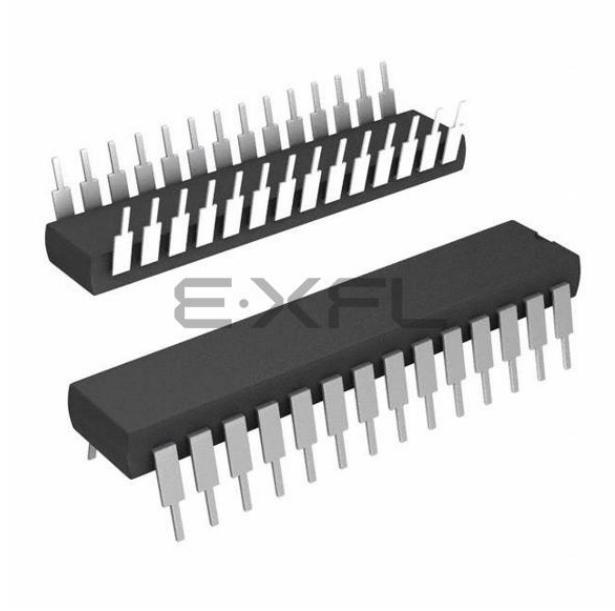


**Welcome to E-XFL.COM**

## What is "Embedded - Microcontrollers"?

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

## Applications of "Embedded - Microcontrollers"

| Details | |
|---|---|
| Product Status | Active |
| Core Processor | PIC |
| Core Size | 8-Bit |
| Speed | 32MHz |
| Connectivity | I²C, LINbus, SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, POR, PSMC, PWM, WDT |
| Number of I/O | 24 |
| Program Memory Size | 28KB (16K x 14) |
| Program Memory Type | FLASH |
| EEPROM Size | 256 x 8 |
| RAM Size | 2K x 8 |
| Voltage - Supply (Vcc/Vdd) | 2.3V ~ 5.5V |
| Data Converters | A/D 11x12b; D/A 1x8b, 3x5b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Through Hole |
| Package / Case | 28-DIP (0.300", 7.62mm) |
| Supplier Device Package | 28-SPDIP |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic16f1788-i-sp |

**TABLE 1-3: PIC16(L)F1789 PINOUT DESCRIPTION (CONTINUED)**

| Name | Function | Input Type | Output Type | Description |
|---|---|---|---|---|
| RE1/AN6/PSMC3B | RE1 | TTL/ST | CMOS | General purpose I/O. |
| | AN6 | AN | — | ADC Channel 6 input. |
| | PSMC3B | — | CMOS | PSMC3 output B. |
| RE2/AN7/PSMC3A | RE2 | TTL/ST | CMOS | General purpose I/O. |
| | AN7 | AN | — | ADC Channel 7 input. |
| | PSMC3A | — | CMOS | PSMC3 output A. |
| RE3/$\overline{MCLR}$/$V_{PP}$ | RE3 | TTL/ST | — | General purpose input. |
| | $\overline{MCLR}$ | ST | — | Master Clear with internal pull-up. |
| | $V_{PP}$ | HV | — | Programming voltage. |
| $V_{DD}$ | $V_{DD}$ | Power | — | Positive supply. |
| $V_{SS}$ | $V_{SS}$ | Power | — | Ground reference. |

**Legend:** AN = Analog input or output CMOS = CMOS compatible input or output OD = Open-Drain
TTL = TTL compatible input ST = Schmitt Trigger input with CMOS levels $I^2C$ = Schmitt Trigger input with $I^2C$ levels
HV = High Voltage XTAL = Crystal

**Note 1:** Pin functions can be assigned to one of two locations via software. See **Register 13-1**.
**2:** All pins have interrupt-on-change functionality.

### 3.3.5 CORE FUNCTION REGISTERS SUMMARY

The Core Function registers listed in Table 3-11 can be addressed from any Bank.

**TABLE 3-11: CORE FUNCTION REGISTERS SUMMARY**

| Addr | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOR | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bank 0-31** | | | | | | | | | | | |
| x00h or x80h | INDF0 | Addressing this location uses contents of FSR0H/FSR0L to address data memory (not a physical register) | | | | | | | | xxxx xxxx | uuuu uuuu |
| x01h or x81h | INDF1 | Addressing this location uses contents of FSR1H/FSR1L to address data memory (not a physical register) | | | | | | | | xxxx xxxx | uuuu uuuu |
| x02h or x82h | PCL | Program Counter (PC) Least Significant Byte | | | | | | | | 0000 0000 | 0000 0000 |
| x03h or x83h | STATUS | — | — | — | $\overline{TO}$ | $\overline{PD}$ | Z | DC | C | ---1 1000 | ---q quuu |
| x04h or x84h | FSR0L | Indirect Data Memory Address 0 Low Pointer | | | | | | | | 0000 0000 | uuuu uuuu |
| x05h or x85h | FSR0H | Indirect Data Memory Address 0 High Pointer | | | | | | | | 0000 0000 | 0000 0000 |
| x06h or x86h | FSR1L | Indirect Data Memory Address 1 Low Pointer | | | | | | | | 0000 0000 | uuuu uuuu |
| x07h or x87h | FSR1H | Indirect Data Memory Address 1 High Pointer | | | | | | | | 0000 0000 | 0000 0000 |
| x08h or x88h | BSR | — | — | — | BSR4 | BSR3 | BSR2 | BSR1 | BSR0 | ---0 0000 | ---0 0000 |
| x09h or x89h | WREG | Working Register | | | | | | | | 0000 0000 | uuuu uuuu |
| x0Ah or x8Ah | PCLATH | — | Write Buffer for the upper 7 bits of the Program Counter | | | | | | | -000 0000 | -000 0000 |
| x0Bh or x8Bh | INTCON | GIE | PEIE | TMR0IE | INTE | IOCIE | TMR0IF | INTF | IOCIF | 0000 0000 | 0000 0000 |

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

## 9.0 POWER-DOWN MODE (SLEEP)

The Power-down mode is entered by executing a SLEEP instruction.

Upon entering Sleep mode, the following conditions exist:

1. WDT will be cleared but keeps running, if enabled for operation during Sleep.
2. $\overline{PD}$ bit of the STATUS register is cleared.
3. $\overline{TO}$ bit of the STATUS register is set.
4. CPU clock is disabled.
5. 31 kHz LFINTOSC is unaffected and peripherals that operate from it may continue operation in Sleep.
6. Timer1 and peripherals that operate from Timer1 continue operation in Sleep when the Timer1 clock source selected is:
   - LFINTOSC
   - T1CKI
   - Timer1 oscillator
7. ADC is unaffected, if the dedicated FRC oscillator is selected.
8. I/O ports maintain the status they had before SLEEP was executed (driving high, low or high-impedance).
9. Resets other than WDT are not affected by Sleep mode.

Refer to individual chapters for more details on peripheral operation during Sleep.

To minimize current consumption, the following conditions should be considered:

- I/O pins should not be floating
- External circuitry sinking current from I/O pins
- Internal circuitry sourcing current from I/O pins
- Current draw from pins with internal weak pull-ups
- Modules using 31 kHz LFINTOSC
- Modules using Timer1 oscillator

I/O pins that are high-impedance inputs should be pulled to VDD or VSS externally to avoid switching currents caused by floating inputs.

Examples of internal circuitry that might be sourcing current include modules such as the DAC and FVR modules. See **Section 19.0 “8-Bit Digital-to-Analog Converter (DAC) Module”** and **Section 15.0 “Fixed Voltage Reference (FVR)”** for more information on these modules.

### 9.1 Wake-up from Sleep

The device can wake-up from Sleep through one of the following events:

1. External Reset input on $\overline{MCLR}$ pin, if enabled
2. BOR Reset, if enabled
3. POR Reset
4. Watchdog Timer, if enabled
5. Any external interrupt
6. Interrupts by peripherals capable of running during Sleep (see individual peripheral for more information)

The first three events will cause a device Reset. The last three events are considered a continuation of program execution. To determine whether a device Reset or wake-up event occurred, refer to **Section 5.12 “Determining the Cause of a Reset”**.

When the SLEEP instruction is being executed, the next instruction (PC + 1) is prefetched. For the device to wake-up through an interrupt event, the corresponding interrupt enable bit must be enabled. Wake-up will occur regardless of the state of the GIE bit. If the GIE bit is disabled, the device continues execution at the instruction after the SLEEP instruction. If the GIE bit is enabled, the device executes the instruction after the SLEEP instruction, the device will then call the Interrupt Service Routine. In cases where the execution of the instruction following SLEEP is not desirable, the user should have a NOP after the SLEEP instruction.

The WDT is cleared when the device wakes up from Sleep, regardless of the source of wake-up.

## 10.0 LOW DROPOUT (LDO) VOLTAGE REGULATOR

The “F” devices have an internal Low Dropout Regulator (LDO) which provide operation above 3.6V. The LDO regulates a voltage for the internal device logic while permitting the VDD and I/O pins to operate at a higher voltage. There is no user enable/disable control available for the LDO, it is always active. The “LF” devices operate at a maximum VDD of 3.6V and does not incorporate an LDO.

A device I/O pin may be configured as the LDO voltage output, identified as the VCAP pin. Although not required, an external low-ESR capacitor may be connected to the VCAP pin for additional regulator stability.

The $\overline{\text{VCAPEN}}$ bit of Configuration Words determines if which pin is assigned as the VCAP pin. Refer to Table 10-1.

**TABLE 10-1: $\overline{\text{VCAPEN}}$ SELECT BIT**

| $\overline{\text{VCAPEN}}$ | Pin |
|---|---|
| 1 | No VCAP |
| 0 | RA6 |

On power-up, the external capacitor will load the LDO voltage regulator. To prevent erroneous operation, the device is held in Reset while a constant current source charges the external capacitor. After the cap is fully charged, the device is released from Reset. For more information on the constant current rate, refer to the LDO Regulator Characteristics Table in **Section 31.0 “Electrical Specifications”**.

**TABLE 10-2: SUMMARY OF CONFIGURATION WORD WITH LDO**

| Name | Bits | Bit -/7 | Bit -/6 | Bit 13/5 | Bit 12/4 | Bit 11/3 | Bit 10/2 | Bit 9/1 | Bit 8/0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|---|
| CONFIG2 | 13:8 | — | — | LVP | $\overline{\text{DEBUG}}$ | $\overline{\text{LPBOR}}$ | BORV | STVREN | PLLEN | 60 |
| | 7:0 | — | — | $\overline{\text{VCAPEN}}$(1) | — | — | — | WRT<1:0> | | |

**Legend:** — = unimplemented locations read as ‘0’. Shaded cells are not used by LDO.

**Note 1:** “F” devices only.

**EXAMPLE 12-3: FLASH PROGRAM MEMORY READ**

```
* This code block will read 1 word of program
* memory at the memory address:
    PROG_ADDR_HI : PROG_ADDR_LO
*   data will be returned in the variables;
*   PROG_DATA_HI, PROG_DATA_LO

    BANKSEL  EEADRL           ; Select Bank for EEPROM registers
    MOVLW    PROG_ADDR_LO     ;
    MOVWF    EEADRL           ; Store LSB of address
    MOVLW    PROG_ADDR_HI     ;
    MOVWL    EEADRH           ; Store MSB of address

    BCF      EECON1,CFGS      ; Do not select Configuration Space
    BSF      EECON1,EEPGD     ; Select Program Memory
    BCF      INTCON,GIE       ; Disable interrupts
    BSF      EECON1,RD        ; Initiate read
    NOP                       ; Executed (Figure 12-1)
    NOP                       ; Ignored (Figure 12-1)
    BSF      INTCON,GIE       ; Restore interrupts

    MOVF     EEDATL,W         ; Get LSB of word
    MOVWF    PROG_DATA_LO     ; Store in user location
    MOVF     EEDATH,W         ; Get MSB of word
    MOVWF    PROG_DATA_HI     ; Store in user location
```

## 19.6 Register Definitions: DAC Control

### REGISTER 19-1: DAC1CON0: VOLTAGE REFERENCE CONTROL REGISTER 0

| R/W-0/0 | U-0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | U-0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| DAC1EN | — | DAC1OE1 | DAC1OE2 | DAC1PSS<1:0> | | — | DAC1NSS |
| bit 7 | | | | | | | bit 0 |

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

bit 7 **DAC1EN:** DAC1 Enable bit
1 = DAC1 is enabled
0 = DAC1 is disabled

bit 6 **Unimplemented:** Read as ‘0’

bit 5 **DAC1OE1:** DAC1 Voltage Output 1 Enable bit
1 = DAC1 voltage level is also an output on the DAC1OUT1 pin
0 = DAC1 voltage level is disconnected from the DAC1OUT1 pin

bit 4 **DAC1OE2:** DAC1 Voltage Output 2 Enable bit
1 = DAC1 voltage level is also an output on the DAC1OUT2 pin
0 = DAC1 voltage level is disconnected from the DAC1OUT2 pin

bit 3-2 **DAC1PSS<1:0>:** DAC1 Positive Source Select bits
11 = Reserved, do not use
10 = FVR Buffer2 output
01 = $V_{REF+}$ pin
00 = $V_{DD}$

bit 1 **Unimplemented:** Read as ‘0’

bit 0 **DAC1NSS:** DAC1 Negative Source Select bits
1 = $V_{REF-}$ pin
0 = $V_{SS}$

### REGISTER 19-2: DAC1CON1: VOLTAGE REFERENCE CONTROL REGISTER 1

| R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| DAC1R<7:0> | | | | | | | |
| bit 7 | | | | | | | bit 0 |

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

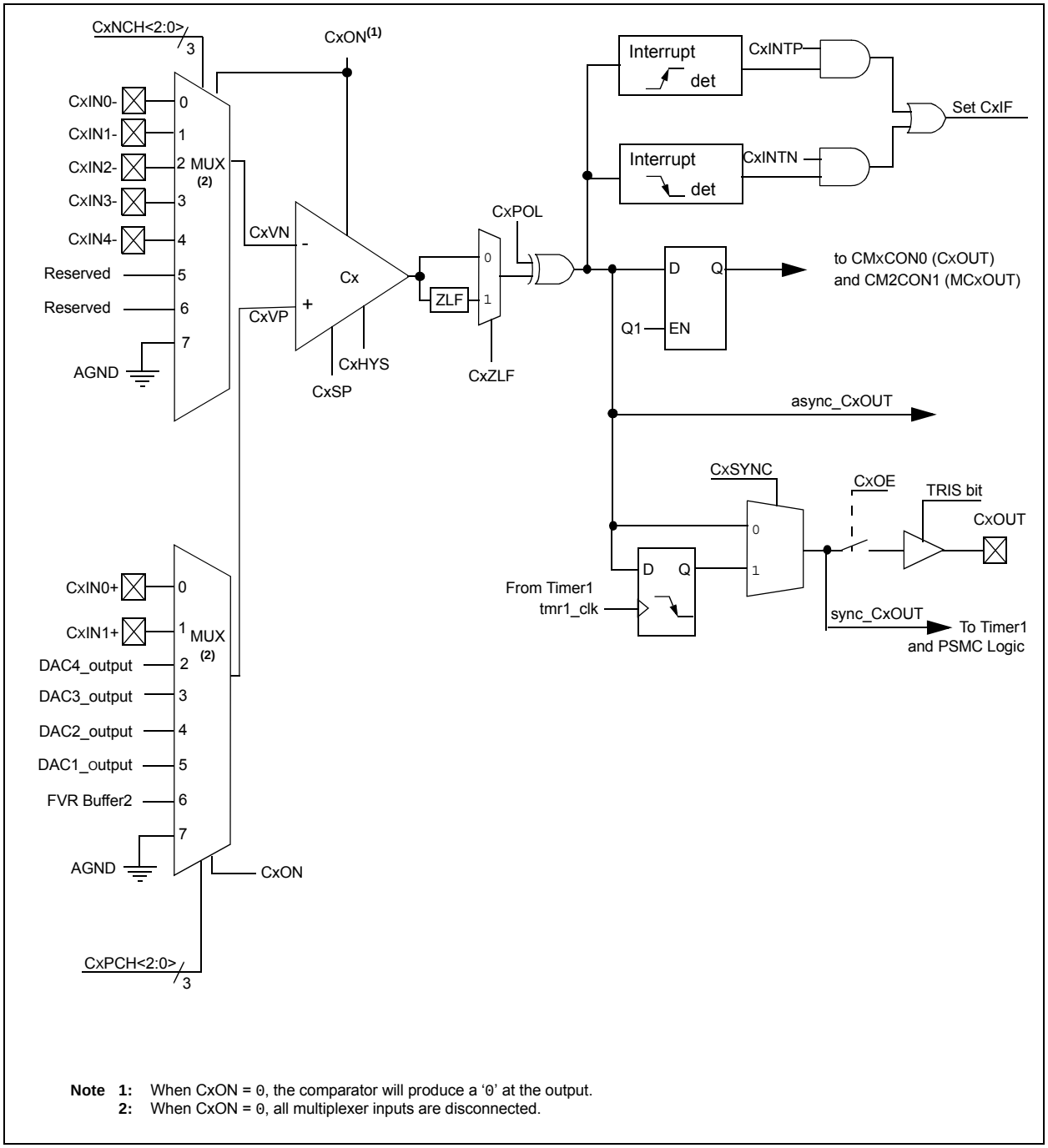
bit 7-0 **DAC1R<7:0>:** DAC1 Voltage Output Select bits

### TABLE 19-1: SUMMARY OF REGISTERS ASSOCIATED WITH THE DAC MODULE

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on page |
|---|---|---|---|---|---|---|---|---|---|
| FVRCON | FVREN | FVRRDY | TSEN | TSRNG | CDAFVR<1:0> | | ADFVR<1:0> | | **167** |
| DAC1CON0 | DAC1EN | — | DAC1OE1 | DAC1OE2 | DAC1PSS<1:0> | | — | DAC1NSS | **192** |
| DAC1CON1 | DAC1R<7:0> | | | | | | | | **192** |

**Legend:** — = Unimplemented location, read as ‘0’. Shaded cells are not used with the DAC module.

**FIGURE 21-2: COMPARATOR MODULE SIMPLIFIED BLOCK DIAGRAM**




**Note 1:** When CxON = 0, the comparator will produce a '0' at the output.
**2:** When CxON = 0, all multiplexer inputs are disconnected.

## 21.3 Comparator Hysteresis

A selectable amount of separation voltage can be added to the input pins of each comparator to provide a hysteresis function to the overall operation. Hysteresis is enabled by setting the CxHYS bit of the CMxCON0 register.

See **Section 30.0 "Electrical Specifications"** for more information.

## 21.4 Timer1 Gate Operation

The output resulting from a comparator operation can be used as a source for gate control of Timer1. See **Section 23.6 "Timer1 Gate"** for more information. This feature is useful for timing the duration or interval of an analog event.

It is recommended that the comparator output be synchronized to Timer1. This ensures that Timer1 does not increment while a change in the comparator is occurring.

### 21.4.1 COMPARATOR OUTPUT SYNCHRONIZATION

The output from a comparator can be synchronized with Timer1 by setting the CxSYNC bit of the CMxCON0 register.

Once enabled, the comparator output is latched on the falling edge of the Timer1 source clock. If a prescaler is used with Timer1, the comparator output is latched after the prescaling function. To prevent a race condition, the comparator output is latched on the falling edge of the Timer1 clock source and Timer1 increments on the rising edge of its clock source. See the Comparator Block Diagram (Figure 21-2) and the Timer1 Block Diagram (Figure 23-1) for more information.

## 21.5 Comparator Interrupt

An interrupt can be generated upon a change in the output value of the comparator for each comparator, a rising edge detector and a falling edge detector are present.

When either edge detector is triggered and its associated enable bit is set (CxINTP and/or CxINTN bits of the CMxCON1 register), the Corresponding Interrupt Flag bit (CxIF bit of the PIR2 register) will be set.

To enable the interrupt, you must set the following bits:

- CxON, CxPOL and CxSP bits of the CMxCON0 register
- CxIE bit of the PIE2 register
- CxINTP bit of the CMxCON1 register (for a rising edge detection)
- CxINTN bit of the CMxCON1 register (for a falling edge detection)
- PEIE and GIE bits of the INTCON register

The associated interrupt flag bit, CxIF bit of the PIR2 register, must be cleared in software. If another edge is detected while this flag is being cleared, the flag will still be set at the end of the sequence.

| | |
|---|---|
| **Note:** | Although a comparator is disabled, an interrupt can be generated by changing the output polarity with the CxPOL bit of the CMxCON0 register, or by switching the comparator on or off with the CxON bit of the CMxCON0 register. |

## 21.6 Comparator Positive Input Selection

Configuring the CxPCH<2:0> bits of the CMxCON1 register directs an internal voltage reference or an analog pin to the non-inverting input of the comparator:

- CxIN+ analog pin
- DAC output
- FVR (Fixed Voltage Reference)
- VSS (Ground)

See **Section 15.0 "Fixed Voltage Reference (FVR)"** for more information on the Fixed Voltage Reference module.

See **Section 19.0 "8-Bit Digital-to-Analog Converter (DAC) Module"** for more information on the DAC input signal.

Any time the comparator is disabled (CxON = 0), all comparator inputs are disabled.

**REGISTER 21-3: CMOUT: COMPARATOR OUTPUT REGISTER**

| U-0 | U-0 | U-0 | U-0 | R-0/0 | R-0/0 | R-0/0 | R-0/0 |
|---|---|---|---|---|---|---|---|
| — | — | — | — | MC4OUT | MC3OUT | MC2OUT | MC1OUT |
| bit 7 | | | | | | | bit 0 |

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

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

bit 3 **MC4OUT:** Mirror Copy of C4OUT bit

bit 2 **MC3OUT:** Mirror Copy of C3OUT bit

bit 1 **MC2OUT:** Mirror Copy of C2OUT bit

bit 0 **MC1OUT:** Mirror Copy of C1OUT bit

**TABLE 21-3: SUMMARY OF REGISTERS ASSOCIATED WITH COMPARATOR MODULE**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| ANSELA | ANSA7 | — | ANSA5 | ANSA4 | ANSA3 | ANSA2 | ANSA1 | ANSA0 | 137 |
| ANSELB | — | ANSB6 | ANSB5 | ANSB4 | ANSB3 | ANSB2 | ANSB1 | ANSB0 | 143 |
| CM1CON0 | C1ON | C1OUT | C1OE | C1POL | C1ZLF | C1SP | C1HYS | C1SYNC | 203 |
| CM2CON0 | C2ON | C2OUT | C2OE | C2POL | C2ZLF | C2SP | C2HYS | C2SYNC | 203 |
| CM1CON1 | C1NTP | C1INTN | C1PCH<2:0> | | | C1NCH<2:0> | | | 204 |
| CM2CON1 | C2NTP | C2INTN | C2PCH<2:0> | | | C2NCH<2:0> | | | 204 |
| CM3CON0 | C3ON | C3OUT | C3OE | C3POL | C3ZLF | C3SP | C3HYS | C3SYNC | 203 |
| CM3CON1 | C3INTP | C3INTN | C3PCH<2:0> | | | C3NCH<2:0> | | | 204 |
| CMOUT | — | — | — | — | MC4OUT | MC3OUT | MC2OUT | MC1OUT | 205 |
| FVRCON | FVREN | FVRRDY | TSEN | TSRNG | CDAFVR<1:0> | | ADFVR<1:0> | | 167 |
| DAC1CON0 | DAC1EN | — | DAC1OE1 | DAC1OE2 | DAC1PSS<1:0> | | — | DAC1NSS | 192 |
| DAC1CON1 | DAC1R<7:0> | | | | | | | | 192 |
| INTCON | GIE | PEIE | TMR0IE | INTE | IOCIE | TMR0IF | INTF | IOCIF | 97 |
| PIE2 | OSFIE | C2IE | C1IE | EEIE | BCL1IE | C4IE | C3IE | CCP2IE | 99 |
| PIR2 | OSFIF | C2IF | C1IF | EEIF | BCL1IF | C4IF | C3IF | CCP2IF | 103 |
| TRISA | TRISA7 | TRISA6 | TRISA5 | TRISA4 | TRISA3 | TRISA2 | TRISA1 | TRISA0 | 137 |
| TRISB | TRISB7 | TRISB6 | TRISB5 | TRISB4 | TRISB3 | TRISB2 | TRISB1 | TRISB0 | 143 |
| TRISC | TRISC7 | TRISC6 | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 | 147 |

**Note 1:** — = unimplemented location, read as ‘0’. Shaded cells are unused by the comparator module.

## 23.7 Timer1 Interrupt

The Timer1 register pair (TMR1H:TMR1L) increments to FFFFh and rolls over to 0000h. When Timer1 rolls over, the Timer1 interrupt flag bit of the PIR1 register is set. To enable the interrupt on rollover, you must set these bits:

- TMR1ON bit of the T1CON register
- TMR1IE bit of the PIE1 register
- PEIE bit of the INTCON register
- GIE bit of the INTCON register

The interrupt is cleared by clearing the TMR1IF bit in the Interrupt Service Routine.

| **Note:** | The TMR1H:TMR1L register pair and the TMR1IF bit should be cleared before enabling interrupts. |
|---|---|

## 23.8 Timer1 Operation During Sleep

Timer1 can only operate during Sleep when setup in Asynchronous Counter mode. In this mode, an external crystal or clock source can be used to increment the counter. To set up the timer to wake the device:

- TMR1ON bit of the T1CON register must be set
- TMR1IE bit of the PIE1 register must be set
- PEIE bit of the INTCON register must be set
- $\overline{\text{T1SYNC}}$ bit of the T1CON register must be set
- TMR1CS bits of the T1CON register must be configured
- T1OSCEN bit of the T1CON register must be configured

The device will wake-up on an overflow and execute the next instructions. If the GIE bit of the INTCON register is set, the device will call the Interrupt Service Routine.

Timer1 oscillator will continue to operate in Sleep regardless of the $\overline{\text{T1SYNC}}$ bit setting.

## 23.9 CCP Capture/Compare Time Base

The CCP modules use the TMR1H:TMR1L register pair as the time base when operating in Capture or Compare mode.

In Capture mode, the value in the TMR1H:TMR1L register pair is copied into the CCPR1H:CCPR1L register pair on a configured event.

In Compare mode, an event is triggered when the value CCPR1H:CCPR1L register pair matches the value in the TMR1H:TMR1L register pair. This event can be a Auto-conversion Trigger.

For more information, see **Section 25.0 "Capture/Compare/PWM Modules"**.

## 23.10 CCP Auto-Conversion Trigger

When any of the CCP's are configured to trigger a auto-conversion, the trigger will clear the TMR1H:TMR1L register pair. This auto-conversion does not cause a Timer1 interrupt. The CCP module may still be configured to generate a CCP interrupt.

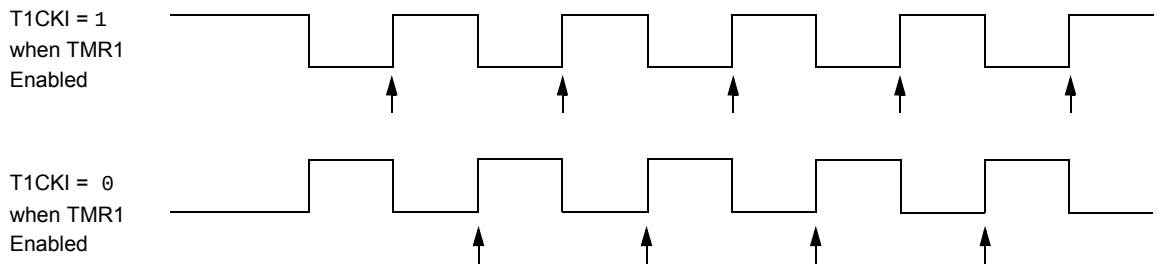
In this mode of operation, the CCPR1H:CCPR1L register pair becomes the period register for Timer1.

Timer1 should be synchronized and FOSC/4 should be selected as the clock source in order to utilize the Auto-conversion Trigger. Asynchronous operation of Timer1 can cause a Auto-conversion Trigger to be missed.

In the event that a write to TMR1H or TMR1L coincides with a Auto-conversion Trigger from the CCP, the write will take precedence.

For more information, see **Section 25.2.4 "Auto-Conversion Trigger"**.

**FIGURE 23-2: TIMER1 INCREMENTING EDGE**




**Note 1:** Arrows indicate counter increments.

**2:** In Counter mode, a falling edge must be registered by the counter prior to the first incrementing rising edge of the clock.

#### 26.2.1.4 16-bit Duty Cycle Register

The PSMCxDC Duty Cycle register is used to determine a synchronous falling edge event referenced to the 16-bit PSMCxTMR digital counter. A match between the PSMCxTMR and PSMCxDC register values will generate a falling edge event.

The match will generate a duty cycle match interrupt, thereby setting the PxTDCIF bit of the PSMC Time Base Interrupt Control (PSMCxINT) register (Register 26-34).

The 16-bit duty cycle value is accessible to software as two 8-bit registers:

- PSMC Duty Cycle Count Low Byte (PSMCxDCL) register (Register 26-23)
- PSMC Duty Cycle Count High Byte (PSMCxDCH) register (Register 26-24)

The 16-bit duty cycle value is double-buffered before it is presented to the 16-bit time base for comparison. The buffered registers are updated on the first period event Reset after the PSMCxLD bit of the PSMCxCON register is set.

When the period, phase, and duty cycle are all determined from the time base, the effective PWM duty cycle can be expressed as shown in Equation 26-2.

**EQUATION 26-2: PWM DUTY CYCLE**

$$DUTYCYCLE = \frac{\text{PSMCxDC[15:0]} - \text{PSMCxPH[15:0]}}{(\text{PSMCxPR[15:0]} + 1)}$$

### 26.2.2 0% DUTY CYCLE OPERATION USING TIME BASE

To configure the PWM for 0% duty cycle set PSMCxDC<15:0> = PSMCxPH<15:0>. This will trigger a falling edge event simultaneous with the rising edge event and prevent the PWM from being asserted.

### 26.2.3 100% DUTY CYCLE OPERATION USING TIME BASE

To configure the PWM for 100% duty cycle set PSMCxDC<15:0> > PSMCxPR<15:0>.

This will prevent a falling edge event from occurring as the PSMCxDC<15:0> value and the time base value PSMCxTMR<15:0> will never be equal.

### 26.2.4 TIME BASE INTERRUPT GENERATION

The Time Base section can generate four unique interrupts:

- Time Base Counter Overflow Interrupt
- Time Base Phase Register Match Interrupt
- Time Base Duty Cycle Register Match Interrupt
- Time Base Period Register Match Interrupt

Each interrupt has an interrupt flag bit and an interrupt enable bit. The interrupt flag bit is set anytime a given event occurs, regardless of the status of the enable bit.

Time base interrupt enables and flags are located in the PSMC Time Base Interrupt Control (PSMCxINT) register (Register 26-34).

PSMC time base interrupts also require that the PSMCxTIE bit in the PIE4 register and the PEIE and GIE bits in the INTCON register be set in order to generate an interrupt. The PSMCxTIF interrupt flag in the PIR4 register will only be set by a time base interrupt when one or more of the enable bits in the PSMCxINT register is set.

The interrupt flag bits need to be cleared in software. However, all PMSCx time base interrupt flags, except PSMCxTIF, are cleared when the PSMCxEN bit is cleared.

Interrupt bits that are set by software will generate an interrupt provided that the corresponding interrupt is enabled.

| **Note:** | Interrupt flags in both the PIE4 and PSMCxINT registers must be cleared to clear the interrupt. The PSMCxINT flags must be cleared first. |
|---|---|

### 26.2.5 PSMC TIME BASE CLOCK SOURCES

There are three clock sources available to the module:

- Internal 64 MHz clock
- Fosc system clock
- External clock input pin

The clock source is selected with the PxCSRC<1:0> bits of the PSMCx Clock Control (PSMCxCLK) register (Register 26-7).

When the Internal 64 MHz clock is selected as the source, the HFINTOSC continues to operate and clock the PSMC circuitry in Sleep. However, the system clock to other peripherals and the CPU is suppressed.

| **Note:** | When the 64 MHz clock is selected, the clock continues to operate in Sleep, even when the PSMC is disabled (PSMCxEN = 0). Select a clock other than the 64 MHz clock to minimize power consumption when the PSMC is not enabled. |
|---|---|

The Internal 64 MHz clock utilizes the system clock 4x PLL. When the system clock source is external and the PSMC is using the Internal 64 MHz clock, the 4x PLL should not be used for the system clock.

### 26.3.2 COMPLEMENTARY PWM

The complementary PWM uses the same events as the single PWM, but two waveforms are generated instead of only one.

The two waveforms are opposite in polarity to each other. The two waveforms may also have dead-band control as well.

#### 26.3.2.1 Mode Features and Controls

- Dead-band control available
- PWM primary output can be steered to the following pins:
  - PSMCxA
  - PSMCxC
  - PSMCxE
- PWM complementary output can be steered to the following pins:
  - PSMCxB
  - PSMCxD
  - PSMCxE

#### 26.3.2.2 Waveform Generation

Rising Edge Event

- Complementary output is set inactive
- Optional rising edge dead band is activated
- Primary output is set active

Falling Edge Event

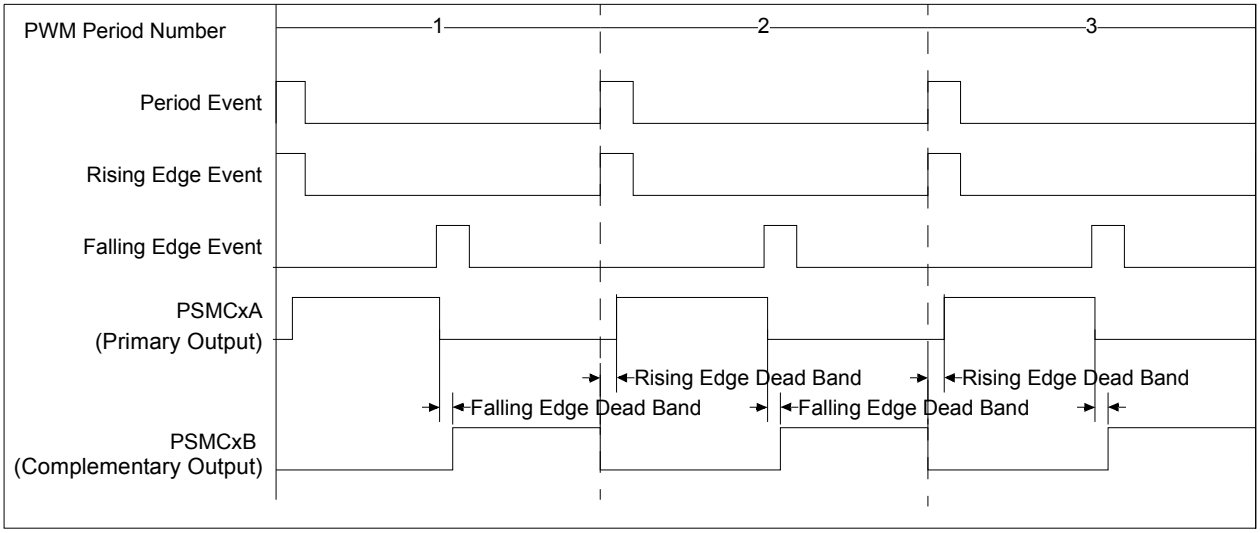
- Primary output is set inactive
- Optional falling edge dead band is activated
- Complementary output is set active

Code for setting up the PSMC generate the complementary single-phase waveform shown in Figure 26-5, and given in Example 26-2.

**EXAMPLE 26-2: COMPLEMENTARY SINGLE-PHASE SETUP**

```
; Complementary Single-phase PWM PSMC setup
; Fully synchronous operation
; Period = 10 us
; Duty cycle = 50%
; Deadband = 93.75 +15.6/-0 ns
   BANKSEL PSMC1CON
   MOVLW   0x02        ; set period
   MOVWF   PSMC1PRH
   MOVLW   0x7F
   MOVWF   PSMC1PRL
   MOVLW   0x01        ; set duty cycle
   MOVWF   PSMC1DCH
   MOVLW   0x3F
   MOVWF   PSMC1DCL
   CLRF    PSMC1PHH    ; no phase offset
   CLRF    PSMC1PHL
   MOVLW   0x01        ; PSMC clock=64 MHz
   MOVWF   PSMC1CLK
; output on A, normal polarity
   MOVLW   B'00000011'; A and B enables
   MOVWF   PSMC1OEN
   MOVWF   PSMC1STR0
   CLRF    PSMC1POL
; set time base as source for all events
   BSF     PSMC1PRS, P1PRST
   BSF     PSMC1PHS, P1PHST
   BSF     PSMC1DCS, P1DCST
; set rising and falling dead-band times
   MOVLW   D'6'
   MOVWF   PSMC1DBR
   MOVWF   PSMC1DBF
; enable PSMC in Complementary Single Mode
; this also loads steering and time buffers
; and enables rising and falling deadbands
   MOVLW   B'11110001'
   MOVWF   PSMC1CON
   BANKSEL TRISC
   BCF     TRISC, 0    ; enable pin drivers
   BCF     TRISC, 1
```

**FIGURE 26-5: COMPLEMENTARY PWM WAVEFORM – PSMCXSTR0 = 03H**



**REGISTER 26-21: PSMCxPHL: PSMC PHASE COUNT LOW BYTE REGISTER**

| R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| PSMCxPHL<7:0> | | | | | | | |
| bit 7 | | | | | | | bit 0 |

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

bit 7-0 **PSMCxPHL<7:0>:** 16-bit Phase Count Least Significant bits
= PSMCxPH<7:0>

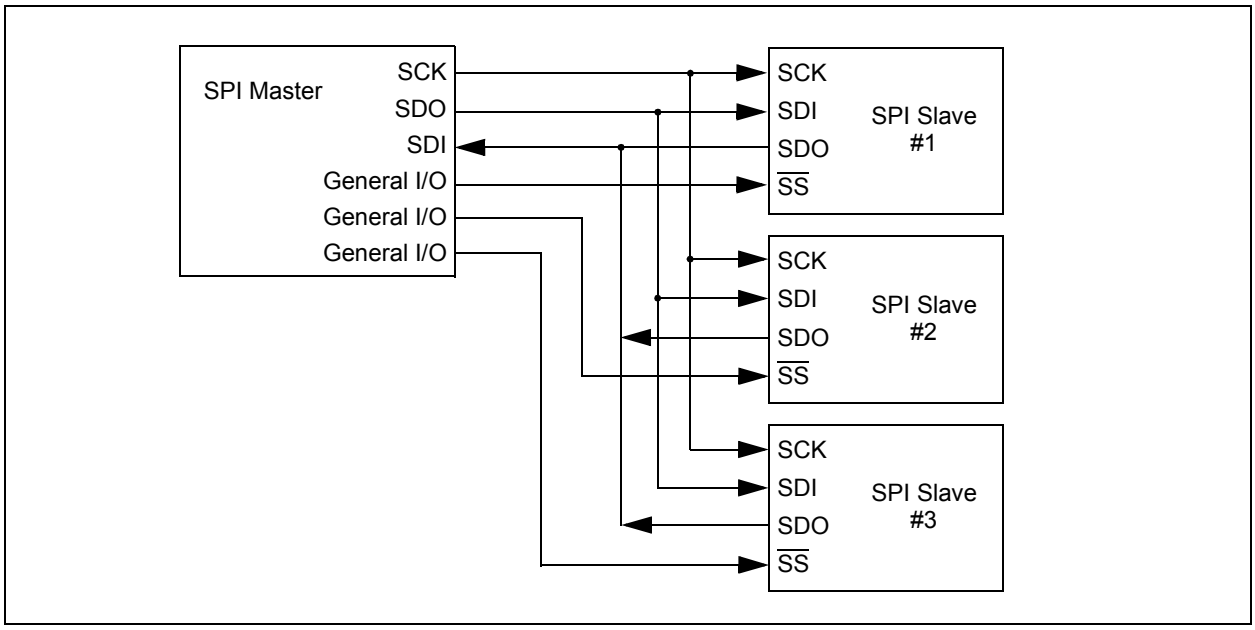
**REGISTER 26-22: PSMCxPHH: PSMC PHASE COUNT HIGH BYTE REGISTER**

| R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| PSMCxPHH<7:0> | | | | | | | |
| bit 7 | | | | | | | bit 0 |

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

bit 7-0 **PSMCxPHH<7:0>:** 16-bit Phase Count Most Significant bits
= PSMCxPH<15:8>

**FIGURE 27-4: SPI MASTER AND MULTIPLE SLAVE CONNECTION**




27.2.1 SPI MODE REGISTERS

The MSSP module has five registers for SPI mode operation. These are:

- MSSP STATUS register (SSPSTAT)
- MSSP Control register 1 (SSPCON1)
- MSSP Control register 3 (SSPCON3)
- MSSP Data Buffer register (SSPBUF)
- MSSP Address register (SSPADD)
- MSSP Shift register (SSPSR)
  (Not directly accessible)

SSPCON1 and SSPSTAT are the control and STATUS registers in SPI mode operation. The SSPCON1 register is readable and writable. The lower six bits of the SSPSTAT are read-only. The upper two bits of the SSPSTAT are read/write.

In one SPI master mode, SSPADD can be loaded with a value used in the Baud Rate Generator. More information on the Baud Rate Generator is available in **Section 27.7 "Baud Rate Generator"**.

SSPSR is the shift register used for shifting data in and out. SSPBUF provides indirect access to the SSPSR register. SSPBUF is the buffer register to which data bytes are written, and from which data bytes are read.

In receive operations, SSPSR and SSPBUF together create a buffered receiver. When SSPSR receives a complete byte, it is transferred to SSPBUF and the SSP1IF interrupt is set.

During transmission, the SSPBUF is not buffered. A write to SSPBUF will write to both SSPBUF and SSPSR.

### 27.5.2 SLAVE RECEPTION

When the R/$\overline{W}$ bit of a matching received address byte is clear, the R/$\overline{W}$ bit of the SSPSTAT register is cleared. The received address is loaded into the SSPBUF register and acknowledged.

When the overflow condition exists for a received address, then not Acknowledge is given. An overflow condition is defined as either bit BF of the SSPSTAT register is set, or bit SSPOV of the SSPCON1 register is set. The BOEN bit of the SSPCON3 register modifies this operation. For more information see Register 27-4.

An MSSP interrupt is generated for each transferred data byte. Flag bit, SSP1IF, must be cleared by software.

When the SEN bit of the SSPCON2 register is set, SCL will be held low (clock stretch) following each received byte. The clock must be released by setting the CKP bit of the SSPCON1 register, except sometimes in 10-bit mode. See **Section 27.2.3 "SPI Master Mode"** for more detail.

#### 27.5.2.1 7-bit Addressing Reception

This section describes a standard sequence of events for the MSSP module configured as an $I^2C$ Slave in 7-bit Addressing mode. All decisions made by hardware or software and their effect on reception. Figure 27-13 and Figure 27-14 is used as a visual reference for this description.

This is a step by step process of what typically must be done to accomplish $I^2C$ communication.

1. Start bit detected.
2. S bit of SSPSTAT is set; SSP1IF is set if interrupt on Start detect is enabled.
3. Matching address with R/$\overline{W}$ bit clear is received.
4. The slave pulls SDA low sending an $\overline{ACK}$ to the master, and sets SSP1IF bit.
5. Software clears the SSP1IF bit.
6. Software reads received address from SSPBUF clearing the BF flag.
7. If SEN = 1; Slave software sets CKP bit to release the SCL line.
8. The master clocks out a data byte.
9. Slave drives SDA low sending an $\overline{ACK}$ to the master, and sets SSP1IF bit.
10. Software clears SSP1IF.
11. Software reads the received byte from SSPBUF clearing BF.
12. Steps 8-12 are repeated for all received bytes from the master.
13. Master sends Stop condition, setting P bit of SSPSTAT, and the bus goes idle.

#### 27.5.2.2 7-bit Reception with AHEN and DHEN

Slave device reception with AHEN and DHEN set operate the same as without these options with extra interrupts and clock stretching added after the 8th falling edge of SCL. These additional interrupts allow the slave software to decide whether it wants to $\overline{ACK}$ the receive address or data byte, rather than the hardware. This functionality adds support for PMBus™ that was not present on previous versions of this module.

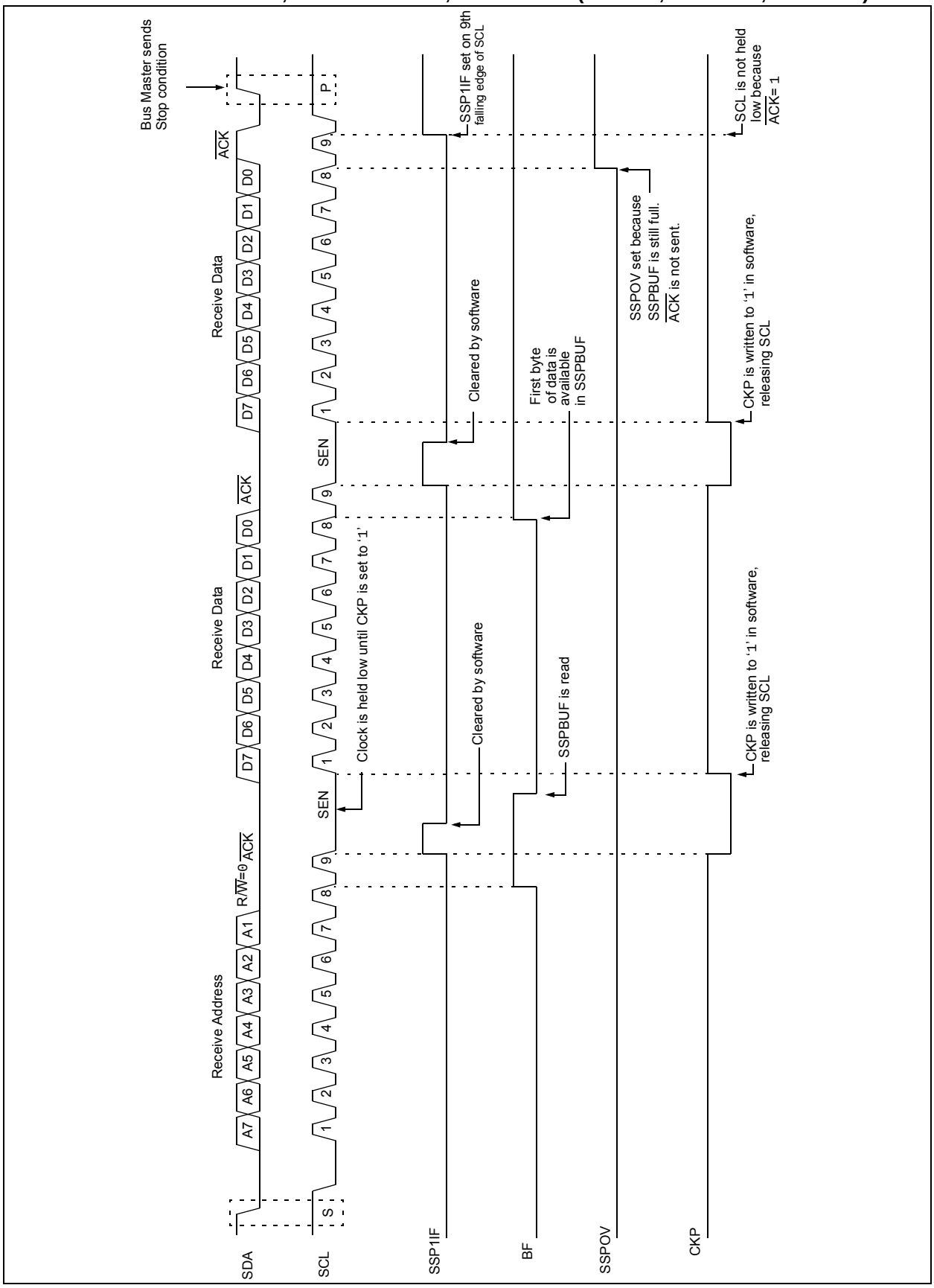
This list describes the steps that need to be taken by slave software to use these options for $I^2C$ communication. Figure 27-15 displays a module using both address and data holding. Figure 27-16 includes the operation with the SEN bit of the SSPCON2 register set.

1. S bit of SSPSTAT is set; SSP1IF is set if interrupt on Start detect is enabled.
2. Matching address with R/$\overline{W}$ bit clear is clocked in. SSP1IF is set and CKP cleared after the 8th falling edge of SCL.
3. Slave clears the SSP1IF.
4. Slave can look at the ACKTIM bit of the SSPCON3 register to determine if the SSP1IF was after or before the $\overline{ACK}$.
5. Slave reads the address value from SSPBUF, clearing the BF flag.
6. Slave sets $\overline{ACK}$ value clocked out to the master by setting ACKDT.
7. Slave releases the clock by setting CKP.
8. SSP1IF is set after an $\overline{ACK}$, not after a NACK.
9. If SEN = 1 the slave hardware will stretch the clock after the $\overline{ACK}$.
10. Slave clears SSP1IF.

**Note:** SSP1IF is still set after the 9th falling edge of SCL even if there is no clock stretching and BF has been cleared. Only if NACK is sent to master is SSP1IF not set

11. SSP1IF set and CKP cleared after 8th falling edge of SCL for a received data byte.
12. Slave looks at ACKTIM bit of SSPCON3 to determine the source of the interrupt.
13. Slave reads the received data from SSPBUF clearing BF.
14. Steps 7-14 are the same for each received data byte.
15. Communication is ended by either the slave sending an $\overline{ACK}$ = 1, or the master sending a Stop condition. If a Stop is sent and Interrupt on Stop Detect is disabled, the slave will only know by polling the P bit of the SSTSTAT register.

**FIGURE 27-15: I²C SLAVE, 7-BIT ADDRESS, RECEPTION (SEN = 1, AHEN = 0, DHEN = 0)**



### 28.1.2 EUSART ASYNCHRONOUS RECEIVER

The Asynchronous mode is typically used in RS-232 systems. The receiver block diagram is shown in Figure 28-2. The data is received on the RX/DT pin and drives the data recovery block. The data recovery block is actually a high-speed shifter operating at 16 times the baud rate, whereas the serial Receive Shift Register (RSR) operates at the bit rate. When all eight or nine bits of the character have been shifted in, they are immediately transferred to a two character First-In-First-Out (FIFO) memory. The FIFO buffering allows reception of two complete characters and the start of a third character before software must start servicing the EUSART receiver. The FIFO and RSR registers are not directly accessible by software. Access to the received data is via the RCREG register.

#### 28.1.2.1 Enabling the Receiver

The EUSART receiver is enabled for asynchronous operation by configuring the following three control bits:

- CREN = 1
- SYNC = 0
- SPEN = 1

All other EUSART control bits are assumed to be in their default state.

Setting the CREN bit of the RCSTA register enables the receiver circuitry of the EUSART. Clearing the SYNC bit of the TXSTA register configures the EUSART for asynchronous operation. Setting the SPEN bit of the RCSTA register enables the EUSART. The programmer must set the corresponding TRIS bit to configure the RX/DT I/O pin as an input.

| **Note:** | If the RX/DT function is on an analog pin, the corresponding ANSEL bit must be cleared for the receiver to function. |
|---|---|

#### 28.1.2.2 Receiving Data

The receiver data recovery circuit initiates character reception on the falling edge of the first bit. The first bit, also known as the Start bit, is always a zero. The data recovery circuit counts one-half bit time to the center of the Start bit and verifies that the bit is still a zero. If it is not a zero then the data recovery circuit aborts character reception, without generating an error, and resumes looking for the falling edge of the Start bit. If the Start bit zero verification succeeds then the data recovery circuit counts a full bit time to the center of the next bit. The bit is then sampled by a majority detect circuit and the resulting '0' or '1' is shifted into the RSR. This repeats until all data bits have been sampled and shifted into the RSR. One final bit time is measured and the level sampled. This is the Stop bit, which is always a '1'. If the data recovery circuit samples a '0' in the Stop bit position then a framing error is set for this character, otherwise the framing error is cleared for this character. See **Section 28.1.2.4 "Receive Framing Error"** for more information on framing errors.

Immediately after all data bits and the Stop bit have been received, the character in the RSR is transferred to the EUSART receive FIFO and the RCIF interrupt flag bit of the PIR1 register is set. The top character in the FIFO is transferred out of the FIFO by reading the RCREG register.

| **Note:** | If the receive FIFO is overrun, no additional characters will be received until the overrun condition is cleared. See **Section 28.1.2.5 "Receive Overrun Error"** for more information on overrun errors. |
|---|---|

#### 28.1.2.3 Receive Interrupts

The RCIF interrupt flag bit of the PIR1 register is set whenever the EUSART receiver is enabled and there is an unread character in the receive FIFO. The RCIF interrupt flag bit is read-only, it cannot be set or cleared by software.

RCIF interrupts are enabled by setting all of the following bits:

- RCIE, Interrupt Enable bit of the PIE1 register
- PEIE, Peripheral Interrupt Enable bit of the INTCON register
- GIE, Global Interrupt Enable bit of the INTCON register

The RCIF interrupt flag bit will be set when there is an unread character in the FIFO, regardless of the state of interrupt enable bits.

**TABLE 31-2: SUPPLY VOLTAGE ($I_{DD}$)(1,2) (CONTINUED)**

| PIC16LF1788/9 | | | Standard Operating Conditions (unless otherwise stated) | | | | |
|---|---|---|---|---|---|---|---|
| PIC16F1788/9 | | | | | | | |
| Param No. | Device Characteristics | Min. | Typ† | Max. | Units | Conditions | |
| | | | | | | $V_{DD}$ | Note |
| D014 | | — | 125 | 550 | μA | 1.8 | $F_{OSC}$ = 4 MHz<br>EC Oscillator mode<br>Medium-Power mode |
| | | — | 280 | 1100 | μA | 3.0 | |
| D014 | | — | 220 | 650 | μA | 2.3 | $F_{OSC}$ = 4 MHz<br>EC Oscillator mode **(Note 5)**<br>Medium-Power mode |
| | | — | 290 | 1000 | μA | 3.0 | |
| | | — | 350 | 1200 | μA | 5.0 | |
| D015 | | — | 2.1 | 6.2 | mA | 3.0 | $F_{OSC}$ = 32 MHz<br>EC Oscillator High-Power mode |
| | | — | 2.5 | 7.5 | mA | 3.6 | |
| D015 | | — | 2.1 | 6.5 | mA | 3.0 | $F_{OSC}$ = 32 MHz<br>EC Oscillator High-Power mode **(Note 5)** |
| | | — | 2.2 | 7.5 | mA | 5.0 | |
| D017 | | — | 130 | 180 | μA | 1.8 | $F_{OSC}$ = 500 kHz<br>MFINTOSC mode |
| | | — | 150 | 250 | μA | 3.0 | |
| D017 | | — | 150 | 250 | μA | 2.3 | $F_{OSC}$ = 500 kHz<br>MFINTOSC mode **(Note 5)** |
| | | — | 170 | 330 | μA | 3.0 | |
| | | — | 220 | 430 | μA | 5.0 | |
| D019 | | — | 0.8 | 2.2 | mA | 1.8 | $F_{OSC}$ = 16 MHz<br>HFINTOSC mode |
| | | — | 1.2 | 3.7 | mA | 3.0 | |
| D019 | | — | 1.0 | 2.3 | mA | 2.3 | $F_{OSC}$ = 16 MHz<br>HFINTOSC mode **(Note 5)** |
| | | — | 1.3 | 3.9 | mA | 3.0 | |
| | | — | 1.4 | 4.1 | mA | 5.0 | |
| D020 | | — | 2.1 | 6.2 | mA | 3.0 | $F_{OSC}$ = 32 MHz<br>HFINTOSC mode |
| | | — | 2.5 | 7.5 | mA | 3.6 | |
| D020 | | — | 2.1 | 6.5 | mA | 3.0 | $F_{OSC}$ = 32 MHz<br>HFINTOSC mode |
| | | — | 2.2 | 7.5 | mA | 5.0 | |
| D022 | | — | 2.1 | 6.2 | mA | 3.0 | $F_{OSC}$ = 32 MHz<br>HS Oscillator mode **(Note 6)** |
| | | — | 2.5 | 7.5 | mA | 3.6 | |
| D022 | | — | 2.1 | 6.5 | mA | 3.0 | $F_{OSC}$ = 32 MHz<br>HS Oscillator mode **(Note 5, 6)** |
| | | — | 2.2 | 7.5 | mA | 5.0 | |

**Note 1:** The test conditions for all $I_{DD}$ measurements in active operation mode are: OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to $V_{DD}$; $\overline{MCLR}$ = $V_{DD}$; WDT disabled.

**2:** The supply current is mainly a function of the operating voltage and frequency. Other factors, such as I/O pin loading and switching rate, oscillator type, internal code execution pattern and temperature, also have an impact on the current consumption.

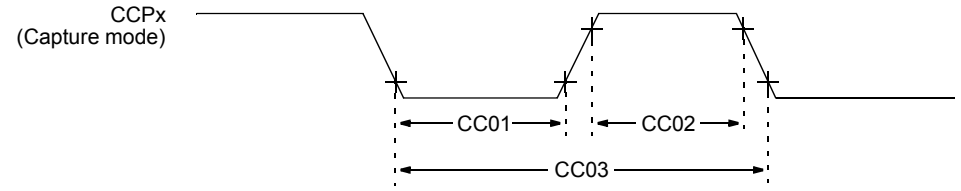
**3:** For RC oscillator configurations, current through $R_{EXT}$ is not included. The current through the resistor can be extended by the formula $I_R = V_{DD}/2R_{EXT}$ (mA) with $R_{EXT}$ in kΩ.

**4:** FVR and BOR are disabled.

**5:** 0.1 μF capacitor on $V_{CAP}$.

**6:** 8 MHz crystal oscillator with 4x PLL enabled.

**FIGURE 31-11: CAPTURE/COMPARE/PWM TIMINGS (CCP)**




**Note:** Refer to Figure 31-5 for load conditions.

**TABLE 31-12: CAPTURE/COMPARE/PWM REQUIREMENTS (CCP)**

| **Standard Operating Conditions (unless otherwise stated)** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym.** | **Characteristic** | | **Min.** | **Typ†** | **Max.** | **Units** | **Conditions** |
| CC01* | TccL | CCPx Input Low Time | No Prescaler | 0.5TCY + 20 | — | — | ns | |
| | | | With Prescaler | 20 | — | — | ns | |
| CC02* | TccH | CCPx Input High Time | No Prescaler | 0.5TCY + 20 | — | — | ns | |
| | | | With Prescaler | 20 | — | — | ns | |
| CC03* | TccP | CCPx Input Period | | $\frac{3T_{CY} + 40}{N}$ | — | — | ns | N = prescale value (1, 4 or 16) |

* These parameters are characterized but not tested.

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

**Note:** Unless otherwise noted, $V_{IN}$ = 5V, $F_{OSC}$ = 300 kHz, $C_{IN}$ = 0.1 µF, $T_A$ = 25°C.

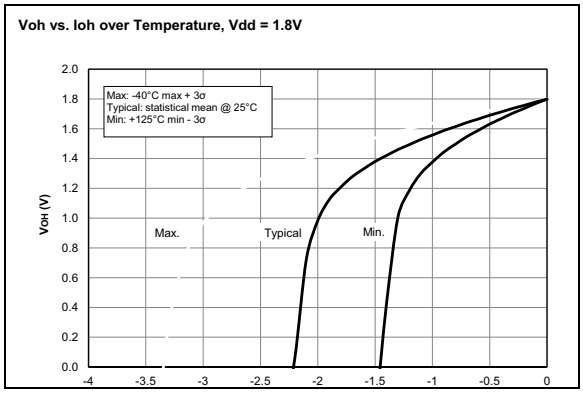



***FIGURE 32-55:*** *$V_{OH}$ vs. $I_{OH}$ Over Temperature, $V_{DD}$ = 1.8V, PIC16LF1788/9 Only.*

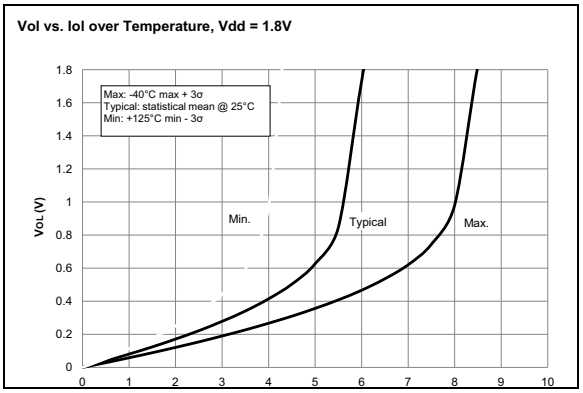



***FIGURE 32-56:*** *$V_{OL}$ vs. $I_{OL}$ Over Temperature, $V_{DD}$ = 1.8V, PIC16LF1788/9 Only.*

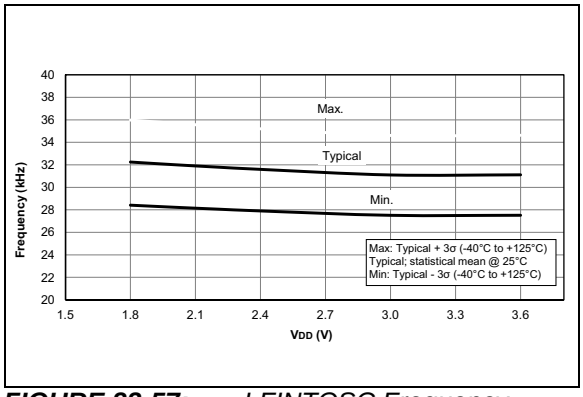



***FIGURE 32-57:*** *LFINTOSC Frequency, PIC16LF1788/9 Only.*

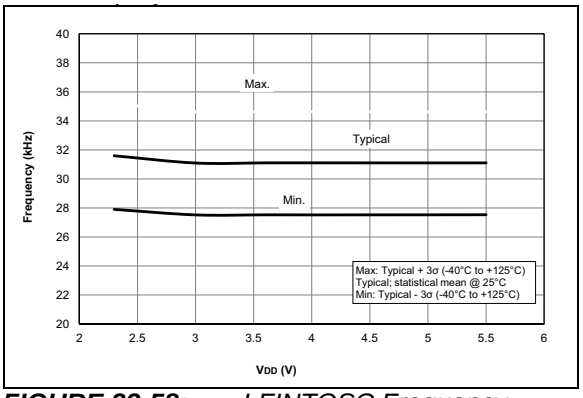



***FIGURE 32-58:*** *LFINTOSC Frequency, PIC16F1788/9 Only.*

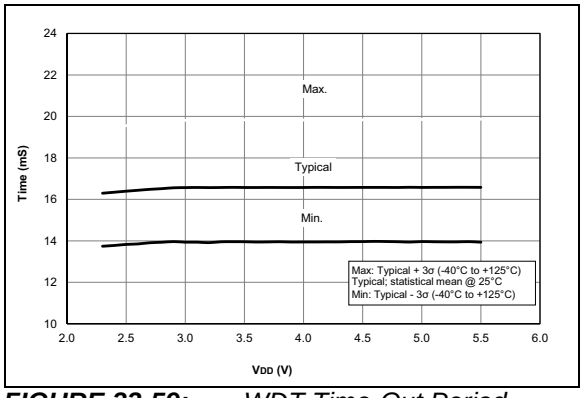



***FIGURE 32-59:*** *WDT Time-Out Period, PIC16F1788/9 Only.*

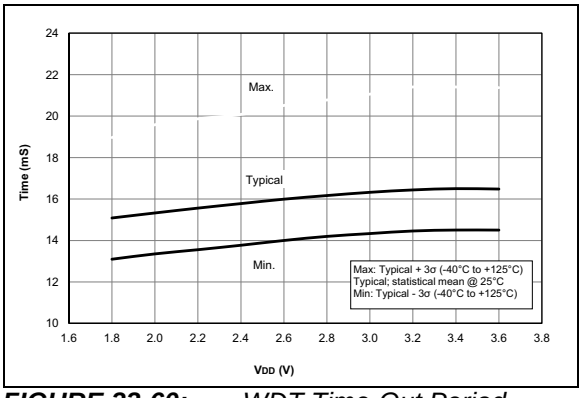



***FIGURE 32-60:*** *WDT Time-Out Period, PIC16LF1788/9 Only.*