

Welcome to [E-XFL.COM](https://www.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	I ² C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	25
Program Memory Size	14KB (8K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 24x10b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-VQFN Exposed Pad
Supplier Device Package	28-QFN (6x6)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf18855-i-ml

4.0 DEVICE CONFIGURATION

Device configuration consists of Configuration Words, Code Protection and Device ID.

4.1 Configuration Words

There are several Configuration Word bits that allow different oscillator and memory protection options. These are implemented as shown in Table 4-1.

TABLE 4-1: CONFIGURATION WORD LOCATIONS

Configuration Word	Location
CONFIG1	8007h
CONFIG2	8008h
CONFIG3	8009h
CONFIG4	800Ah
CONFIG5	800Bh

Note: The $\overline{\text{DEBUG}}$ bit in Configuration Words is managed automatically by device development tools including debuggers and programmers. For normal device operation, this bit should be maintained as a '1'.

PIC16(L)F18855/75

4.2 Register Definitions: Configuration Words

REGISTER 4-1: CONFIG1: CONFIGURATION WORD 1: OSCILLATORS

R/P-1	U-1	R/P-1	U-1	U-1	R/P-1
FCMEN	—	CSWEN	—	—	CLKOUTEN
bit 13			bit 8		

U-1	R/P-1	R/P-1	R/P-1	U-1	R/P-1	R/P-1	R/P-1
—	RSTOSC<2:0>			—	FEXTOSC<2:0>		
bit 7				bit 0			

Legend:

R = Readable bit
'0' = Bit is cleared

P = Programmable bit
'1' = Bit is set

U = Unimplemented bit, read as '1'
n = Value when blank or after Bulk Erase

bit 13 **FCMEN:** Fail-Safe Clock Monitor Enable bit

1 = ON FSCM timer enabled
0 = OFF FSCM timer disabled

bit 12 **Unimplemented:** Read as '1'

bit 11 **CSWEN:** Clock Switch Enable bit

1 = ON Writing to NOSC and NDIV is allowed
0 = OFF The NOSC and NDIV bits cannot be changed by user software

bit 10-9 **Unimplemented:** Read as '1'

bit 8 **CLKOUTEN:** Clock Out Enable bit

If FEXTOSC = EC (high, mid or low) or Not Enabled

1 = OFF CLKOUT function is disabled; I/O or oscillator function on OSC2
0 = ON CLKOUT function is enabled; Fosc/4 clock appears at OSC2

Otherwise

This bit is ignored.

bit 7 **Unimplemented:** Read as '1'

bit 6-4 **RSTOSC<2:0>:** Power-up Default Value for COSC bits

This value is the Reset default value for COSC, and selects the oscillator first used by user software

111 = EXT1X EXTOSC operating per FEXTOSC bits

110 = HFINT1 HFINTOSC (1 MHz)

101 = LFINT LFINTOSC

100 = SOSC SOSC

011 = Reserved

010 = EXT4X EXTOSC with 4x PLL, with EXTOSC operating per FEXTOSC bits

001 = HFINTPLL HFINTOSC with 2x PLL, with OSCFRQ = 16 MHz and CDIV = 1:1 (Fosc = 32 MHz)

000 = HFINT32 HFINTOSC with OSCFRQ = 32 MHz and CDIV = 1:1

bit 3 **Unimplemented:** Read as '1'

bit 2-0 **FEXTOSC<2:0>:** FEXTOSC External Oscillator mode Selection bits

111 = ECH EC (External Clock)

110 = ECM EC (External Clock)

101 = ECL EC (External Clock)

100 = OFF External Oscillator is disabled. RA7 is available as a general purpose I/O.

011 = Reserved

010 = HS HS (Crystal oscillator)

001 = XT XT (Crystal oscillator)

000 = LP LP (Crystal oscillator)

PIC16(L)F18855/75

TABLE 8-1: SUMMARY OF REGISTERS ASSOCIATED WITH POWER-DOWN MODE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	134
PIE0	—	—	TMR0IE	IOCFIE	—	—	—	INTE	135
PIE1	OSFIE	CSWIE	—	—	—	—	ADTIE	ADIE	136
PIE2	—	ZCDIE	—	—	—	—	C2IE	C1IE	137
PIE3	—	—	RCIE	TXIE	BCL2IE	SSP2IE	BCL1IE	SSP1IE	138
PIE4	—	—	TMR6IE	TMR5IE	TMR4IE	TMR3IE	TMR2IE	TMR1IE	139
PIR0	—	—	TMR0IF	IOCFIF	—	—	—	INTF	144
PIR1	OSFIF	CSWIF	—	—	—	—	ADTIF	ADIF	145
PIR2	—	ZCDIF	—	—	—	—	C2IF	C1IF	146
PIR3	—	—	RCIF	TXIF	BCL2IF	SSP2IF	BCL1IF	SSP1IF	147
PIR4	—	—	TMR6IF	TMR5IF	TMR4IF	TMR3IF	TMR2IF	TMR1IF	148
IOCAP	IOCAP7	IOCAP6	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0	262
IOCAN	IOCAN7	IOCAN6	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0	262
IOCAF	IOCAF7	IOCAF6	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0	262
IOCCP	IOCCP7	IOCCP6	IOCCP5	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0	264
IOCBP	IOCBP7	IOCBP6	IOCBP5	IOCBP4	IOCBP3	IOCBP2	IOCBP1	IOCBP0	263
IOCBN	IOCBN7	IOCBN6	IOCBN5	IOCBN4	IOCBN3	IOCBN2	IOCBN1	IOCBN0	263
IOCBF	IOCBF7	IOCBF6	IOCBF5	IOCBF4	IOCBF3	IOCBF2	IOCBF1	IOCBF0	263
IOCCN	IOCCN7	IOCCN6	IOCCN5	IOCCN4	IOCCN3	IOCCN2	IOCCN1	IOCCN0	264
IOCCF	IOCCF7	IOCCF6	IOCCF5	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0	264
IOCEP	—	—	—	—	IOCEP3	—	—	—	265
IOCEN	—	—	—	—	IOCEN3	—	—	—	265
IOCEF	—	—	—	—	IOCEF3	—	—	—	266
STATUS	—	—	—	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	38
VREGCON	—	—	—	—	—	—	VREGPM	Reserved	159
CPUDOZE	IDLEN	DOZEN	ROI	DOE	—	DOZE<2:0>			160
WDTCON0	—	—	WDTPS<4:0>					SWDTEN	166
IOCEP	—	—	—	—	IOCEP3	—	—	—	265
IOCEN	—	—	—	—	IOCEN3	—	—	—	265
IOCEF	—	—	—	—	IOCEF3	—	—	—	266

Legend: — = unimplemented location, read as '0'. Shaded cells are not used in Power-Down mode.

11.7 Configuring the CRC

The following steps illustrate how to properly configure the CRC.

1. Determine if the automatic Program Memory scan will be used with the scanner or manual calculation through the SFR interface and perform the actions specified in **Section 11.4 “CRC Data Sources”**, depending on which decision was made.
2. If desired, seed a starting CRC value into the CRCACCH/L registers.
3. Program the CRCXORH/L registers with the desired generator polynomial.
4. Program the DLEN<3:0> bits of the CRCCON1 register with the length of the data word – 1 (refer to Example 11-1). This determines how many times the shifter will shift into the accumulator for each data word.
5. Program the PLEN<3:0> bits of the CRCCON1 register with the length of the polynomial – 2 (refer to Example 11-1).
6. Determine whether shifting in trailing zeros is desired and set the ACCM bit of CRCCON0 register appropriately.
7. Likewise, determine whether the MSb or LSb should be shifted first and write the SHIFTM bit of CRCCON0 register appropriately.
8. Write the CRCGO bit of the CRCCON0 register to begin the shifting process.
- 9a. If manual SFR entry is used, monitor the FULL bit of CRCCON0 register. When FULL = 0, another word of data can be written to the CRCDATH/L registers, keeping in mind that CRCDATH should be written first if the data has >8 bits, as the shifter will begin upon the CRCDATH register being written.
- 9b. If the scanner is used, the scanner will automatically stuff words into the CRCDATH/L registers as needed, as long as the SCANGO bit is set.
- 10a. If using the Flash memory scanner, monitor the SCANIF (or the SCANGO bit) for the scanner to finish pushing information into the CRCDATA registers. After the scanner is completed, monitor the CRCIF (or the BUSY bit) to determine that the CRC has been completed and the check value can be read from the CRCACC registers. If both the interrupt flags are set (or both BUSY and SCANGO bits are cleared), the completed CRC calculation can be read from the CRCACCH/L registers.
- 10b. If manual entry is used, monitor the CRCIF (or BUSY bit) to determine when the CRCACC registers will hold the check value.

11.8 Program Memory Scan Configuration

If desired, the Program Memory Scan module may be used in conjunction with the CRC module to perform a CRC calculation over a range of program memory addresses. In order to set up the Scanner to work with the CRC you need to perform the following steps:

1. Set the EN bit to enable the module. This can be performed at any point preceding the setting of the SCANGO bit, but if it gets disabled, all internal states of the Scanner are reset (registers are unaffected).
2. Choose which memory access mode is to be used (see **Section 11.10 “Scanning Modes”**) and set the MODE bits of the SCANCON0 register appropriately.
3. Based on the memory access mode, set the INTM bits of the SCANCON0 register to the appropriate interrupt mode (see **Section 11.10.5 “Interrupt Interaction”**).
4. Set the SCANLADRL/H and SCANHADRL/H registers with the beginning and ending locations in memory that are to be scanned.
5. Begin the scan by setting the SCANGO bit in the SCANCON0 register. The scanner will wait (CRCGO must be set) for the signal from the CRC that it is ready for the first Flash memory location, then begin loading data into the CRC. It will continue to do so until it either hits the configured end address or an address that is unimplemented on the device, at which point the SCANGO bit will clear, Scanner functions will cease, and the SCANIF interrupt will be triggered. Alternately, the SCANGO bit can be cleared in software if desired.

11.9 Scanner Interrupt

The scanner will trigger an interrupt when the SCANGO bit transitions from ‘1’ to ‘0’. The SCANIF interrupt flag of PIR7 is set when the last memory location is reached and the data is entered into the CRCDATA registers. The SCANIF bit can only be cleared in software. The SCAN interrupt enable is the SCANIE bit of the PIE7 register.

11.10 Scanning Modes

The memory scanner can scan in four modes: Burst, Peek, Concurrent, and Triggered. These modes are controlled by the MODE bits of the SCANCON0 register. The four modes are summarized in Table 11-1.

PIC16(L)F18855/75

REGISTER 12-18: SLRCONB: PORTB SLEW RATE CONTROL REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
SLRB7	SLRB6	SLRB5	SLRB4	SLRB3	SLRB2	SLRB1	SLRB0
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 **SLRB<7:0>**: PORTB Slew Rate Enable bits
For RB<7:0> pins, respectively
1 = Port pin slew rate is limited
0 = Port pin slews at maximum rate

REGISTER 12-19: INLVLB: PORTB INPUT LEVEL CONTROL REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
INLVLB7	INLVLB6	INLVLB5	INLVLB4	INLVLB3	INLVLB2	INLVLB1	INLVLB0
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 **INLVLB<7:0>**: PORTB Input Level Select bits
For RB<7:0> pins, respectively
1 = ST input used for PORT reads and interrupt-on-change
0 = TTL input used for PORT reads and interrupt-on-change

18.3 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 Comparator Specifications in Table 37-14 for more information.

18.4 Timer1 Gate Operation

The output resulting from a comparator operation can be used as a source for gate control of Timer1. See **Section 28.7 “Timer 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.

18.4.1 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 18-2) and the Timer1 Block Diagram (Figure 28-1) for more information.

18.5 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.

18.6 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:

- CxIN0+ analog pin
- DAC output
- FVR (Fixed Voltage Reference)
- Vss (Ground)

See **Section 16.0 “Fixed Voltage Reference (FVR)”** for more information on the Fixed Voltage Reference module.

See **Section 25.0 “5-Bit Digital-to-Analog Converter (DAC1) Module”** for more information on the DAC input signal.

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

18.7 Comparator Negative Input Selection

The CxNCH<2:0> bits of the CMxCON1 register direct an analog input pin and internal reference voltage or analog ground to the inverting input of the comparator:

- CxIN- pin
- FVR (Fixed Voltage Reference)
- Analog Ground

Some inverting input selections share a pin with the operational amplifier output function. Enabling both functions at the same time will direct the operational amplifier output to the comparator inverting input.

Note: To use CxINy+ and CxINy- pins as analog input, the appropriate bits must be set in the ANSEL register and the corresponding TRIS bits must also be set to disable the output drivers.

PIC16(L)F18855/75

19.1.7 CHANGES IN SYSTEM CLOCK FREQUENCY

The PWM frequency is derived from the system clock frequency. Any changes in the system clock frequency will result in changes to the PWM frequency. See **Section 6.0 “Oscillator Module (with Fail-Safe Clock Monitor)”** for additional details.

19.1.8 EFFECTS OF RESET

Any Reset will force all ports to Input mode and the PWMx registers to their Reset states.

TABLE 19-1: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS (Fosc = 20 MHz)

PWM Frequency	1.22 kHz	4.88 kHz	19.53 kHz	78.12 kHz	156.3 kHz	208.3 kHz
Timer Prescale	16	4	1	1	1	1
PR2 Value	0xFF	0xFF	0xFF	0x3F	0x1F	0x17
Maximum Resolution (bits)	10	10	10	8	7	6.6

TABLE 19-2: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS (Fosc = 8 MHz)

PWM Frequency	1.22 kHz	4.90 kHz	19.61 kHz	76.92 kHz	153.85 kHz	200.0 kHz
Timer Prescale	16	4	1	1	1	1
PR2 Value	0xFF	0xFF	0xFF	0x3F	0x1F	0x17
Maximum Resolution (bits)	10	10	10	8	7	6.6

19.1.9 SETUP FOR PWM OPERATION

The following steps should be taken when configuring the module for using the PWMx outputs:

1. Disable the PWMx pin output driver(s) by setting the associated TRIS bit(s).
2. Configure the PWM output polarity by configuring the PWMxPOL bit of the PWMxCON register.
3. Load the PR2 register with the PWM period value, as determined by Equation 19-1.
4. Load the PWMxDCH register and bits <7:6> of the PWMxDCL register with the PWM duty cycle value, as determined by Equation 19-2.
5. Configure and start Timer2:
 - Clear the TMR2IF interrupt flag bit of the PIR1 register.
 - Select the Timer2 prescale value by configuring the T2CKPS<1:0> bits of the T2CON register.
 - Enable Timer2 by setting the TMR2ON bit of the T2CON register.
6. Wait until the TMR2IF is set.
7. When the TMR2IF flag bit is set:
 - Clear the associated TRIS bit(s) to enable the output driver.

- Route the signal to the desired pin by configuring the RxyPPS register.
- Enable the PWMx module by setting the PWMxEN bit of the PWMxCON register.

In order to send a complete duty cycle and period on the first PWM output, the above steps must be followed in the order given. If it is not critical to start with a complete PWM signal, then the PWM module can be enabled during Step 2 by setting the PWMxEN bit of the PWMxCON register.

20.1.2 PUSH-PULL MODE

In Push-Pull mode, two output signals are generated, alternating copies of the input as illustrated in Figure 20-2. This alternation creates the push-pull effect required for driving some transformer-based power supply designs.

The push-pull sequencer is reset whenever $EN = 0$ or if an auto-shutdown event occurs. The sequencer is clocked by the first input pulse, and the first output appears on CWGxA.

The unused outputs CWGxC and CWGxD drive copies of CWGxA and CWGxB, respectively, but with polarity controlled by the POLC and POLD bits of the CWGxCON1 register, respectively.

20.1.3 FULL-BRIDGE MODES

In Forward and Reverse Full-Bridge modes, three outputs drive static values while the fourth is modulated by the input data signal. In Forward Full-Bridge mode, CWGxA is driven to its active state, CWGxB and CWGxC are driven to their inactive state, and CWGxD is modulated by the input signal. In Reverse Full-Bridge mode, CWGxC is driven to its active state, CWGxA and CWGxD are driven to their inactive states, and CWGxB is modulated by the input signal. In Full-Bridge mode, the dead-band period is used when there is a switch from forward to reverse or vice-versa. This dead-band control is described in **Section 20.5 “Dead-Band Control”**, with additional details in **Section 20.6 “Rising Edge and Reverse Dead Band”** and **Section 20.7 “Falling Edge and Forward Dead Band”**.

The mode selection may be toggled between forward and reverse toggling the MODE<0> bit of the CWGxCON0 while keeping MODE<2:1> static, without disabling the CWG module.

PIC16(L)F18855/75

TABLE 22-4: SUMMARY OF REGISTERS ASSOCIATED WITH CLCx

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	134
PIR5	CLC4IF	CLC3IF	CLC2IF	CLC1IF	—	TMR5GIF	TMR3GIF	TMR1GIF	149
PIE5	CLC4IE	CLC4IE	CLC2IE	CLC1IE	—	TMR5GIE	TMR3GIE	TMR1GIE	140
CLC1CON	LC1EN	—	LC1OUT	LC1INTP	LC1INTN	LC1MODE<2:0>			327
CLC1POL	LC1POL	—	—	—	LC1G4POL	LC1G3POL	LC1G2POL	LC1G1POL	328
CLC1SEL0	—	—	LC1D1S<5:0>						329
CLC1SEL1	—	—	LC1D2S<5:0>						329
CLC1SEL2	—	—	LC1D3S<5:0>						329
CLC1SEL3	—	—	LC1D4S<5:0>						329
CLC1GLS0	LC1G1D4T	LC1G1D4N	LC1G1D3T	LC1G1D3N	LC1G1D2T	LC1G1D2N	LC1G1D1T	LC1G1D1N	330
CLC1GLS1	LC1G2D4T	LC1G2D4N	LC1G2D3T	LC1G2D3N	LC1G2D2T	LC1G2D2N	LC1G2D1T	LC1G2D1N	331
CLC1GLS2	LC1G3D4T	LC1G3D4N	LC1G3D3T	LC1G3D3N	LC1G3D2T	LC1G3D2N	LC1G3D1T	LC1G3D1N	332
CLC1GLS3	LC1G4D4T	LC1G4D4N	LC1G4D3T	LC1G4D3N	LC1G4D2T	LC1G4D2N	LC1G4D1T	LC1G4D1N	333
CLC2CON	LC2EN	—	LC2OUT	LC2INTP	LC2INTN	LC2MODE<2:0>			327
CLC2POL	LC2POL	—	—	—	LC2G4POL	LC2G3POL	LC2G2POL	LC2G1POL	328
CLC2SEL0	—	—	LC2D1S<5:0>						329
CLC2SEL1	—	—	LC2D2S<5:0>						329
CLC2SEL2	—	—	LC2D3S<5:0>						329
CLC2SEL3	—	—	LC2D4S<5:0>						329
CLC2GLS0	LC2G1D4T	LC2G1D4N	LC2G1D3T	LC2G1D3N	LC2G1D2T	LC2G1D2N	LC2G1D1T	LC2G1D1N	330
CLC2GLS1	LC2G2D4T	LC2G2D4N	LC2G2D3T	LC2G2D3N	LC2G2D2T	LC2G2D2N	LC2G2D1T	LC2G2D1N	331
CLC2GLS2	LC2G3D4T	LC2G3D4N	LC2G3D3T	LC2G3D3N	LC2G3D2T	LC2G3D2N	LC2G3D1T	LC2G3D1N	332
CLC2GLS3	LC2G4D4T	LC2G4D4N	LC2G4D3T	LC2G4D3N	LC2G4D2T	LC2G4D2N	LC2G4D1T	LC2G4D1N	333
CLC3CON	LC3EN	—	LC3OUT	LC3INTP	LC3INTN	LC3MODE<2:0>			327
CLC3POL	LC3POL	—	—	—	LC3G4POL	LC3G3POL	LC3G2POL	LC3G1POL	328
CLC3SEL0	—	—	LC3D1S<5:0>						329
CLC3SEL1	—	—	LC3D2S<5:0>						329
CLC3SEL2	—	—	LC3D3S<5:0>						329
CLC3SEL3	—	—	LC3D4S<5:0>						329
CLC3GLS0	LC3G1D4T	LC3G1D4N	LC3G1D3T	LC3G1D3N	LC3G1D2T	LC3G1D2N	LC3G1D1T	LC3G1D1N	330
CLC3GLS1	LC3G2D4T	LC3G2D4N	LC3G2D3T	LC3G2D3N	LC3G2D2T	LC3G2D2N	LC3G2D1T	LC3G2D1N	331
CLC3GLS2	LC3G3D4T	LC3G3D4N	LC3G3D3T	LC3G3D3N	LC3G3D2T	LC3G3D2N	LC3G3D1T	LC3G3D1N	332
CLC3GLS3	LC3G4D4T	LC3G4D4N	LC3G4D3T	LC3G4D3N	LC3G4D2T	LC3G4D2N	LC3G4D1T	LC3G4D1N	333
CLC4CON	LC4EN	—	LC4OUT	LC4INTP	LC4INTN	LC4MODE<2:0>			327
CLC4POL	LC4POL	—	—	—	LC4G4POL	LC4G3POL	LC4G2POL	LC4G1POL	328
CLC4SEL0	—	—	LC4D1S<5:0>						329
CLC4SEL1	—	—	LC4D2S<5:0>						329
CLC4SEL2	—	—	LC4D3S<5:0>						329
CLC4SEL3	—	—	LC4D4S<5:0>						329
CLC4GLS0	LC4G1D4T	LC4G1D4N	LC4G1D3T	LC4G1D3N	LC4G1D2T	LC4G1D2N	LC4G1D1T	LC4G1D1N	330

Legend: — = unimplemented, read as '0'. Shaded cells are unused by the CLCx modules.

23.5.8 CONTINUOUS SAMPLING MODE

Setting the ADCONT bit in the ADCON0 register automatically retriggers a new conversion cycle after updating the ADACC register. That means the ADGO bit is set to generate automatic retriggering, until the device Reset occurs or the A/D Stop-on-interrupt bit (ADSOI in the ADCON3 register) is set (correct logic).

23.5.9 DOUBLE SAMPLE CONVERSION

Double sampling is enabled by setting the ADDSEN bit of the ADCON1 register. When this bit is set, two conversions are required before the module will calculate threshold error (each conversion must still be triggered separately). The first conversion will set the ADMATH bit of the ADSTAT register and update ADACC, but will not calculate ADERR or trigger ADTIF. When the second conversion completes, the first value is transferred to ADPREV (depending on the setting of ADPSIS) and the value of the second conversion is placed into ADRES. Only upon the completion of the second conversion is ADERR calculated and ADTIF triggered (depending on the value of ADCALC).

PIC16(L)F18855/75

REGISTER 23-13: ADCNT: ADC CONVERSION COUNTER 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
ADCNT<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

ADCNT<7:0>: ADC Conversion Counter

Counts the number of times that the ADC is triggered. Determines when the threshold is checked for the Low-Pass Filter, Burst Average, and Average Computation modes. Count saturates at 0xFF and does not roll-over to 0x00.

REGISTER 23-14: ADFLTRH: ADC FILTER HIGH BYTE REGISTER

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
ADFLTR<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

ADFLTR<15:8>: ADC Filter Output Most Significant bits and Sign bit

In Accumulate, Average, and Burst Average mode, this is equal to ADACC right shifted by the ADCRS bits of ADCON2. In LPF mode, this is the output of the Lowpass Filter.

REGISTER 23-15: ADFLTRL: ADC FILTER LOW BYTE REGISTER

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
ADFLTR<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

ADFLTR<7:0>: ADC Filter Output Least Significant bits

In Accumulate, Average, and Burst Average mode, this is equal to ADACC right shifted by the ADCRS bits of ADCON2. In LPF mode, this is the output of the Lowpass Filter.

PIC16(L)F18855/75

REGISTER 26-4: MDCARH: MODULATION HIGH CARRIER CONTROL REGISTER

U-0	U-0	U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	—	—	MDCHS<3:0> ⁽¹⁾			
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 3-0 **MDCHS<3:0>** Modulator Data High Carrier Selection bits ⁽¹⁾

1111 = LC4_out
1110 = LC3_out
1101 = LC2_out
1100 = LC1_out
1011 = NCO output
1010 = PWM7_out
1001 = PWM6_out
1000 = CCP5 output (PWM Output mode only)
0111 = CCP4 output (PWM Output mode only)
0110 = CCP3 output (PWM Output mode only)
0101 = CCP2 output (PWM Output mode only)
0100 = CCP1 output (PWM Output mode only)
0011 = Reference clock module signal (CLKR)
0010 = HFINTOSC
0001 = FOSC
0000 = Pin selected by MDCARHPPS

Note 1: Narrowed carrier pulse widths or spurs may occur in the signal stream if the carrier is not synchronized.

PIC16(L)F18855/75

REGISTER 29-2: TxCON: TIMER2/4/6 CONTROL REGISTER

R/W/HC-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
ON ⁽¹⁾	CKPS<2:0>			OUTPS<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 Resets
'1' = Bit is set	'0' = Bit is cleared	HC = Bit is cleared by hardware

bit 7	ON: Timerx On bit 1 = Timerx is on 0 = Timerx is off: all counters and state machines are reset
bit 6-4	CKPS<2:0>: Timer2-type Clock Prescale Select bits 111 = 1:128 Prescaler 110 = 1:64 Prescaler 101 = 1:32 Prescaler 100 = 1:16 Prescaler 011 = 1:8 Prescaler 010 = 1:4 Prescaler 001 = 1:2 Prescaler 000 = 1:1 Prescaler
bit 3-0	OUTPS<3:0>: Timerx Output Postscaler Select bits 1111 = 1:16 Postscaler 1110 = 1:15 Postscaler 1101 = 1:14 Postscaler 1100 = 1:13 Postscaler 1011 = 1:12 Postscaler 1010 = 1:11 Postscaler 1001 = 1:10 Postscaler 1000 = 1:9 Postscaler 0111 = 1:8 Postscaler 0110 = 1:7 Postscaler 0101 = 1:6 Postscaler 0100 = 1:5 Postscaler 0011 = 1:4 Postscaler 0010 = 1:3 Postscaler 0001 = 1:2 Postscaler 0000 = 1:1 Postscaler

Note 1: In certain modes, the ON bit will be auto-cleared by hardware. See **Section 29.5 “Operation Examples”**.

31.5.2 SLAVE RECEPTION

When the $\overline{R/W}$ bit of a matching received address byte is clear, the $\overline{R/W}$ bit of the SSPxSTAT register is cleared. The received address is loaded into the SSPxBUF 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 SSPxSTAT register is set, or bit SSPOV of the SSPxCON1 register is set. The BOEN bit of the SSPxCON3 register modifies this operation. For more information see Register 31-4.

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

When the SEN bit of the SSPxCON2 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 SSPxCON1 register, except sometimes in 10-bit mode. See **Section 31.5.6.2 “10-bit Addressing Mode”** for more detail.

31.5.2.1 7-bit Addressing Reception

This section describes a standard sequence of events for the MSSP module configured as an I²C slave in 7-bit Addressing mode. Figure 31-14 and Figure 31-15 is used as a visual reference for this description.

This is a step by step process of what typically must be done to accomplish I²C communication.

1. Start bit detected.
2. S bit of SSPxSTAT is set; SSPxIF is set if interrupt on Start detect is enabled.
3. Matching address with $\overline{R/W}$ bit clear is received.
4. The slave pulls SDA low sending an \overline{ACK} to the master, and sets SSPxIF bit.
5. Software clears the SSPxIF bit.
6. Software reads received address from SSPxBUF 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 SSPxIF bit.
10. Software clears SSPxIF.
11. Software reads the received byte from SSPxBUF clearing BF.
12. Steps 8-12 are repeated for all received bytes from the master.
13. Master sends Stop condition, setting P bit of SSPxSTAT, and the bus goes idle.

31.5.2.2 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 eighth falling edge of SCL. These additional interrupts allow the slave software to decide whether it wants to 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 I²C communication. Figure 31-16 displays a module using both address and data holding. Figure 31-17 includes the operation with the SEN bit of the SSPxCON2 register set.

1. S bit of SSPxSTAT is set; SSPxIF is set if interrupt on Start detect is enabled.
2. Matching address with $\overline{R/W}$ bit clear is clocked in. SSPxIF is set and CKP cleared after the eighth falling edge of SCL.
3. Slave clears the SSPxIF.
4. Slave can look at the ACKTIM bit of the SSPxCON3 register to determine if the SSPxIF was after or before the \overline{ACK} .
5. Slave reads the address value from SSPxBUF, 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. SSPxIF 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 SSPxIF.

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

11. SSPxIF set and CKP cleared after eighth falling edge of SCL for a received data byte.
12. Slave looks at ACKTIM bit of SSPxCON3 to determine the source of the interrupt.
13. Slave reads the received data from SSPxBUF 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 SSPxSTAT register.

32.0 SIGNAL MEASUREMENT TIMER (SMT)

The SMT is a 24-bit counter with advanced clock and gating logic, which can be configured for measuring a variety of digital signal parameters such as pulse width, frequency and duty cycle, and the time difference between edges on two signals.

Features of the SMT include:

- 24-bit timer/counter
 - Three 8-bit registers (SMTxL/H/U)
 - Readable and writable
 - Optional 16-bit operating mode
- Two 24-bit measurement capture registers
- One 24-bit period match register
- Multi-mode operation, including relative timing measurement
- Interrupt on period match
- Multiple clock, gate and signal sources
- Interrupt on acquisition complete
- Ability to read current input values

Note: These devices implement two SMT modules. All references to SMTx apply to SMT1 and SMT2.

32.6.8 CAPTURE MODE

This mode captures the Timer value based on a rising or falling edge on the SMTWINx input and triggers an interrupt. This mimics the capture feature of a CCP module. The timer begins incrementing upon the SMTxGO bit being set, and updates the value of the SMTxCPR register on each rising edge of SMTWINx, and updates the value of the CPW register on each falling edge of the SMTWINx. The timer is not reset by any hardware conditions in this mode and must be reset by software, if desired. See Figure 32-16 and Figure 32-17.

32.7 Interrupts

The SMT can trigger an interrupt under three different conditions:

- PW Acquisition Complete
- PR Acquisition Complete
- Counter Period Match

The interrupts are controlled by the PIR and PIE registers of the device.

32.7.1 PW AND PR ACQUISITION INTERRUPTS

The SMT can trigger interrupts whenever it updates the SMTxCPW and SMTxCPR registers, the circumstances for which are dependent on the SMT mode, and are discussed in each mode's specific section. The SMTxCPW interrupt is controlled by SMTxPWAIF and SMTxPWAIE bits in registers PIR8 and PIE8, respectively. The SMTxCPR interrupt is controlled by the SMTxPRAIF and SMTxPRAIE bits, also located in registers PIR8 and PIE8, respectively.

In synchronous SMT modes, the interrupt trigger is synchronized to the SMTxCLK. In Asynchronous modes, the interrupt trigger is asynchronous. In either mode, once triggered, the interrupt will be synchronized to the CPU clock.

32.7.2 COUNTER PERIOD MATCH INTERRUPT

As described in **Section 32.1.2 “Period Match interrupt”**, the SMT will also interrupt upon SMTxTMR, matching SMTxPR with its period match limit functionality described in **Section 32.3 “Halt Operation”**. The period match interrupt is controlled by SMTxIF and SMTxIE, located in registers PIR8 and PIE8, respectively.

PIC16(L)F18855/75

REGISTER 32-6: SMTxSIG: SMT1 SIGNAL INPUT SELECT REGISTER

U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	SSEL<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 condition

bit 7-5 **Unimplemented:** Read as '0'

bit 4-0 **SSEL<4:0>:** SMTx Signal Selection bits

11111 = Reserved

•
•
•

11000 = LC4_out

10111 = LC3_out

10110 = LC2_out

10101 = LC1_out

10100 = ZCD1_output

10011 = C2OUT_sync

10010 = C1OUT_sync

10001 = NCO output

10000 = PWM7_out

01111 = PWM6_out

01110 = CCP5_out

01101 = CCP4_out

01100 = CCP3_out

01011 = CCP2_out

01010 = CCP1_out

01001 = SMT2_match⁽¹⁾

01000 = SMT1_match⁽¹⁾

00111 = TMR6_postscaled

00110 = TMR5_postscaled

00101 = TMR4_postscaled

00100 = TMR3_postscaled

00011 = TMR2_postscaled

00010 = TMR1_postscaled

00001 = TMR0_overflow

00000 = SMTxSIGPPS

Note 1: The SMT_match corresponding to the SMT selected becomes reserved.

PIC16(L)F18855/75

33.1.1.5 TSR Status

The TRMT bit of the TX1STA register indicates the status of the TSR register. This is a read-only bit. The TRMT bit is set when the TSR register is empty and is cleared when a character is transferred to the TSR register from the TXREG. The TRMT bit remains clear until all bits have been shifted out of the TSR register. No interrupt logic is tied to this bit, so the user has to poll this bit to determine the TSR status.

Note: The TSR register is not mapped in data memory, so it is not available to the user.

33.1.1.6 Transmitting 9-Bit Characters

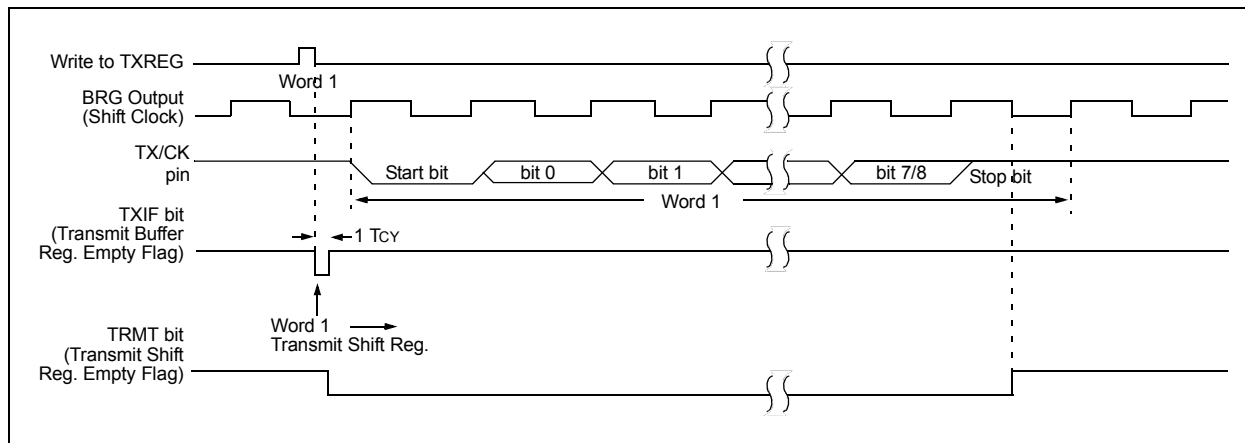
The EUSART supports 9-bit character transmissions. When the TX9 bit of the TX1STA register is set, the EUSART will shift nine bits out for each character transmitted. The TX9D bit of the TX1STA register is the ninth, and Most Significant data bit. When transmitting 9-bit data, the TX9D data bit must be written before writing the eight Least Significant bits into the TXREG. All nine bits of data will be transferred to the TSR shift register immediately after the TXREG is written.

A special 9-bit Address mode is available for use with multiple receivers. See **Section 33.1.2.7 “Address Detection”** for more information on the Address mode.

33.1.1.7 Asynchronous Transmission Set-up:

1. Initialize the SPBRGH, SPBRGL register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see **Section 33.3 “EUSART Baud Rate Generator (BRG)”**).
2. Enable the asynchronous serial port by clearing the SYNC bit and setting the SPEN bit.
3. If 9-bit transmission is desired, set the TX9 control bit. A set ninth data bit will indicate that the eight Least Significant data bits are an address when the receiver is set for address detection.
4. Set SCKP bit if inverted transmit is desired.
5. Enable the transmission by setting the TXEN control bit. This will cause the TXIF interrupt bit to be set.
6. If interrupts are desired, set the TXIE interrupt enable bit of the PIE3 register. An interrupt will occur immediately provided that the GIE and PEIE bits of the INTCON register are also set.
7. If 9-bit transmission is selected, the ninth bit should be loaded into the TX9D data bit.
8. Load 8-bit data into the TXREG register. This will start the transmission.

FIGURE 33-3: ASYNCHRONOUS TRANSMISSION



PIC16(L)F18855/75

FIGURE 37-7: CLKOUT AND I/O TIMING

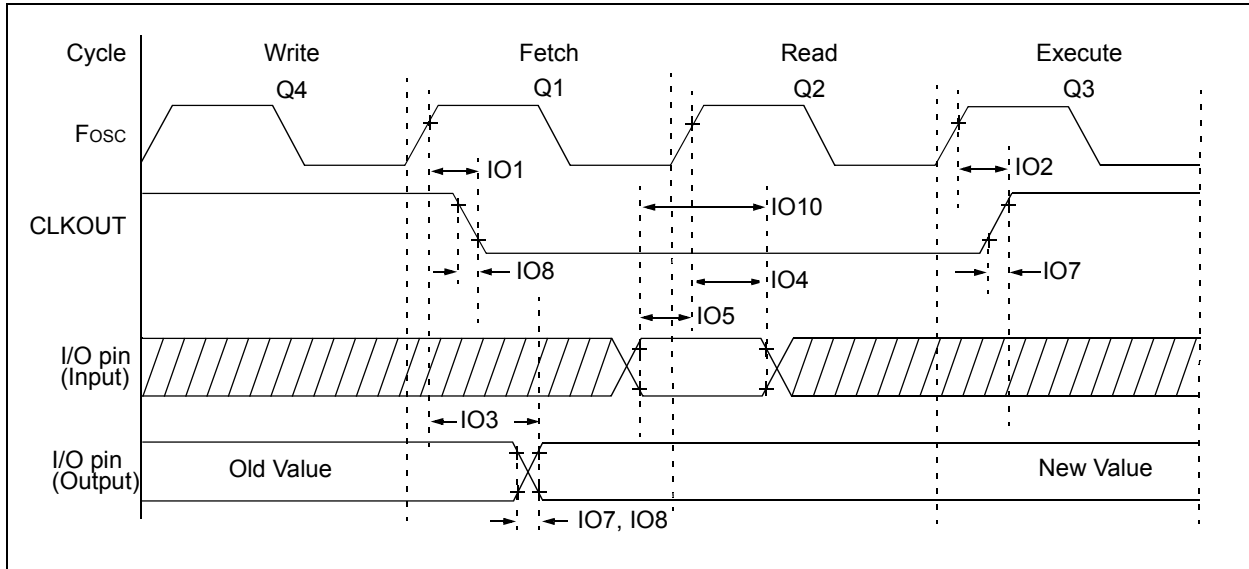


TABLE 37-10: I/O AND CLKOUT TIMING SPECIFICATIONS

Standard Operating Conditions (unless otherwise stated)							
Param. No.	Sym.	Characteristic	Min.	Typ†	Max.	Units	Conditions
IO1*	T _{CLKOUTH}	CLKOUT rising edge delay (rising edge Fosc (Q1 cycle) to falling edge CLKOUT)	—	—	70	ns	
IO2*	T _{CLKOUTL}	CLKOUT falling edge delay (rising edge Fosc (Q3 cycle) to rising edge CLKOUT)	—	—	72	ns	
IO3*	T _{IO_VALID}	Port output valid time (rising edge Fosc (Q1 cycle) to port valid)	—	50	70	ns	
IO4*	T _{IO_SETUP}	Port input setup time (Setup time before rising edge Fosc – Q2 cycle)	20	—	—	ns	
IO5*	T _{IO_HOLD}	Port input hold time (Hold time after rising edge Fosc – Q2 cycle)	50	—	—	ns	
IO6*	T _{IOR_SLREN}	Port I/O rise time, slew rate enabled	—	25	—	ns	V _{DD} = 3.0V
IO7*	T _{IOR_SLRDIS}	Port I/O rise time, slew rate disabled	—	5	—	ns	V _{DD} = 3.0V
IO8*	T _{IOF_SLREN}	Port I/O fall time, slew rate enabled	—	25	—	ns	V _{DD} = 3.0V
IO9*	T _{IOF_SLRDIS}	Port I/O fall time, slew rate disabled	—	5	—	ns	V _{DD} = 3.0V
IO10*	T _{INT}	INT pin high or low time to trigger an interrupt	25	—	—	ns	
IO11*	T _{IOC}	Interrupt-on-Change minimum high or low time to trigger interrupt	25	—	—	ns	

*These parameters are characterized but not tested.