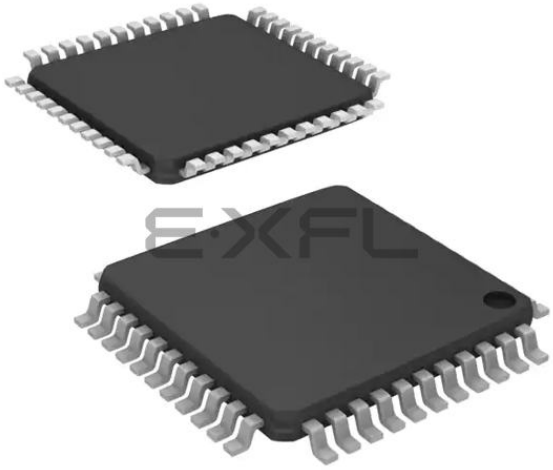


**Welcome to E-XFL.COM**

## 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 | 38 |
| 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 16x12b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 44-TQFP |
| Supplier Device Package | 44-TQFP (10x10) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic24fv32ka304-i-pt |

**REGISTER 6-1: NVMCON: NONVOLATILE MEMORY CONTROL REGISTER**

| R/S-0, HC | R/W-0 | R/W-0 | R/W-0 | U-0 | U-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| WR | WREN | WRERR | PGMONLY | — | — | — | — |
| bit 15 | | | | | | | bit 8 |

| U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | ERASE | NVMOP5 | NVMOP4 | NVMOP3 | NVMOP2 | NVMOP1 | NVMOP0 |
| bit 7 | | | | | | | bit 0 |

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

bit 15 **WR:** Write Control bit (program or erase)
1 = Initiates a data EEPROM erase or write cycle (can be set, but not cleared in software)
0 = Write cycle is complete (cleared automatically by hardware)

bit 14 **WREN:** Write Enable bit (erase or program)
1 = Enables an erase or program operation
0 = No operation allowed (device clears this bit on completion of the write/erase operation)

bit 13 **WRERR:** Write Flash Error Flag bit
1 = A write operation is prematurely terminated (any $\overline{\text{MCLR}}$ or WDT Reset during programming operation)
0 = The write operation completed successfully

bit 12 **PGMONLY:** Program Only Enable bit
1 = Write operation is executed without erasing target address(es) first
0 = Automatic erase-before-write
Write operations are preceded automatically by an erase of the target address(es).

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

bit 6 **ERASE:** Erase Operation Select bit
1 = Performs an erase operation when WR is set
0 = Performs a write operation when WR is set

bit 5-0 **NVMOP<5:0>:** Programming Operation Command Byte bits
Erase Operations (when ERASE bit is '1'):
011010 = Erases 8 words
011001 = Erases 4 words
011000 = Erases 1 word
0100xx = Erases entire data EEPROM
Programming Operations (when ERASE bit is '0'):
0010xx = Writes 1 word

### 7.5.2 DETECTING BOR

When BOR is enabled, the BOR bit (RCON<1>) is always reset to '1' on any BOR or POR event. This makes it difficult to determine if a BOR event has occurred just by reading the state of BOR alone. A more reliable method is to simultaneously check the state of both POR and BOR. This assumes that the POR and BOR bits are reset to '0' in the software immediately after any POR event. If the BOR bit is '1' while POR is '0', it can be reliably assumed that a BOR event has occurred.

| | |
|---|---|
| **Note:** | Even when the device exits from Deep Sleep mode, both the POR and BOR bits are set. |

### 7.5.3 DISABLING BOR IN SLEEP MODE

When BOREN<1:0> = 10, BOR remains under hardware control and operates as previously described. However, whenever the device enters Sleep mode, BOR is automatically disabled. When the device returns to any other operating mode, BOR is automatically re-enabled.

This mode allows for applications to recover from brown-out situations, while actively executing code when the device requires BOR protection the most. At the same time, it saves additional power in Sleep mode by eliminating the small incremental BOR current.

| | |
|---|---|
| **Note:** | BOR levels differ depending on device type; PIC24FV32KA3XX devices are at different levels than those of PIC24F32KA3XX devices. See **Section 29.0 "Electrical Characteristics"** for BOR voltage levels. |

**REGISTER 8-2: CORCON: CPU CONTROL 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 | U-0 | U-0 | R/C-0, HSC | R/W-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| — | — | — | — | IPL3(2) | PSV(1) | — | — |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | C = Clearable bit | 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 15-4 **Unimplemented:** Read as '0'

bit 3 **IPL3:** CPU Interrupt Priority Level Status bit(2)
1 = CPU Interrupt Priority Level is greater than 7
0 = CPU Interrupt Priority Level is 7 or less

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

**Note 1:** See Register 3-2 for the description of this bit, which is not dedicated to interrupt control functions.
**2:** The IPL3 bit is concatenated with the IPL<2:0> bits (SR<7:5>) to form the CPU Interrupt Priority Level.

**Note:** Bit 2 is described in **Section 3.0 "CPU"**.

**REGISTER 8-5: IFS0: INTERRUPT FLAG STATUS REGISTER 0 (CONTINUED)**

bit 2 **OC1IF:** Output Compare Channel 1 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 1 **IC1IF:** Input Capture Channel 1 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 0 **INT0IF:** External Interrupt 0 Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

**REGISTER 8-16: IEC5: INTERRUPT ENABLE CONTROL REGISTER 5**

| 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 | U-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | — | — | — | — | — | ULPWUIE |
| 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 **Unimplemented:** Read as '0'

bit 0 **ULPWUIE:** Ultra Low-Power Wake-up Interrupt Enable Bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

**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'

## 11.0 I/O PORTS

**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 I/O Ports, refer to the *"PIC24F Family Reference Manual"*, **Section 12. "I/O Ports with Peripheral Pin Select (PPS)"** (DS39711). Note that the PIC24FV32KA304 family devices do not support Peripheral Pin Select features.

All of the device pins (except VDD and VSS) are shared between the peripherals and the parallel I/O ports. All I/O input ports feature Schmitt Trigger inputs for improved noise immunity.

### 11.1 Parallel I/O (PIO) Ports

A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral. The peripheral's output buffer data and control signals are provided to a pair of multiplexers. The multiplexers select whether the peripheral or the associated port has ownership of the output data and control signals of the I/O pin. The logic also prevents "loop through", in which a port's digital output can drive the input of a peripheral that shares the same pin. Figure 11-1 illustrates how ports are shared with other peripherals and the associated I/O pin to which they are connected.

When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled. The I/O pin may be read, but the output driver for the parallel port bit will be disabled. If a peripheral is enabled, but the peripheral is not actively driving a pin, that pin may be driven by a port.

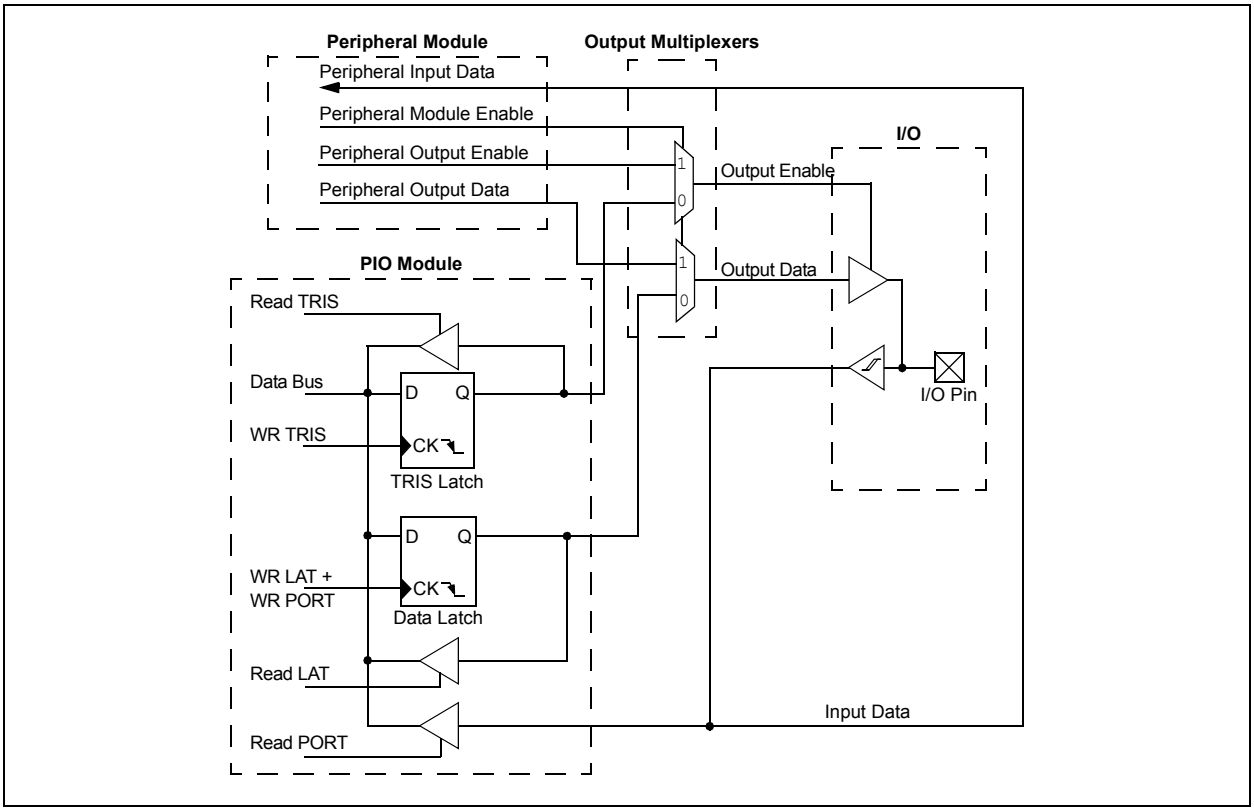
All port pins have three registers directly associated with their operation as digital I/O. The Data Direction register (TRISx) determines whether the pin is an input or an output. If the data direction bit is a '1', then the pin is an input. All port pins are defined as inputs after a Reset. Reads from the Data Latch register (LAT), read the latch. Writes to the latch, write the latch. Reads from the port (PORT), read the port pins; writes to the port pins, write the latch.

Any bit and its associated data and control registers that are not valid for a particular device will be disabled. That means the corresponding LATx and TRISx registers, and the port pin will read as zeros.

When a pin is shared with another peripheral or function that is defined as an input only, it is nevertheless regarded as a dedicated port because there is no other competing source of outputs.

**Note:** The I/O pins retain their state during Deep Sleep. They will retain this state at wake-up until the software restore bit (RELEASE) is cleared.

**FIGURE 11-1: BLOCK DIAGRAM OF A TYPICAL SHARED PORT STRUCTURE**



### 11.2.2 I/O PORT WRITE/READ TIMING

One instruction cycle is required between a port direction change or port write operation and a read operation of the same port. Typically, this instruction would be a NOP.

## 11.3 Input Change Notification (ICN)

The Input Change Notification function of the I/O ports allows the PIC24FV32KA304 family of devices to generate interrupt requests to the processor in response to a Change-of-State (COS) on selected input pins. This feature is capable of detecting input Change-of-States, even in Sleep mode, when the clocks are disabled. Depending on the device pin count, there are up to 23 external signals (CN0 through CN22) that may be selected (enabled) for generating an interrupt request on a Change-of-State.

There are six control registers associated with the ICN module. The CNEN1 and CNEN2 registers contain the interrupt enable control bits for each of the CN input pins. Setting any of these bits enables a CN interrupt for the corresponding pins.

Each CN pin also has a weak pull-up/pull-down connected to it. The pull-ups act as a current source that is connected to the pin. The pull-downs act as a current sink to eliminate the need for external resistors when push button or keypad devices are connected.

On any pin, only the pull-up resistor or the pull-down resistor should be enabled, but not both of them. If the push button or the keypad is connected to $V_{DD}$, enable the pull-down, or if they are connected to $V_{SS}$, enable the pull-up resistors. The pull-ups are enabled separately, using the CNPU1 and CNPU2 registers, which contain the control bits for each of the CN pins.

Setting any of the control bits enables the weak pull-ups for the corresponding pins. The pull-downs are enabled separately, using the CNPD1 and CNPD2 registers, which contain the control bits for each of the CN pins. Setting any of the control bits enables the weak pull-downs for the corresponding pins.

When the internal pull-up is selected, the pin uses $V_{DD}$ as the pull-up source voltage. When the internal pull-down is selected, the pins are pulled down to $V_{SS}$ by an internal resistor. Make sure that there is no external pull-up source/pull-down sink when the internal pull-ups/pull-downs are enabled.

| **Note:** | Pull-ups and pull-downs on Change Notification pins should always be disabled whenever the port pin is configured as a digital output. |
|---|---|

**EXAMPLE 11-1: PORT WRITE/READ EXAMPLE**

```
MOV   0xFF00, W0;          //Configure PORTB<15:8> as inputs and PORTB<7:0> as outputs
MOV   W0, TRISB;
NOP;                       //Delay 1 cycle
BTSS  PORTB, #13;          //Next Instruction

Equivalent 'C' Code
TRISB = 0xFF00;            //Configure PORTB<15:8> as inputs and PORTB<7:0> as outputs
NOP();                     //Delay 1 cycle
if(PORTBbits.RB13 == 1)    // execute following code if PORTB pin 13 is set.
{
}
```

## 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.

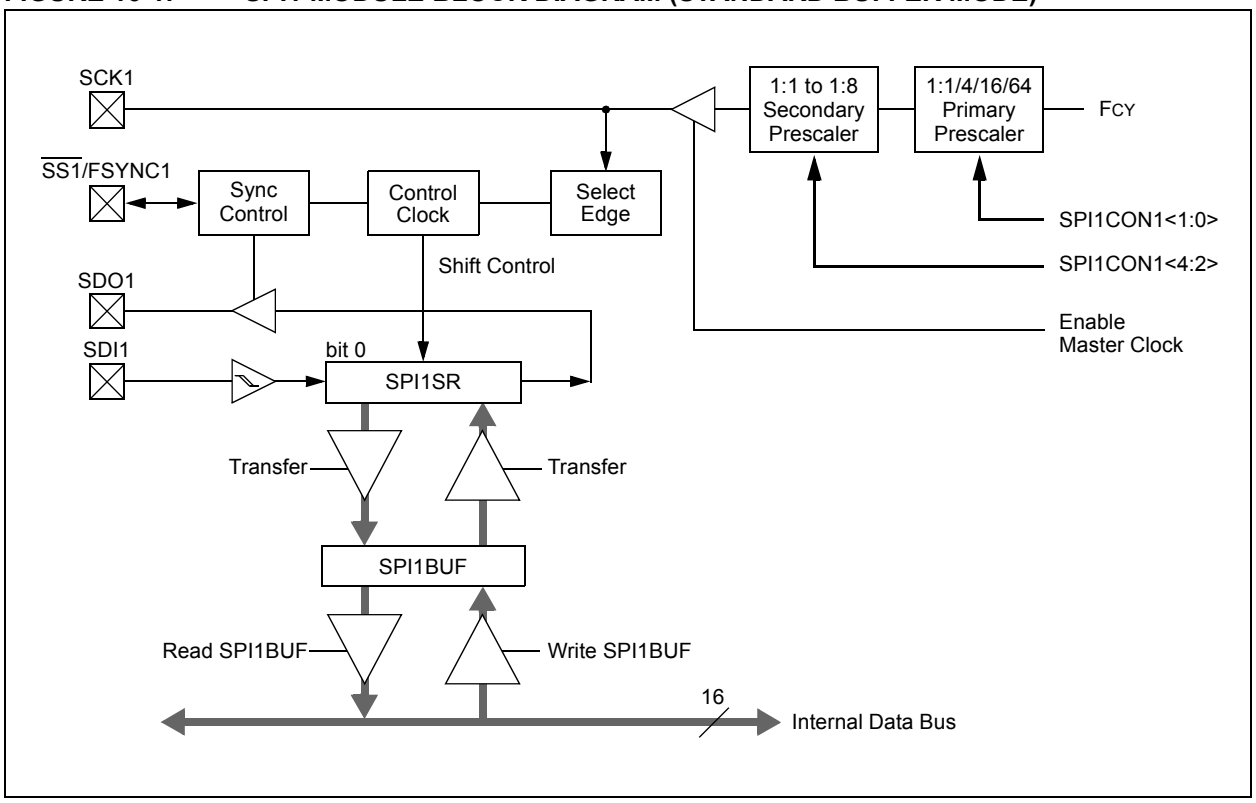
To set up the SPI1 module for the Standard Master mode of operation:

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>).

**FIGURE 16-1: SPI1 MODULE BLOCK DIAGRAM (STANDARD BUFFER MODE)**



**REGISTER 19-3: ALCFGRPT: ALARM CONFIGURATION REGISTER**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| ALRMEN | CHIME | AMASK3 | AMASK2 | AMASK1 | AMASK0 | ALRMPTR1 | ALRMPTR0 |
| 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 |
|---|---|---|---|---|---|---|---|
| ARPT7 | ARPT6 | ARPT5 | ARPT4 | ARPT3 | ARPT2 | ARPT1 | ARPT0 |
| 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 **ALRMEN:** Alarm Enable bit

1 = Alarm is enabled (cleared automatically after an alarm event whenever ARPT<7:0> = 00h and CHIME = 0)

0 = Alarm is disabled

bit 14 **CHIME:** Chime Enable bit

1 = Chime is enabled; ARPT<7:0> bits are allowed to roll over from 00h to FFh

0 = Chime is disabled; ARPT<7:0> bits stop once they reach 00h

bit 13-10 **AMASK<3:0>:** Alarm Mask Configuration bits

0000 = Every half second
0001 = Every second
0010 = Every 10 seconds
0011 = Every minute
0100 = Every 10 minutes
0101 = Every hour
0110 = Once a day
0111 = Once a week
1000 = Once a month
1001 = Once a year (except when configured for February 29th, once every 4 years)
101x = Reserved – do not use
11xx = Reserved – do not use

bit 9-8 **ALRMPTR<1:0>:** Alarm Value Register Window Pointer bits

Points to the corresponding Alarm Value registers when reading the ALRMVALH and ALRMVALL registers. The ALRMPTR<1:0> value decrements on every read or write of ALRMVALH until it reaches '00'.

ALRMVAL<15:8>:
00 = ALRMMIN
01 = ALRMWD
10 = ALRMMNTH
11 = Unimplemented

ALRMVAL<7:0>:
00 = ALRMSEC
01 = ALRMHR
10 = ALRMDAY
11 = Unimplemented

bit 7-0 **ARPT<7:0>:** Alarm Repeat Counter Value bits

11111111 = Alarm will repeat 255 more times
.
.
.
00000000 = Alarm will not repeat

The counter decrements on any alarm event; it is prevented from rolling over from 00h to FFh unless CHIME = 1.

**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'.

**REGISTER 25-1: CTMUCON1: CTMU CONTROL REGISTER 1**

| R/W-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| CTMUEN | — | CTMUSIDL | TGEN | EDGEN | EDGSEQEN | IDISSEN | CTTRIG |
| bit 15 | | | | | | | bit 8 |

| U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-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 | x = Bit is unknown |

bit 15 **CTMUEN:** CTMU Enable bit
1 = Module is enabled
0 = Module is disabled

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

bit 13 **CTMUSIDL:** CTMU Stop in Idle Mode bit
1 = Discontinues module operation when device enters Idle mode
0 = Continues module operation in Idle mode

bit 12 **TGEN:** Time Generation Enable bit
1 = Enables edge delay generation
0 = Disables edge delay generation

bit 11 **EDGEN:** Edge Enable bit
1 = Edges are not blocked
0 = Edges are blocked

bit 10 **EDGSEQEN:** Edge Sequence Enable bit
1 = Edge 1 event must occur before Edge 2 event can occur
0 = No edge sequence is needed

bit 9 **IDISSEN:** Analog Current Source Control bit
1 = Analog current source output is grounded
0 = Analog current source output is not grounded

bit 8 **CTTRIG: CTMU** Trigger Control bit
1 = Trigger output is enabled
0 = Trigger output is disabled

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

**REGISTER 26-10: DEVREV: DEVICE REVISION REGISTER**

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

| 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 | R | R | R | R |
|---|---|---|---|---|---|---|---|
| — | — | — | — | REV3 | REV2 | REV1 | REV0 |
| 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 23-4 **Unimplemented:** Read as ‘0’

bit 3-0 **REV<3:0>:** Minor Revision Identifier bits

If the WDT is enabled in hardware (FWDTEN<1:0> = 11), it will continue to run during Sleep or Idle modes. When the WDT time-out occurs, the device will wake and code execution will continue from where the PWRSAV instruction was executed. The corresponding SLEEP or IDLE bit (RCON<3:2>) will need to be cleared in software after the device wakes up.

The WDT Flag bit, WDTO (RCON<4>), is not automatically cleared following a WDT time-out. To detect subsequent WDT events, the flag must be cleared in software.

**Note:** The CLRWDT and PWRSAV instructions clear the prescaler and postscaler counts when executed.

### 26.3.1 WINDOWED OPERATION

The Watchdog Timer has an optional Fixed Window mode of operation. In this Windowed mode, CLRWDT instructions can only reset the WDT during the last 1/4 of the programmed WDT period. A CLRWDT instruction executed before that window causes a WDT Reset, similar to a WDT time-out.

Windowed WDT mode is enabled by programming the Configuration bit, WINDIS (FWDT<6>), to '0'.

### 26.3.2 CONTROL REGISTER

The WDT is enabled or disabled by the FWDTEN<1:0> Configuration bits. When both the FWDTEN<1:0> Configuration bits are set, the WDT is always enabled.

The WDT can be optionally controlled in software when the FWDTEN<1:0> Configuration bits have been programmed to '10'. The WDT is enabled in software by setting the SWDTEN control bit (RCON<5>). The SWDTEN control bit is cleared on any device Reset. The software WDT option allows the user to enable the WDT for critical code segments, and disable the WDT during non-critical segments, for maximum power savings. When the FWTEN<1:0> bits are set to '01', the WDT is only enabled in Run and Idle modes, and is disabled in Sleep. Software control of the SWDTEN bit (RCON<5>) is disabled with this setting.

**FIGURE 26-2: WDT BLOCK DIAGRAM**

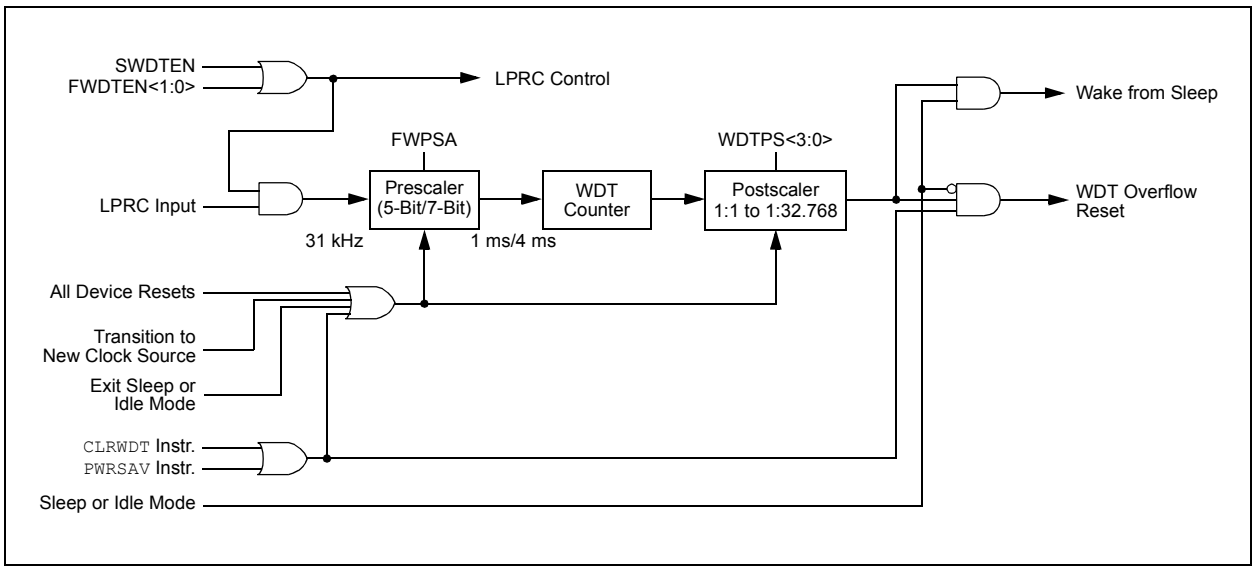


**TABLE 29-24: COMPARATOR TIMINGS**

| Param No. | Symbol | Characteristic | Min | Typ | Max | Units | Comments |
|---|---|---|---|---|---|---|---|
| 300 | TRESP | Response Time*(1) | — | 150 | 400 | ns | |
| 301 | TMC2OV | Comparator Mode Change to Output Valid* | — | — | 10 | μs | |

* Parameters are characterized but not tested.

**Note 1:** Response time is measured with one comparator input at (VDD – 1.5)/2, while the other input transitions from VSS to VDD.

**TABLE 29-25: COMPARATOR VOLTAGE REFERENCE SETTLING TIME SPECIFICATIONS**

| Param No. | Symbol | Characteristic | Min | Typ | Max | Units | Comments |
|---|---|---|---|---|---|---|---|
| VR310 | TSET | Settling Time(1) | — | — | 10 | μs | |

**Note 1:** Settling time is measured while CVRSS = 1 and the CVR<3:0> bits transition from '0000' to '1111'.

**FIGURE 29-6: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING CHARACTERISTICS**

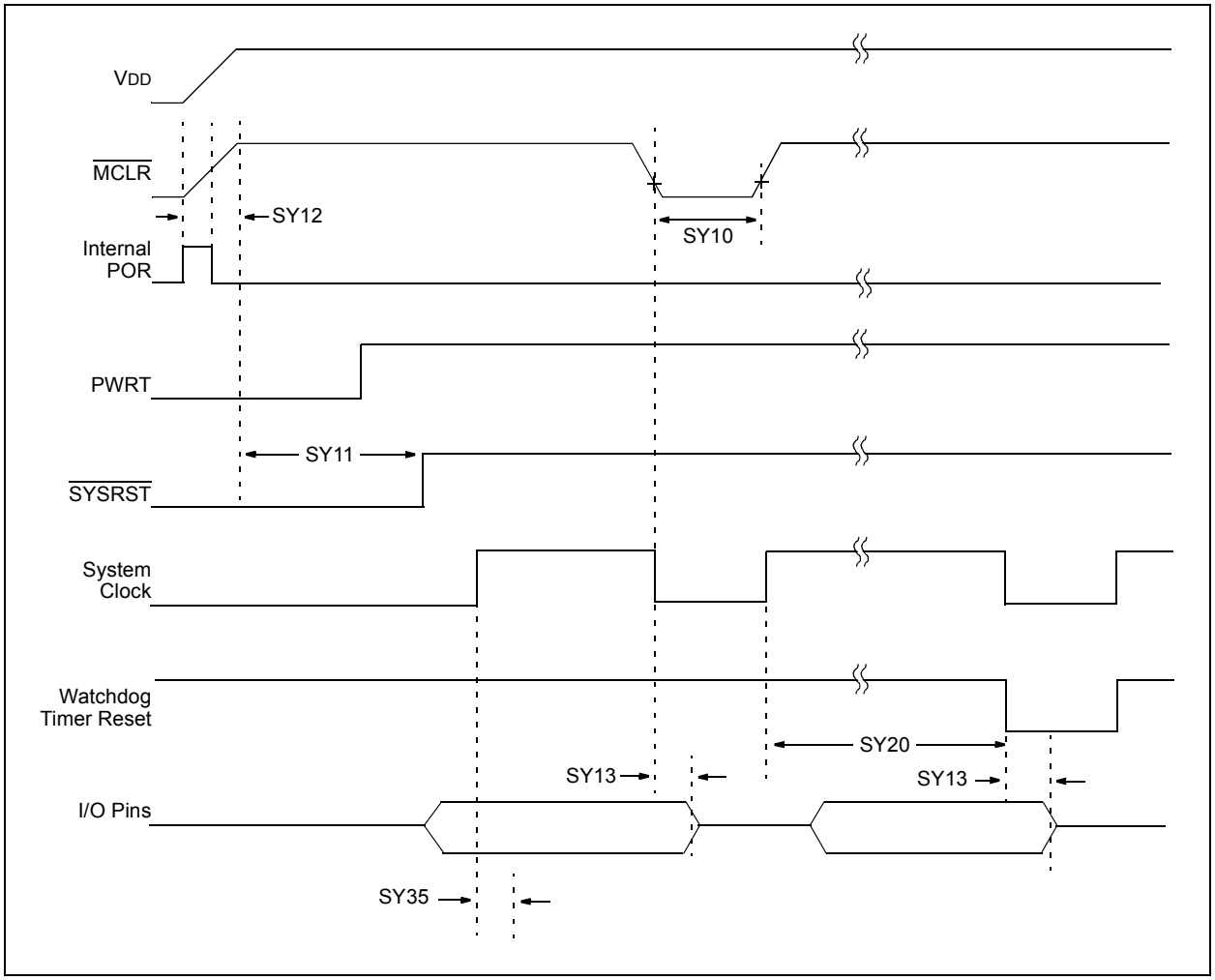


**FIGURE 30-51: VIL/VIH vs. VDD (GENERAL PURPOSE I/O, TEMPERATURES AS NOTED)**

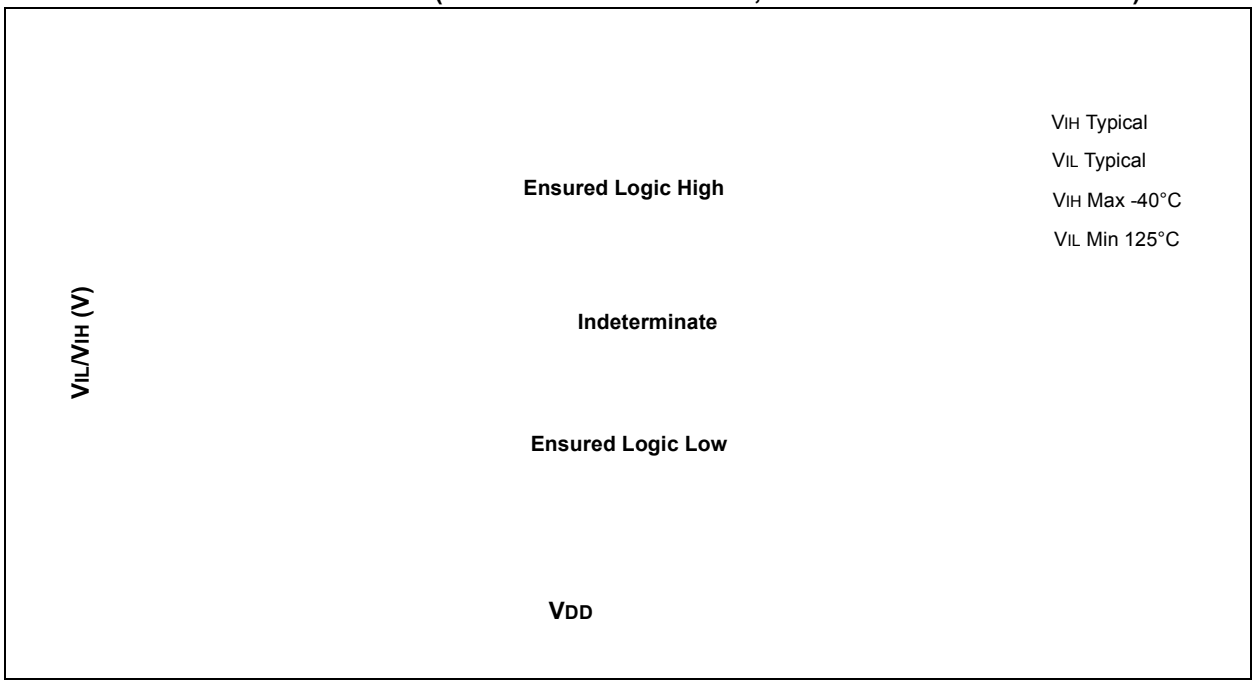
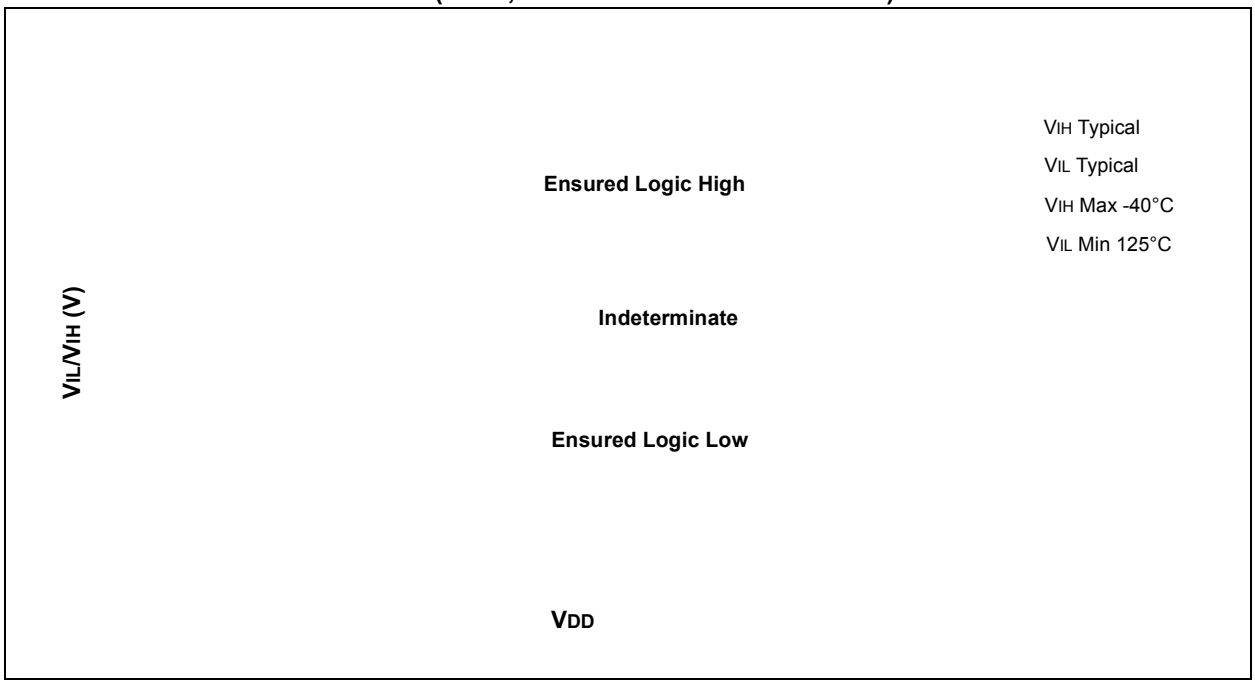
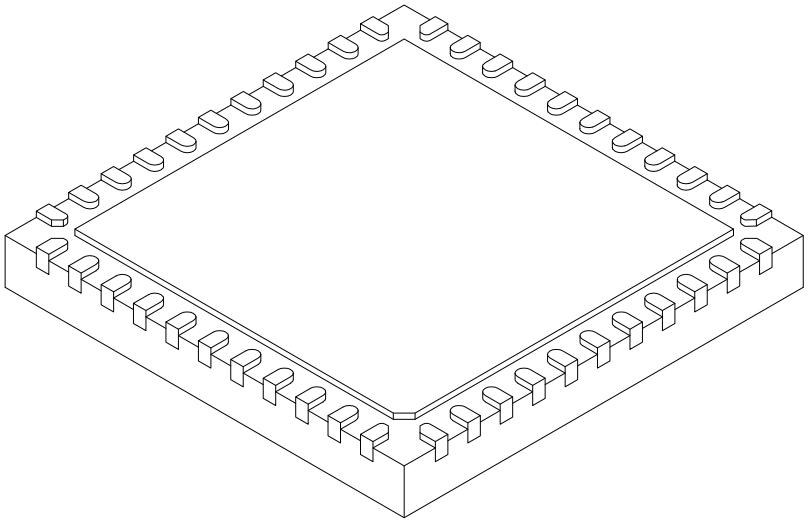



**FIGURE 30-52: VIL/VIH vs. VDD (I2C™, TEMPERATURES AS NOTED)**



## 44-Lead Plastic Quad Flat, No Lead Package (ML) - 8x8 mm Body [QFN]

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



| Units | | MILLIMETERS | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Number of Pins | N | | 44 | |
| 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 | | 8.00 BSC | |
| Exposed Pad Width | E2 | 6.25 | 6.45 | 6.60 |
| Overall Length | D | | 8.00 BSC | |
| Exposed Pad Length | D2 | 6.25 | 6.45 | 6.60 |
| Terminal Width | b | 0.20 | 0.30 | 0.35 |
| Terminal Length | L | 0.30 | 0.40 | 0.50 |
| 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

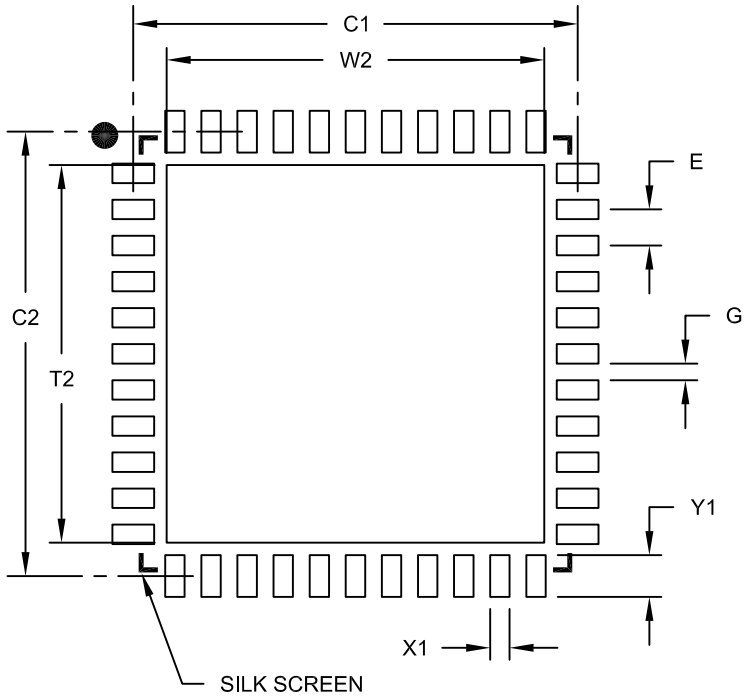
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

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

Microchip Technology Drawing C04-103C Sheet 2 of 2

44-Lead Plastic Quad Flat, No Lead Package (ML) - 8x8 mm Body [QFN]

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




RECOMMENDED LAND PATTERN

| Units | | MILLIMETERS | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Contact Pitch | E | | 0.65 BSC | |
| Optional Center Pad Width | W2 | | | 6.60 |
| Optional Center Pad Length | T2 | | | 6.60 |
| Contact Pad Spacing | C1 | | 8.00 | |
| Contact Pad Spacing | C2 | | 8.00 | |
| Contact Pad Width (X44) | X1 | | | 0.35 |
| Contact Pad Length (X44) | Y1 | | | 0.85 |
| Distance Between Pads | G | 0.25 | | |

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2103B

## S



## T