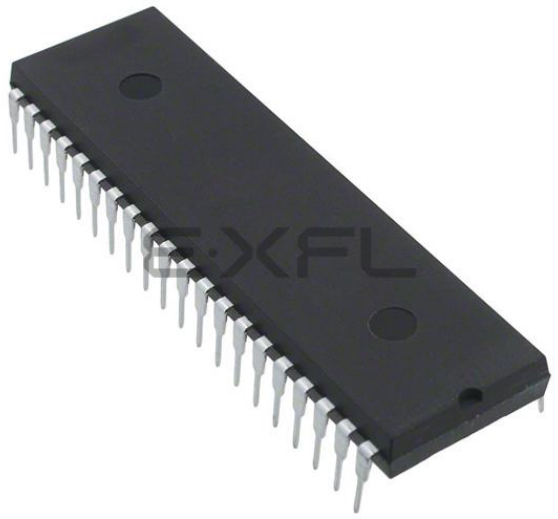
# Microchip Technology - PIC16LF19176-I/P Datasheet



**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, LCD, POR, PWM, WDT |
| Number of I/O | 35 |
| 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) | 1.8V ~ 3.6V |
| Data Converters | A/D 31x12b; D/A 1x5b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Through Hole |
| Package / Case | 40-DIP (0.600", 15.24mm) |
| Supplier Device Package | 40-PDIP |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic16lf19176-i-p |

### TABLE 4: 40/44-PIN ALLOCATION TABLE (PIC16(L)F19175/76) (CONTINUED)

| I/O(2) | 40-Pin PDIP | 40-Pin UQFN | 44-Pin TQFP | 44-Pin QFN | ADC | Reference | Comparator | Zero-Cross Detect | DAC | Timers/SMT | CCP | PWM | CWG | MSSP | EUSART | CLC | RTCC | LCD | Interrupt-on-Change | High Current | Pull-up | Basic |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| VLCD3 | 24 | 39 | 43 | 43 | — | — | — | — | — | — | — | — | — | — | — | — | — | VLCD3 | — | — | — | — |
| $V_{DD}$ | 11<br>32 | 7<br>26 | 7<br>28 | 7<br>28 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | $V_{DD}$ |
| $V_{SS}$ | 12<br>31 | 6<br>27 | 6<br>29 | 6<br>30 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | $V_{SS}$ |
| OUT(2) | — | — | — | — | ADGRDA<br>ADGRDB | — | C1OUT<br>C2OUT | — | — | TMR0 | CCP1<br>CCP2 | PWM3<br>PWM4 | CWG1A<br>CWG1B<br>CWG1C<br>CWG1D | SDO<br>SCK<br>SCL<br>SDA | TX1<br>DT1<br>CK1<br>TX2<br>DT2<br>CK2 | CLC1OUT | RTCC | — | — | — | — | — |

**Note 1:** This is a PPS remappable input signal. The input function may be moved from the default location shown to one of several other PORTx pins.

**2:** All digital output signals shown in this row are PPS remappable. These signals may be mapped to output onto one or more PORTx pin options.

**3:** This is a bidirectional signal. For normal module operation, the firmware should map this signal to the same pin in both the PPS input and PPS output registers.

**4:** These pins are configured for $I^2C$ logic levels. PPS assignments to the other pins will operate, but input logic levels will be standard TTL/ST as selected by INLCVL register, instead of the $I^2C$ specific or SMBUS input buffer thresholds.

**5:** These are alternative $I^2C$ logic levels pins.

**6:** In $I^2C$ logic levels configuration, these pins can operate as either SCL and SDA pins.

### TABLE 1-3: PIC16(L)F19175/76 PINOUT DESCRIPTION (CONTINUED)

| Name | Function | Input Type | Output Type | Description |
|---|---|---|---|---|
| RB3/C1IN2-/C2IN2-/IOCB3/ANB3/SEG11/SEGCFLY2 | RB3 | TTL/ST | CMOS/OD | General purpose I/O. |
| | C1IN2- | AN | — | Comparator negative input. |
| | C2IN2- | AN | — | Comparator negative input. |
| | IOCB3 | TTL/ST | — | Interrupt-on-change input. |
| | ANB3 | AN | — | ADC Channel input. |
| | SEG11 | — | AN | LCD Analog output. |
| | SEGCFLY2 | AN | — | LCD Drive Charge Pump Capacitor Inputs |
| RB4/ADCACT(1)/IOCB4/ANB4/COM0 | RB4 | TTL/ST | CMOS/OD | General purpose I/O. |
| | ADCACT(1) | TTL/ST | — | ADC Auto-Conversion Trigger input |
| | IOCB4 | TTL/ST | — | Interrupt-on-change input. |
| | ANB4 | AN | — | ADC Channel input. |
| | COM0 | — | AN | LCD Driver Common Outputs. |
| RB5/T1G(1)/IOCB5/ANB5/SEG13/COM1 | RB5 | TTL/ST | CMOS/OD | General purpose I/O. |
| | T1G(1) | — | — | Timer1 Gate input. |
| | IOCB5 | TTL/ST | — | Interrupt-on-change input. |
| | ANB5 | AN | — | ADC Channel input. |
| | SEG13 | - | AN | LCD Analog output. |
| | COM1 | - | AN | LCD Driver Common Outputs. |
| RB6/CK2(3)/TX2(1)/CLCIN2(1)/IOCB6/ANB6/SEG14/ ICSPCLK | RB6 | TTL/ST | CMOS/OD | General purpose I/O. |
| | CK2(3) | — | — | EUSART synchronous clock out |
| | TX2(1) | — | — | EUSART asynchronous TX data out |
| | CLCIN2(1) | — | — | Configurable Logic Cell source input. |
| | IOCB6 | TTL/ST | — | Interrupt-on-change input. |
| | ANB6 | AN | — | ADC Channel input. |
| | SEG14 | — | AN | LCD Analog output. |
| | ICSPCLK | ST | — | In-Circuit Serial Programming™ and debugging clock input. |
| RB7/DK2(3)/RX2(1)/CLCIN3(1)/IOCB7/ANB7/SEG15/ DAC1OUT2/ICSPDAT | RB7 | TTL/ST | CMOS/OD | General purpose I/O. |
| | DK2(3) | — | — | EUSART synchronous data output |
| | RX2(1) | — | — | EUSART receive input. |
| | CLCIN3(1) | — | — | Configurable Logic Cell source input. |
| | IOCB7 | TTL/ST | — | Interrupt-on-change input. |
| | ANB7 | — | AN | ADC Channel input. |
| | SEG15 | — | AN | LCD Analog output. |
| | DAC1OUT2 | — | AN | Digital-to-Analog Converter output. |
| | ICSPDAT | TTL/ST | TTL/ST | In-Circuit Serial Programming™ and debugging data input/output. |

**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 $I^2C$
HV = High Voltage XTAL = Crystal levels

**Note 1:** This is a PPS remappable input signal. The input function may be moved from the default location shown to one of several other PORTx pins. Refer to Table 14-2 for details on which PORT pins may be used for this signal.

**2:** All output signals shown in this row are PPS remappable. These signals may be mapped to output onto one of several PORTx pin options as described in Table 14-3.

**3:** This is a bidirectional signal. For normal module operation, the firmware should map this signal to the same pin in both the PPS input and PPS output registers.

**4:** These pins are configured for $I^2C$ logic levels. The SCLx/SDAx signals may be assigned to any of the RB1/RB2/RC3/RC4 pins. PPS assignments to the other pins (e.g., RA5) will operate, but input logic levels will be standard TTL/ST, as selected by the INLVL register, instead of the $I^2C$ specific or SMBus input buffer thresholds.

### TABLE 1-4: PIC16(L)F19185/86 PINOUT DESCRIPTION

| Name | Function | Input Types | Output Types | Description |
|---|---|---|---|---|
| RA0/C1IN0-/C2IN0-/ANA0/CLCIN0(1)/IOCA0/SEG0 | RA0 | TTL/ST | CMOS/OD | General purpose I/O. |
| | C1IN0- | AN | — | Comparator negative input. |
| | C2IN0- | AN | — | Comparator negative input. |
| | ANA0 | AN | — | ADC Channel input. |
| | CLCIN0(1) | — | — | Configurable Logic Cell source input. |
| | IOCA0 | TTL/ST | — | Interrupt-on-change input. |
| | SEG0 | - | AN | LCD Analog output. |
| RA1/C1IN1-/C2IN1-/ANA1/CLCIN1(1)/SEG1 | RA1 | TTL/ST | CMOS/OD | General purpose I/O. |
| | C1IN1- | AN | — | Comparator negative input. |
| | C2IN1- | AN | — | Comparator negative input. |
| | ANA1 | AN | — | ADC Channel input. |
| | CLCIN1(1) | — | — | Configurable Logic Cell source input. |
| | IOCA1 | TTL/ST | — | Interrupt-on-change input. |
| | SEG1 | — | AN | LCD Analog output. |
| RA2/C1IN0+/C2IN0+/ANA2/DAC1OUT1/IOCA2/SEG2 | RA2 | TTL/ST | CMOS/OD | General purpose I/O. |
| | C1IN0+ | AN | — | Comparator positive input. |
| | C2IN0+ | AN | — | Comparator positive input. |
| | ANA2 | AN | — | ADC Channel input. |
| | DAC1OUT1 | — | AN | Digital-to-Analog Converter output. |
| | IOCA2 | TTL/ST | — | Interrupt-on-change input. |
| | SEG2 | — | AN | LCD Analog output. |
| RA3/C1IN1+/ANA3/SEG3/IOCA3/VREF+ (ADC)/VREF+ (DAC1) | RA3 | TTL/ST | CMOS/OD | General purpose I/O. |
| | C1IN1+ | AN | — | Comparator positive input. |
| | ANA3 | AN | — | ADC Channel input. |
| | SEG3 | — | AN | LCD Analog output. |
| | IOCA3 | TTL/ST | — | Interrupt-on-change input. |
| | VREF+ (ADC) | AN | — | ADC positive reference. |
| | VREF+ (DAC1) | AN | — | DAC positive reference. |
| RA4/ANA4/T0CKI(1)/IOCA4/SEG4/COM3 | RA4 | TTL/ST | CMOS/OD | General purpose I/O. |
| | ANA4 | AN | — | ADC Channel input. |
| | T0CKI(1) | — | — | Timer0 clock input. |
| | IOCA4 | TTL/ST | — | Interrupt-on-change input. |
| | SEG4 | — | AN | LCD Analog output. |
| | COM3 | — | AN | LCD Driver Common Outputs. |
| RA5/SS(1)/IOCA5/VBAT | RA5 | TTL/ST | CMOS/OD | General purpose I/O. |
| | SS(1) | TTL/ST | — | MSSP SPI slave select input. |
| | IOCA5 | TTL/ST | — | Interrupt-on-change input. |
| | VBAT | AN | — | RTCC Back-up Battery. |

**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 $I^2C$
HV = High Voltage XTAL = Crystal levels

**Note 1:** This is a PPS remappable input signal. The input function may be moved from the default location shown to one of several other PORTx pins. Refer to Table 14-2 for details on which PORT pins may be used for this signal.

**2:** All output signals shown in this row are PPS remappable. These signals may be mapped to output onto one of several PORTx pin options as described in Table 14-3.

**3:** This is a bidirectional signal. For normal module operation, the firmware should map this signal to the same pin in both the PPS input and PPS output registers.

**4:** These pins are configured for $I^2C$ logic levels. The SCLx/SDAx signals may be assigned to any of the RB1/RB2/RC3/RC4 pins. PPS assignments to the other pins (e.g., RA5) will operate, but input logic levels will be standard TTL/ST, as selected by the INLVL register, instead of the $I^2C$ specific or SMBus input buffer thresholds.

### EXAMPLE 13-5: DEVICE ID ACCESS

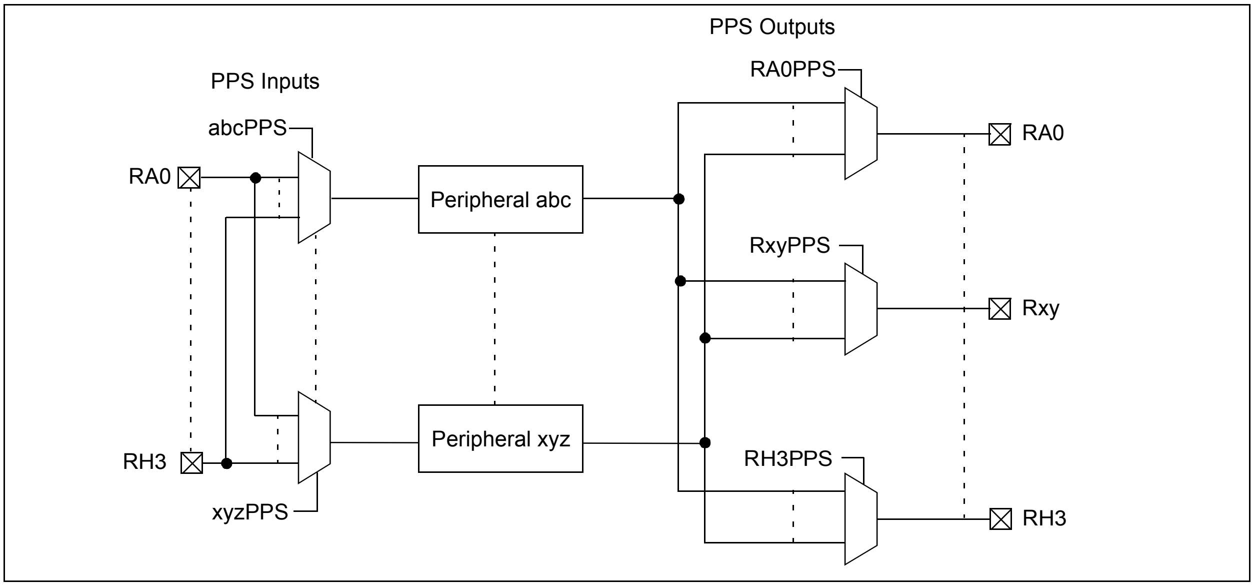
```
; This write routine assumes the following:
; 1. A full row of data are loaded, starting at the address in DATA_ADDR
; 2. Each word of data to be written is made up of two adjacent bytes in DATA_ADDR,
; stored in little endian format
; 3. A valid starting address (the least significant bits = 00000) is loaded in ADDRH:ADDRL
; 4. ADDRH and ADDRL are located in common RAM (locations 0x70 - 0x7F)
; 5. NVM interrupts are not taken into account
BANKSEL NVMADRH
MOVF ADDRH,W
MOVWF NVMADRH               ; Load initial address
MOVF ADDRL,W
MOVWF NVMADRL
MOVLW LOW DATA_ADDR            ; Load initial data address
MOVWF FSR0L
MOVLW HIGH DATA_ADDR
MOVWF FSR0H
BCF NVMCON1,NVMREGS           ; Set PFM as write location
BSF NVMCON1,WREN               ; Enable writes
BSF NVMCON1,LWLO             ; Load only write latches
LOOP
MOVIW FSR0++
MOVWF NVMDATL             ; Load first data byte
MOVIW FSR0++
MOVWF NVMDATH               ; Load second data byte
CALL UNLOCK_SEQ             ; If not, go load latch
INCF NVMADRL,F              ; Increment address
MOVF NVMADRL,W
XORLW 0x1F                  ; Check if lower bits of address are 00000
ANDLW 0x1F                  ; and if on last of 32 addresses
BTFSC STATUS,Z             ; Last of 32 words?
GOTO START_WRITE             ; If so, go write latches into memory
GOTO LOOP
START_WRITE
BCF NVMCON1,LWLO             ; Latch writes complete, now write memory
CALL UNLOCK_SEQ              ; Perform required unlock sequence
BCF NVMCON1,LWLO             ; Disable writes
UNLOCK_SEQ
MOVLW 55h
BCF INTCON,GIE                ; Disable interrupts
MOVWF NVMCON2                 ; Begin unlock sequence
MOVLW AAh
MOVWF NVMCON2
BSF NVMCON1,WR
BSF INTCON,GIE                ; Unlock sequence complete, re-enable interrupts
```

## 15.0 PERIPHERAL PIN SELECT (PPS) MODULE

The Peripheral Pin Select (PPS) module connects peripheral inputs and outputs to the device I/O pins. Only digital signals are included in the selections.

All analog inputs and outputs remain fixed to their assigned pins. Input and output selections are independent as shown in the simplified block diagram Figure 15-1.

**FIGURE 15-1: SIMPLIFIED PPS BLOCK DIAGRAM**



### 15.1 PPS Inputs

Each peripheral has a PPS register with which the inputs to the peripheral are selected. Inputs include the device pins.

Although every peripheral has its own PPS input selection register, the selections are identical for every peripheral as shown in Register 15-1.

| **Note:** | The notation “xxx” in the register name is a place holder for the peripheral identifier. For example, CLC1PPS. |
|---|---|

### 15.2 PPS Outputs

Each I/O pin has a PPS register with which the pin output source is selected. With few exceptions, the port TRIS control associated with that pin retains control over the pin output driver. Peripherals that control the pin output driver as part of the peripheral operation will override the TRIS control as needed. These peripherals are (See **Section 15.3 “Bidirectional Pins”**):

- EUSART (synchronous operation)
- MSSP ($I^2C$)
- CWG

Although every pin has its own PPS peripheral selection register, the selections are identical for every pin as shown in Register 15-2.

| **Note:** | The notation “Rxy” is a place holder for the pin port and bit identifiers. For example, x and y for PORTA bit 0 would be A and 0, respectively, resulting in the pin PPS output selection register RA0PPS. |
|---|---|

## 15.8 Register Definitions: PPS Input Selection

### REGISTER 15-1: xxxPPS: PERIPHERAL xxx INPUT SELECTION[1]

| U-0 | U-0 | U-0 | R/W-q/u | R/W/q/u | R/W-q/u | R/W-q/u | R/W-q/u |
|---|---|---|---|---|---|---|---|
| — | — | — | xxxPPS<4:0> | | | | |
| 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 | q = value depends on peripheral |

bit 7-5 **Unimplemented:** Read as ‘0’

bit 4-0 **xxxPPS<4:0>:** Peripheral xxx Input Selection bits
See Table 15-1 and Table 15-2.[2]

**Note 1:** The “xxx” in the register name “xxxPPS” represents the input signal function name, such as “INT”, “T0CKI”, “RX”, etc. This register summary shown here is only a prototype of the array of actual registers, as each input function has its own dedicated SFR (ex: INTPPS, T0CKIPPS, RXPPS, etc.).

**2:** Each specific input signal may only be mapped to a subset of these I/O pins, as shown in Table 15-2. Attempting to map an input signal to a non-supported I/O pin will result in undefined behavior. For example, the “INT” signal map be mapped to any PORTA or PORTB pin. Therefore, the INTPPS register may be written with values from 0x00-0x0F (corresponding to RA0-RB7). Attempting to write 0x10 or higher to the INTPPS register is not supported and will result in undefined behavior.

### REGISTER 16-4: PMD3: PMD CONTROL REGISTER 3

| U-0 | U-0 | U-0 | U-0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| — | — | — | — | PWM4MD | PWM3MD | CCP2MD | CCP1MD |
| 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 | q = Value depends on condition |

| | |
|---|---|
| bit 7-4 | **Unimplemented:** Read as ‘0’ |
| bit 3 | **PWM4MD:** Disable Pulse-Width Modulator PWM4 bit<br>1 = PWM4 module disabled<br>0 = PWM4 module enabled |
| bit 2 | **PWM3MD:** Disable Pulse-Width Modulator PWM3 bit<br>1 = PWM3 module disabled<br>0 = PWM3 module enabled |
| bit 1 | **CCP2MD:** Disable CCP2 bit<br>1 = CCP2 module disabled<br>0 = CCP2 module enabled |
| bit 0 | **CCP1MD:** Disable CCP1 bit<br>1 = CCP1 module disabled<br>0 = CCP1 module enabled |

### REGISTER 17-4: IOCBP: INTERRUPT-ON-CHANGE PORTB POSITIVE EDGE 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 |
|---|---|---|---|---|---|---|---|
| IOCBP7 | IOCBP6 | IOCBP5 | IOCBP4 | IOCBP3 | IOCBP2 | IOCBP1 | IOCBP0 |
| 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 **IOCBP<7:0>:** Interrupt-on-Change PORTB Positive Edge Enable bits
`1` = Interrupt-on-Change enabled on the pin for a positive-going edge. IOCBFx bit and IOCIF flag will be set upon detecting an edge.
`0` = Interrupt-on-Change disabled for the associated pin

### REGISTER 17-5: IOCBN: INTERRUPT-ON-CHANGE PORTB NEGATIVE EDGE 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 |
|---|---|---|---|---|---|---|---|
| IOCBN7 | IOCBN6 | IOCBN5 | IOCBN4 | IOCBN3 | IOCBN2 | IOCBN1 | IOCBN0 |
| 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 **IOCBN<7:0>:** Interrupt-on-Change PORTB Negative Edge Enable bits
`1` = Interrupt-on-Change enabled on the pin for a negative-going edge. IOCBFx bit and IOCIF flag will be set upon detecting an edge.
`0` = Interrupt-on-Change disabled for the associated pin

### REGISTER 17-6: IOCBF: INTERRUPT-ON-CHANGE PORTB FLAG REGISTER

| R/W/HS-0/0 | R/W/HS-0/0 | R/W/HS-0/0 | R/W/HS-0/0 | R/W/HS-0/0 | R/W/HS-0/0 | R/W/HS-0/0 | R/W/HS-0/0 |
|---|---|---|---|---|---|---|---|
| IOCBF7 | IOCBF6 | IOCBF5 | IOCBF4 | IOCBF3 | IOCBF2 | IOCBF1 | IOCBF0 |
| 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 | HS - Bit is set in hardware |

bit 7-0 **IOCBF<7:0>:** Interrupt-on-Change PORTB Flag bits
`1` = An enabled change was detected on the associated pin
Set when IOCBPx = `1` and a rising edge was detected on RBx, or when IOCBNx = `1` and a falling edge was detected on RBx.
`0` = No change was detected, or the user cleared the detected change

## 25.2 Clock Source Selection

The T0CS<2:0> bits of the T0CON1 register are used to select the clock source for Timer0. Register 25-2 displays the clock source selections.

### 25.2.1 INTERNAL CLOCK SOURCE

When the internal clock source is selected, Timer0 operates as a timer and will increment on multiples of the clock source, as determined by the Timer0 prescaler.

### 25.2.2 EXTERNAL CLOCK SOURCE

When an external clock source is selected, Timer0 can operate as either a timer or a counter. Timer0 will increment on multiples of the rising edge of the external clock source, as determined by the Timer0 prescaler.

## 25.3 Programmable Prescaler

A software programmable prescaler is available for exclusive use with Timer0. There are 16 prescaler options for Timer0 ranging in powers of two from 1:1 to 1:32768. The prescaler values are selected using the T0CKPS<3:0> bits of the T0CON1 register.

The prescaler is not directly readable or writable. Clearing the prescaler register can be done by writing to the TMR0L register or the T0CON1 register.

## 25.4 Programmable Postscaler

A software programmable postscaler (output divider) is available for exclusive use with Timer0. There are 16 postscaler options for Timer0 ranging from 1:1 to 1:16. The postscaler values are selected using the T0OUTPS<3:0> bits of the T0CON0 register.

The postscaler is not directly readable or writable. Clearing the postscaler register can be done by writing to the TMR0L register or the T0CON0 register.

## 25.5 Operation during Sleep

When operating synchronously, Timer0 will halt. When operating asynchronously, Timer0 will continue to increment and wake the device from Sleep (if Timer0 interrupts are enabled) provided that the input clock source is active.

## 25.6 Timer0 Interrupts

The Timer0 Interrupt Flag bit (TMR0IF) is set when either of the following conditions occur:

- 8-bit TMR0L matches the TMR0H value
- 16-bit TMR0 rolls over from ‘FFFFh’

When the postscaler bits (T0OUTPS<3:0>) are set to 1:1 operation (no division), the T0IF flag bit will be set with every TMR0 match or rollover. In general, the TMR0IF flag bit will be set every T0OUTPS +1 matches or rollovers.

If Timer0 interrupts are enabled (TMR0IE bit of the PIE0 register = 1), the CPU will be interrupted and the device may wake from sleep (see **Section 25.2 “Clock Source Selection”** for more details).

## 25.7 Timer0 Output

The Timer0 output can be routed to any I/O pin via the RxyPPS output selection register (see **Section 15.0 “Peripheral Pin Select (PPS) Module”** for additional information). The Timer0 output can also be used by other peripherals, such as the Auto-conversion Trigger of the Analog-to-Digital Converter. Finally, the Timer0 output can be monitored through software via the Timer0 output bit (T0OUT) of the T0CON0 register (Register 25-1).

TMR0_out will be one postscaled clock period when a match occurs between TMR0L and TMR0H in 8-bit mode, or when TMR0 rolls over in 16-bit mode. The Timer0 output is a 50% duty cycle that toggles on each TMR0_out rising clock edge.

## 26.7 Timer1 Interrupts

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

- ON bit of the T1CON register
- TMR1IE bit of the PIE4 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:** | To avoid immediate interrupt vectoring, the TMR1H:TMR1L register pair should be preloaded with a value that is not imminently about to rollover, and the TMR1IF flag should be cleared prior to enabling the timer interrupts. |
|---|---|

## 26.8 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:

- ON bit of the T1CON register must be set
- TMR1IE bit of the PIE4 register must be set
- PEIE bit of the INTCON register must be set
- $\overline{\text{SYNC}}$ bit of the T1CON register must be set

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.

## 26.9 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 CCPRxH:CCPRxL register pair on a configured event.

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

For more information, see **Section 29.0 "Capture/Compare/PWM Modules"**.

## 26.10 CCP Special Event Trigger

When any of the CCPs are configured to trigger a special event, the trigger will clear the TMRxH:TMRxL register pair. This special event does not cause a Timer1 interrupt. The CCP module may still be configured to generate a CCP interrupt. In this mode of operation, the CCPRxH:CCPRxL 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 Special Event Trigger. Asynchronous operation of Timer1 can cause a Special Event Trigger to be missed. In the event that a write to TMRxH or TMRxL coincides with a Special Event Trigger from the CCP, the write will take precedence.

### REGISTER 26-2: T1GCON: TIMER1 GATE CONTROL REGISTER

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| R/W-0/u | R/W-0/u | R/W-0/u | R/W-0/u | R/W/HC-0/u | R-x/x | U-0 | U-0 |
| GE | GPOL | GTM | GSPM | GGO/$\overline{\text{DONE}}$ | GVAL | — | — |
| 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 | HC = Bit is cleared by hardware |

bit 7 **GE:** Timer1 Gate Enable bit

If ON = 0:
This bit is ignored
If ON = 1:
1 = Timer1 counting is controlled by the Timer1 gate function
0 = Timer1 is always counting

bit 6 **GPOL:** Timer1 Gate Polarity bit

1 = Timer1 gate is active-high (Timer1 counts when gate is high)
0 = Timer1 gate is active-low (Timer1 counts when gate is low)

bit 5 **GTM:** Timer1 Gate Toggle Mode bit

1 = Timer1 Gate Toggle mode is enabled
0 = Timer1 Gate Toggle mode is disabled and toggle flip-flop is cleared
Timer1 gate flip-flop toggles on every rising edge.

bit 4 **GSPM:** Timer1 Gate Single-Pulse Mode bit

1 = Timer1 Gate Single-Pulse mode is enabled
0 = Timer1 Gate Single-Pulse mode is disabled

bit 3 **GGO/$\overline{\text{DONE}}$:** Timer1 Gate Single-Pulse Acquisition Status bit

1 = Timer1 gate single-pulse acquisition is ready, waiting for an edge
0 = Timer1 gate single-pulse acquisition has completed or has not been started
This bit is automatically cleared when GSPM is cleared

bit 2 **GVAL:** Timer1 Gate Value Status bit

Indicates the current state of the Timer1 gate that could be provided to TMR1H:TMR1L
Unaffected by Timer1 Gate Enable (GE)

bit 1-0 **Unimplemented:** Read as ‘0’

### 27.5.9 EDGE-TRIGGERED MONOSTABLE MODES

The Edge-Triggered Monostable modes start the timer on an edge from the external Reset signal input, after the ON bit is set, and stop incrementing the timer when the timer matches the PRx period value. The following edges will start the timer:

- Rising edge (MODE<4:0> = `10001`)
- Falling edge (MODE<4:0> = `10010`)
- Rising or Falling edge (MODE<4:0> = `10011`)

When an Edge-Triggered Monostable mode is used in conjunction with the CCP PWM operation the PWM drive goes active with the external Reset signal edge that starts the timer, but will not go active when the timer matches the PRx value. While the timer is incrementing, additional edges on the external Reset signal will not affect the CCP PWM.

### REGISTER 28-16: SMTxPRL: SMT PERIOD REGISTER – LOW BYTE

| R/W-x/1 | R/W-x/1 | R/W-x/1 | R/W-x/1 | R/W-x/1 | R/W-x/1 | R/W-x/1 | R/W-x/1 |
|---|---|---|---|---|---|---|---|
| SMTxPR<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 **SMTxPR<7:0>**: Significant bits of the SMT Timer Value for Period Match – Low Byte

### REGISTER 28-17: SMTxPRH: SMT PERIOD REGISTER – HIGH BYTE

| R/W-x/1 | R/W-x/1 | R/W-x/1 | R/W-x/1 | R/W-x/1 | R/W-x/1 | R/W-x/1 | R/W-x/1 |
|---|---|---|---|---|---|---|---|
| SMTxPR<15:8> | | | | | | | |
| 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 **SMTxPR<15:8>**: Significant bits of the SMT Timer Value for Period Match – High Byte

### REGISTER 28-18: SMTxPRU: SMT PERIOD REGISTER – UPPER BYTE

| R/W-x/1 | R/W-x/1 | R/W-x/1 | R/W-x/1 | R/W-x/1 | R/W-x/1 | R/W-x/1 | R/W-x/1 |
|---|---|---|---|---|---|---|---|
| SMTxPR<23:16> | | | | | | | |
| 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 **SMTxPR<23:16>**: Significant bits of the SMT Timer Value for Period Match – Upper Byte

## 29.4 Register Definitions: CCP Control

Long bit name prefixes for the CCP peripherals are shown in **Section 1.1 “Register and Bit Naming Conventions”**.

**TABLE 29-4: LONG BIT NAMES PREFIXES FOR CCP PERIPHERALS**

| Peripheral | Bit Name Prefix |
|---|---|
| CCP1 | CCP1 |
| CCP2 | CCP2 |

**REGISTER 29-1: CCPxCON: CCPx CONTROL REGISTER**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| R/W-0/0 | U-0 | R-x | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
| EN | — | OUT | FMT | MODE<3:0> | | | |
| bit 7 | | | | | | | bit 0 |

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

bit 7 **EN:** CCPx Module Enable bit
1 = CCPx is enabled
0 = CCPx is disabled

bit 6 **Unimplemented:** Read as ‘0’

bit 5 **OUT:** CCPx Output Data bit (read-only)

bit 4 **FMT:** CCPW (Pulse Width) Alignment bit
MODE = Capture mode
Unused
MODE = Compare mode
Unused
MODE = PWM mode
1 = Left-aligned format
0 = Right-aligned format

**TABLE 31-2: SUMMARY OF REGISTERS ASSOCIATED WITH CWG**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| CWG1CLKCON | — | — | — | — | — | — | — | CS | 492 |
| CWG1ISM | — | — | — | — | IS<3:0> | | | | 492 |
| CWG1DBR | — | — | DBR<5:0> | | | | | | 488 |
| CWG1DBF | — | — | DBF<5:0> | | | | | | 488 |
| CWG1CON0 | EN | LD | — | — | — | MODE<2:0> | | | 486 |
| CWG1CON1 | — | — | IN | — | POLD | POLC | POLB | POLA | 487 |
| CWG1AS0 | SHUTDOWN | REN | LSBD<1:0> | | LSAC<1:0> | | — | — | 489 |
| CWG1AS1 | — | — | — | AS4E | AS3E | AS2E | AS1E | AS0E | 490 |
| CWG1STR | OVRD | OVRC | OVRB | OVRA | STRD | STRC | STRB | STRA | 491 |
| INTCON | GIE | PEIE | — | — | — | — | — | INTEDG | 164 |
| PIE7 | — | — | NVMIE | — | — | — | — | CWG1IE | 172 |
| PIR7 | — | — | NVMIF | — | — | — | — | CWG1IF | 181 |

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

### REGISTER 32-8: CLCxGLS1: GATE 1 LOGIC SELECT REGISTER

| R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u |
|---|---|---|---|---|---|---|---|
| LCxG2D4T | LCxG2D4N | LCxG2D3T | LCxG2D3N | LCxG2D2T | LCxG2D2N | LCxG2D1T | LCxG2D1N |
| 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 **LCxG2D4T:** Gate 1 Data 4 True (non-inverted) bit
1 = CLCIN3 (true) is gated into CLCx Gate 1
0 = CLCIN3 (true) is not gated into CLCx Gate 1

bit 6 **LCxG2D4N:** Gate 1 Data 4 Negated (inverted) bit
1 = CLCIN3 (inverted) is gated into CLCx Gate 1
0 = CLCIN3 (inverted) is not gated into CLCx Gate 1

bit 5 **LCxG2D3T:** Gate 1 Data 3 True (non-inverted) bit
1 = CLCIN2 (true) is gated into CLCx Gate 1
0 = CLCIN2 (true) is not gated into CLCx Gate 1

bit 4 **LCxG2D3N:** Gate 1 Data 3 Negated (inverted) bit
1 = CLCIN2 (inverted) is gated into CLCx Gate 1
0 = CLCIN2 (inverted) is not gated into CLCx Gate 1

bit 3 **LCxG2D2T:** Gate 1 Data 2 True (non-inverted) bit
1 = CLCIN1 (true) is gated into CLCx Gate 1
0 = CLCIN1 (true) is not gated into CLCx Gate 1

bit 2 **LCxG2D2N:** Gate 1 Data 2 Negated (inverted) bit
1 = CLCIN1 (inverted) is gated into CLCx Gate 1
0 = CLCIN1 (inverted) is not gated into CLCx Gate 1

bit 1 **LCxG2D1T:** Gate 1 Data 1 True (non-inverted) bit
1 = CLCIN0 (true) is gated into CLCx Gate 1
0 = CLCIN0 (true) is not gated into CLCx Gate1

bit 0 **LCxG2D1N:** Gate 1 Data 1 Negated (inverted) bit
1 = CLCIN0 (inverted) is gated into CLCx Gate 1
0 = CLCIN0 (inverted) is not gated into CLCx Gate 1

### 33.5.3 SLAVE TRANSMISSION

When the R/$\overline{W}$ bit of the incoming address byte is set and an address match occurs, the R/$\overline{W}$ bit of the SSPxSTAT register is set. The received address is loaded into the SSPxBUF register, and an $\overline{ACK}$ pulse is sent by the slave on the ninth bit.

Following the $\overline{ACK}$, slave hardware clears the CKP bit and the SCL pin is held low (see **Section 33.5.6 "Clock Stretching"** for more detail). By stretching the clock, the master will be unable to assert another clock pulse until the slave is done preparing the transmit data.

The transmit data must be loaded into the SSPxBUF register which also loads the SSPxSR register. Then the SCL pin should be released by setting the CKP bit of the SSPxCON1 register. The eight data bits are shifted out on the falling edge of the SCL input. This ensures that the SDA signal is valid during the SCL high time.

The $\overline{ACK}$ pulse from the master-receiver is latched on the rising edge of the ninth SCL input pulse. This $\overline{ACK}$ value is copied to the ACKSTAT bit of the SSPxCON2 register. If ACKSTAT is set (not $\overline{ACK}$), then the data transfer is complete. When the not $\overline{ACK}$ is latched by the slave, the slave goes idle and waits for another occurrence of the Start bit. If the SDA line was low ($\overline{ACK}$), the next transmit data must be loaded into the SSPxBUF register. Again, the SCL pin must be released by setting bit CKP.

An MSSP interrupt is generated for each data transfer byte. The SSPxIF bit must be cleared by software and the SSPxSTAT register is used to determine the status of the byte. The SSPxIF bit is set on the falling edge of the ninth clock pulse.

#### 33.5.3.1 Slave Mode Bus Collision

A slave receives a read request and begins shifting data out on the SDA line. If a bus collision is detected and the SBCDE bit of the SSPxCON3 register is set, the BCL1IF bit of the PIR3 register is set. Once a bus collision is detected, the slave goes idle and waits to be addressed again. User software can use the BCL1IF bit to handle a slave bus collision.

#### 33.5.3.2 7-bit Transmission

A master device can transmit a read request to a slave, and then clock data out of the slave. The list below outlines what software for a slave will need to do to accomplish a standard transmission. Figure 33-18 can be used as a reference to this list.

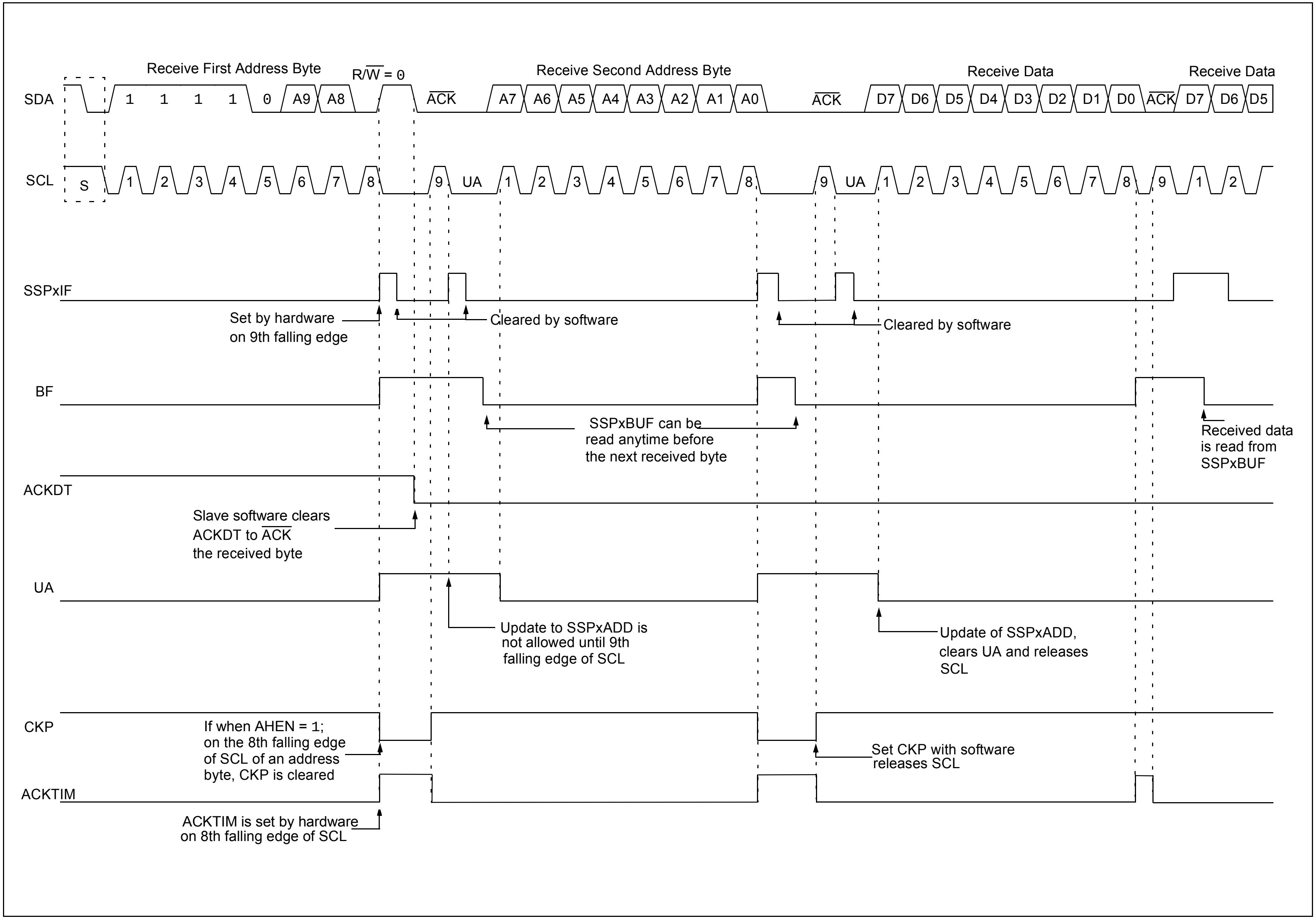
1. Master sends a Start condition on SDA and SCL.
2. S bit of SSPxSTAT is set; SSPxIF is set if interrupt on Start detect is enabled.
3. Matching address with R/$\overline{W}$ bit set is received by the Slave setting SSPxIF bit.
4. Slave hardware generates an $\overline{ACK}$ and sets SSPxIF.
5. SSPxIF bit is cleared by user.
6. Software reads the received address from SSPxBUF, clearing BF.
7. R/$\overline{W}$ is set so CKP was automatically cleared after the $\overline{ACK}$.
8. The slave software loads the transmit data into SSPxBUF.
9. CKP bit is set releasing SCL, allowing the master to clock the data out of the slave.
10. SSPxIF is set after the $\overline{ACK}$ response from the master is loaded into the ACKSTAT register.
11. SSPxIF bit is cleared.
12. The slave software checks the ACKSTAT bit to see if the master wants to clock out more data.

**Note 1:** If the master $\overline{ACK}$s the clock will be stretched.

**2:** ACKSTAT is the only bit updated on the rising edge of SCL (9th) rather than the falling.

13. Steps 9-13 are repeated for each transmitted byte.
14. If the master sends a not $\overline{ACK}$; the clock is not held, but SSPxIF is still set.
15. The master sends a Restart condition or a Stop.
16. The slave is no longer addressed.

**FIGURE 33-21: I²C SLAVE, 10-BIT ADDRESS, RECEPTION (SEN = 0, AHEN = 1, DHEN = 0)**


Receive First Address Byte
R/W = 0
Receive Second Address Byte
Receive Data
Receive Data
SDA
1 1 1 1 0 A9 A8
ACK
A7 A6 A5 A4 A3 A2 A1 A0
ACK
D7 D6 D5 D4 D3 D2 D1 D0 ACK D7 D6 D5
SCL
S
1 2 3 4 5 6 7 8 9 UA 1 2 3 4 5 6 7 8 9 UA 1 2 3 4 5 6 7 8 9 1 2
SSPxIF
Set by hardware on 9th falling edge
Cleared by software
Cleared by software
BF
SSPxBUF can be read anytime before the next received byte
Received data is read from SSPxBUF
ACKDT
Slave software clears ACKDT to ACK the received byte
UA
Update to SSPxADD is not allowed until 9th falling edge of SCL
Update of SSPxADD, clears UA and releases SCL
CKP
If when AHEN = 1; on the 8th falling edge of SCL of an address byte, CKP is cleared
Set CKP with software releases SCL
ACKTIM
ACKTIM is set by hardware on 8th falling edge of SCL

### 33.6.7 I$^2$C MASTER MODE RECEPTION

Master mode reception (Figure 33-29) is enabled by programming the Receive Enable bit, RCEN bit of the SSPxCON2 register.

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

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

#### 33.6.7.1 BF Status Flag

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

#### 33.6.7.2 SSPOV Status Flag

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

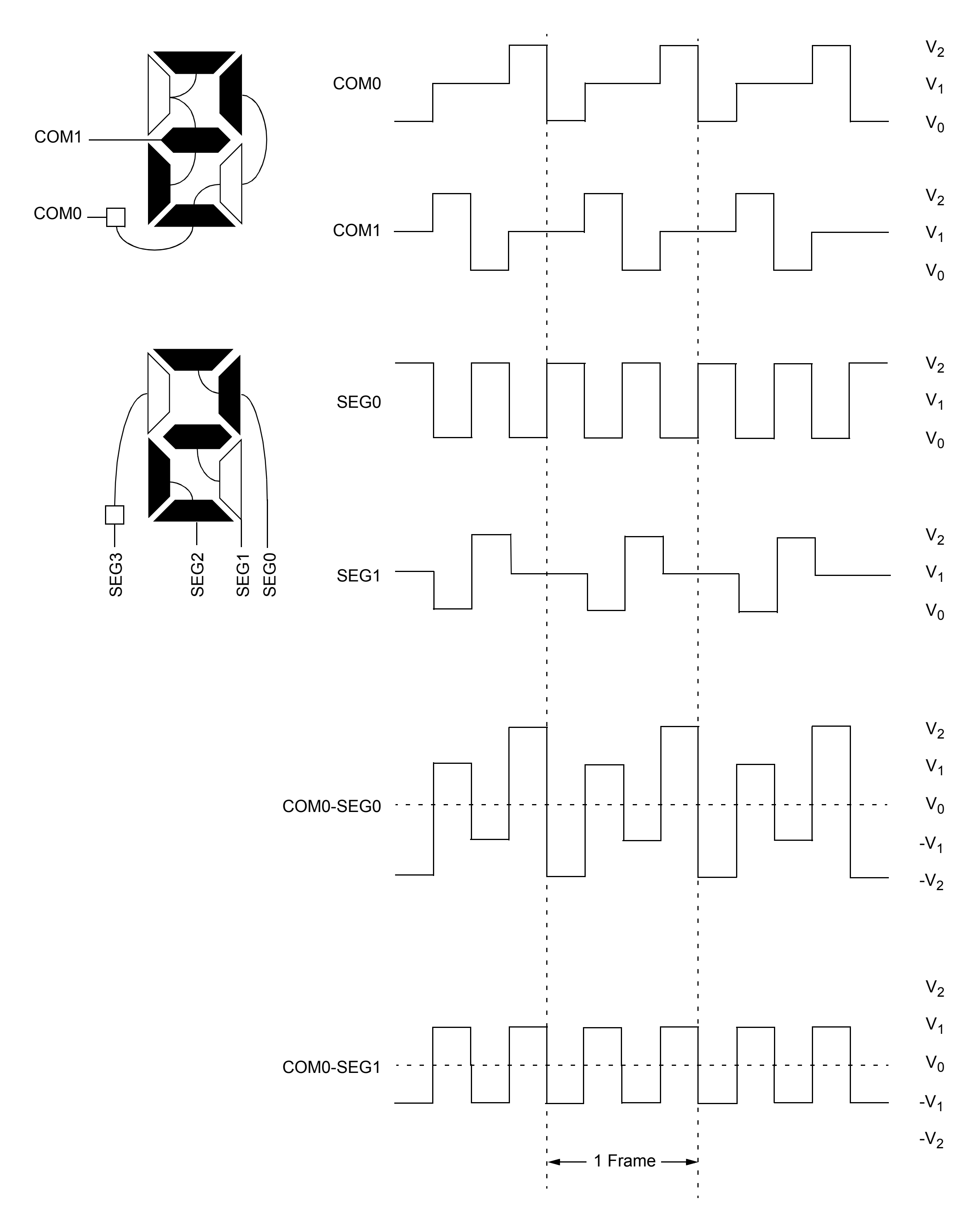
#### 33.6.7.3 WCOL Status Flag

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

#### 33.6.7.4 Typical Receive Sequence:

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

**FIGURE 35-9: TYPE-A WAVEFORMS IN 1/2 MUX, 1/2 BIAS DRIVE**


COM1
COM0
SEG3
SEG2
SEG1
SEG0
COM0
COM1
SEG0
SEG1
COM0-SEG0
COM0-SEG1
V2
V1
V0
-V1
-V2
1 Frame