 |
| Molded Package Thickness | A2 | 2.05 | - | - |
| Standoff § | A1 | 0.10 | - | 0.30 |
| Overall Width | E | | 10.30 BSC | |
| Molded Package Width | E1 | | 7.50 BSC | |
| Overall Length | D | | 17.90 BSC | |
| Chamfer (Optional) | h | 0.25 | - | 0.75 |
| Foot Length | L | 0.40 | - | 1.27 |
| Footprint | L1 | | 1.40 REF | |
| Lead Angle | Θ | 0° | - | - |
| Foot Angle | φ | 0° | - | 8° |
| Lead Thickness | c | 0.18 | - | 0.33 |
| Lead Width | b | 0.31 | - | 0.51 |
| Mold Draft Angle Top | α | 5° | - | 15° |
| Mold Draft Angle Bottom | β | 5° | - | 15° |

**Notes:**

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic
3. Dimension D does not include mold flash, protrusions or gate burrs, which shall not exceed 0.15 mm per end. Dimension E1 does not include interlead flash or protrusion, which shall not exceed 0.25 mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M

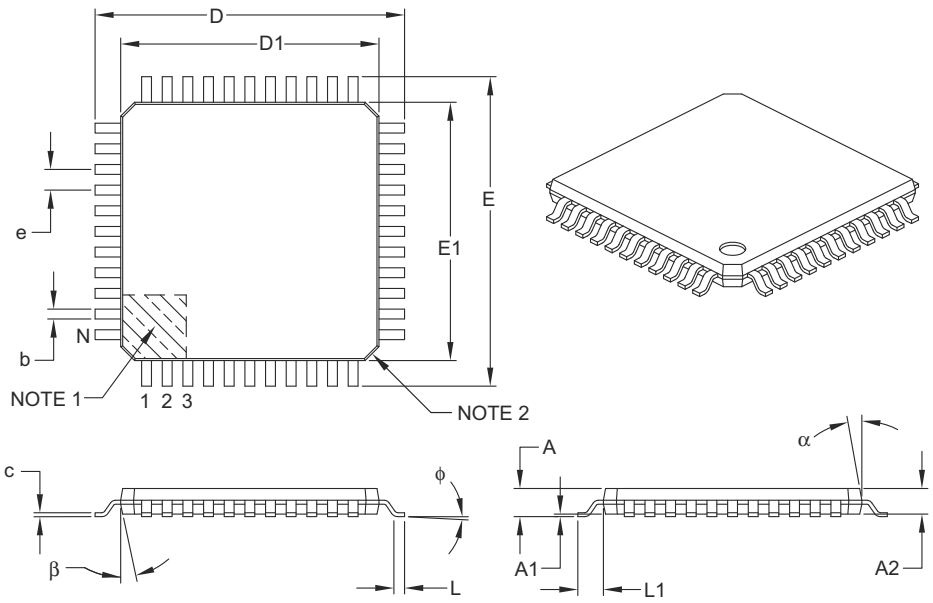
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

   REF: Reference Dimension, usually without tolerance, for information purposes only.
5. Datums A & B to be determined at Datum H.

Microchip Technology Drawing C04-052C Sheet 2 of 2

### 44-Lead Plastic Thin Quad Flatpack (PT) – 10x10x1 mm Body, 2.00 mm [TQFP]

**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 Leads | N | | 44 | |
| Lead Pitch | e | | 0.80 BSC | |
| Overall Height | A | – | – | 1.20 |
| Molded Package Thickness | A2 | 0.95 | 1.00 | 1.05 |
| Standoff | A1 | 0.05 | – | 0.15 |
| Foot Length | L | 0.45 | 0.60 | 0.75 |
| Footprint | L1 | | 1.00 REF | |
| Foot Angle | φ | 0° | 3.5° | 7° |
| Overall Width | E | | 12.00 BSC | |
| Overall Length | D | | 12.00 BSC | |
| Molded Package Width | E1 | | 10.00 BSC | |
| Molded Package Length | D1 | | 10.00 BSC | |
| Lead Thickness | c | 0.09 | – | 0.20 |
| Lead Width | b | 0.30 | 0.37 | 0.45 |
| Mold Draft Angle Top | α | 11° | 12° | 13° |
| Mold Draft Angle Bottom | β | 11° | 12° | 13° |

**Notes:**

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Chamfers at corners are optional; size may vary.
3. Dimensions D1 and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.25 mm per side.
4. 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 C04-076B