



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	20MHz
Connectivity	I ² C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	25
Program Memory Size	7KB (4K x 14)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	192 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	A/D 11x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f723t-i-so

PIC16(L)F722/3/4/6/7

PIC16(L)F72X Family Types

Device	Data Sheet Index	Program Memory Flash (words)	Data SRAM (bytes)	High-Endurance Flash Memory (bytes)	I/O's ⁽²⁾	8-bit ADC (ch)	CapSense (ch)	Timers (8/16-bit)	AUSART	SSP (I ² C/SPI)	CCP	Debug ⁽¹⁾	XLP
PIC16(L)F707	(1)	8192	363	0	36	14	32	4/2	1	1	2	I	Y
PIC16(L)F720	(2)	2048	128	128	18	12	—	2/1	1	1	1	I	Y
PIC16(L)F721	(2)	4096	256	128	18	12	—	2/1	1	1	1	I	Y
PIC16(L)F722	(4)	2048	128	0	25	11	8	2/1	1	1	2	I	Y
PIC16(L)F722A	(3)	2048	128	0	25	11	8	2/1	1	1	2	I	Y
PIC16(L)F723	(4)	4096	192	0	25	11	8	2/1	1	1	2	I	Y
PIC16(L)F723A	(3)	4096	192	0	25	11	8	2/1	1	1	2	I	Y
PIC16(L)F724	(4)	4096	192	0	36	14	16	2/1	1	1	2	I	Y
PIC16(L)F726	(4)	8192	368	0	25	11	8	2/1	1	1	2	I	Y
PIC16(L)F727	(4)	8192	368	0	36	14	16	2/1	1	1	2	I	Y

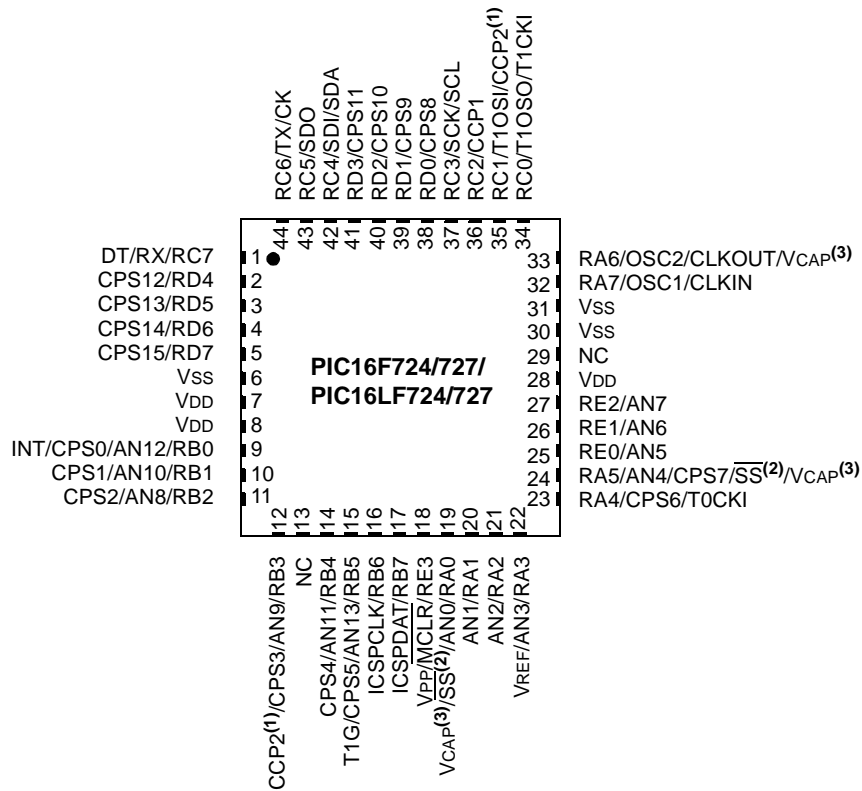
Note 1: I - Debugging, Integrated on Chip; H - Debugging, Requires Debug Header.

2: One pin is input-only.

Data Sheet Index: (Unshaded devices are described in this document.)

- | | | |
|----|---------|---|
| 1: | DS41418 | PIC16(L)F707 Data Sheet, 40/44-Pin Flash, 8-bit Microcontrollers |
| 2: | DS41430 | PIC16(L)F720/721 Data Sheet, 20-Pin Flash, 8-bit Microcontrollers |
| 3: | DS41417 | PIC16(L)F722A/723A Data Sheet, 28-Pin Flash, 8-bit Microcontrollers |
| 4: | DS41341 | PIC16(L)F72X Data Sheet, 28/40/44-Pin Flash, 8-bit Microcontrollers |

Pin Diagrams – 44-PIN QFN (PIC16F724/727/PIC16LF724/727)



- Note** 1: CCP2 pin location may be selected as RB3 or RC1.
2: SS pin location may be selected as RA5 or RA0.
3: PIC16F724/727 devices only.

PIC16(L)F722/3/4/6/7

4.5.3 PIE2 REGISTER

The PIE2 register contains the interrupt enable bits, as shown in Register 4-3.

Note: Bit PEIE of the INTCON register must be set to enable any peripheral interrupt.

REGISTER 4-3: PIE2: PERIPHERAL INTERRUPT ENABLE REGISTER 2

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	CCP2IE
bit 7							bit 0

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

bit 7-1

Unimplemented: Read as '0'

bit 0

CCP2IE: CCP2 Interrupt Enable bit

1 = Enables the CCP2 interrupt

0 = Disables the CCP2 interrupt

6.5.6 RD4/CPS12

Figure 6-21 shows the diagram for these pins. They are configurable to function as one of the following:

- a general purpose I/O
- a capacitive sensing input

6.5.7 RD5/CPS13

Figure 6-21 shows the diagram for these pins. They are configurable to function as one of the following:

- a general purpose I/O
- a capacitive sensing input

6.5.8 RD6/CPS14

Figure 6-21 shows the diagram for these pins. They are configurable to function as one of the following:

- a general purpose I/O
- a capacitive sensing input

6.5.9 RD7/CPS15

Figure 6-21 shows the diagram for these pins. They are configurable to function as one of the following:

- a general purpose I/O
- a capacitive sensing input

FIGURE 6-21: BLOCK DIAGRAM OF RD<7:0>

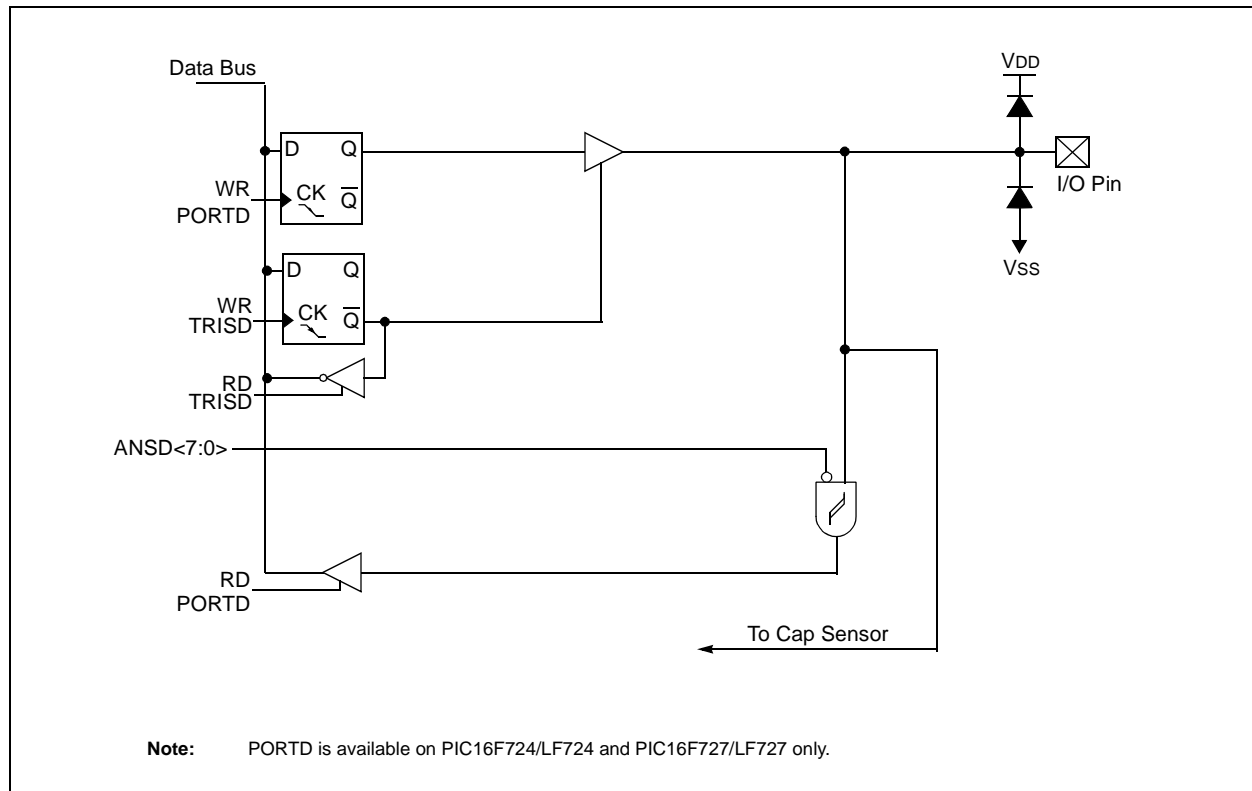


TABLE 6-4: SUMMARY OF REGISTERS ASSOCIATED WITH PORTD⁽¹⁾

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
ANSELD	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	1111 1111	1111 1111
CPSCON0	CPSON	—	—	—	CPSRNG1	CPSRNG0	CPSOUT	T0XCS	0--- 0000	0--- 0000
CPSCON1	—	—	—	—	CPSCH3	CPSCH2	CPSCH1	CPSCH0	---- 0000	---- 0000
PORTD	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	xxxx xxxx	xxxx xxxx
TRISD	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	1111 1111	1111 1111

Legend: x = unknown, u = unchanged, — = unimplemented locations read as '0'. Shaded cells are not used by PORTD.

Note 1: These registers are not implemented on the PIC16F722/723/726/PIC16LF722/723/726 devices, read as '0'.

PIC16(L)F722/3/4/6/7

7.2 Clock Source Modes

Clock source modes can be classified as external or internal.

- Internal clock source (INTOSC) is contained within the oscillator module and derived from a 500 kHz high precision oscillator. The oscillator module has eight selectable output frequencies, with a maximum internal frequency of 16 MHz.
- External clock modes rely on external circuitry for the clock source. Examples are: oscillator modules (EC mode), quartz crystal resonators or ceramic resonators (LP, XT and HS modes) and Resistor-Capacitor (RC) mode circuits.

The system clock can be selected between external or internal clock sources via the FOSC bits of the Configuration Word 1.

7.3 Internal Clock Modes

The oscillator module has eight output frequencies derived from a 500 kHz high precision oscillator. The IRCF bits of the OSCCON register select the postscaler applied to the clock source dividing the frequency by 1, 2, 4 or 8. Setting the PLEN bit of the Configuration Word 1 locks the internal clock source to 16 MHz before the postscaler is selected by the IRCF bits. The PLEN bit must be set or cleared at the time of programming; therefore, only the upper or low four clock source frequencies are selectable in software.

7.3.1 INTOSC AND INTOSCIO MODES

The INTOSC and INTOSCIO modes configure the internal oscillators as the system clock source when the device is programmed using the oscillator selection or the FOSC<2:0> bits in the CONFIG1 register. See **Section 8.0 “Device Configuration”** for more information.

In INTOSC mode, OSC1/CLKIN is available for general purpose I/O. OSC2/CLKOUT outputs the selected internal oscillator frequency divided by 4. The CLKOUT signal may be used to provide a clock for external circuitry, synchronization, calibration, test or other application requirements.

In INTOSCIO mode, OSC1/CLKIN and OSC2/CLKOUT are available for general purpose I/O.

7.3.2 FREQUENCY SELECT BITS (IRCF)

The output of the 500 kHz INTOSC and 16 MHz INTOSC, with Phase-Locked Loop enabled, connect to a postscaler and multiplexer (see Figure 7-1). The Internal Oscillator Frequency Select bits (IRCF) of the OSCCON register select the frequency output of the internal oscillator. Depending upon the PLEN bit, one of four frequencies of two frequency sets can be selected via software:

If PLEN = 1, frequency selection is as follows:

- 16 MHz
- 8 MHz (default after Reset)
- 4 MHz
- 2 MHz

If PLEN = 0, frequency selection is as follows:

- 500 kHz
- 250 kHz (default after Reset)
- 125 kHz
- 62.5 kHz

Note: Following any Reset, the IRCF<1:0> bits of the OSCCON register are set to '10' and the frequency selection is set to 8 MHz or 250 kHz. The user can modify the IRCF bits to select a different frequency.

There is no start-up delay before a new frequency selected in the IRCF bits takes effect. This is because the old and new frequencies are derived from INTOSC via the postscaler and multiplexer.

Start-up delay specifications are located in the Table 23-2 in **Section 23.0 “Electrical Specifications”**.

12.6 Timer1 Gate

Timer1 can be configured to count freely or the count can be enabled and disabled using Timer1 Gate circuitry. This is also referred to as Timer1 Gate Count Enable.

Timer1 Gate can also be driven by multiple selectable sources.

12.6.1 TIMER1 GATE COUNT ENABLE

The Timer1 Gate is enabled by setting the TMR1GE bit of the T1GCON register. The polarity of the Timer1 Gate is configured using the T1GPOL bit of the T1GCON register.

When Timer1 Gate (T1G) input is active, Timer1 will increment on the rising edge of the Timer1 clock source. When Timer1 Gate input is inactive, no incrementing will occur and Timer1 will hold the current count. See Figure 12-3 for timing details.

TABLE 12-3: TIMER1 GATE ENABLE SELECTIONS

T1CLK	T1GPOL	T1G	Timer1 Operation
↑	0	0	Counts
↑	0	1	Holds Count
↑	1	0	Holds Count
↑	1	1	Counts

12.6.2 TIMER1 GATE SOURCE SELECTION

The Timer1 Gate source can be selected from one of four different sources. Source selection is controlled by the T1GSS bits of the T1GCON register. The polarity for each available source is also selectable. Polarity selection is controlled by the T1GPOL bit of the T1GCON register.

TABLE 12-4: TIMER1 GATE SOURCES

T1GSS	Timer1 Gate Source
00	Timer1 Gate Pin
01	Overflow of Timer0 (TMR0 increments from FFh to 00h)
10	Timer2 match PR2 (TMR2 increments to match PR2)
11	Count Enabled by WDT Overflow (Watchdog Time-out interval expired)

12.6.2.1 T1G Pin Gate Operation

The T1G pin is one source for Timer1 Gate Control. It can be used to supply an external source to the Timer1 Gate circuitry.

12.6.2.2 Timer0 Overflow Gate Operation

When Timer0 increments from FFh to 00h, a low-to-high pulse will automatically be generated and internally supplied to the Timer1 Gate circuitry.

12.6.2.3 Timer2 Match Gate Operation

The TMR2 register will increment until it matches the value in the PR2 register. On the very next increment cycle, TMR2 will be reset to 00h. When this Reset occurs, a low-to-high pulse will automatically be generated and internally supplied to the Timer1 Gate circuitry.

12.6.2.4 Watchdog Overflow Gate Operation

The Watchdog Timer oscillator, prescaler and counter will be automatically turned on when TMR1GE = 1 and T1GSS selects the WDT as a gate source for Timer1 (T1GSS = 11). TMR1ON does not factor into the oscillator, prescaler and counter enable. See Table 12-5.

The PSA and PS bits of the OPTION register still control what time-out interval is selected. Changing the prescaler during operation may result in a spurious capture.

Enabling the Watchdog Timer oscillator does not automatically enable a Watchdog Reset or Wake-up from Sleep upon counter overflow.

Note: When using the WDT as a gate source for Timer1, operations that clear the Watchdog Timer (CLRWDT, SLEEP instructions) will affect the time interval being measured for capacitive sensing. This includes waking from Sleep. All other interrupts that might wake the device from Sleep should be disabled to prevent them from disturbing the measurement period.

As the gate signal coming from the WDT counter will generate different pulse widths depending on if the WDT is enabled, when the CLRWDT instruction is executed, and so on, Toggle mode must be used. A specific sequence is required to put the device into the correct state to capture the next WDT counter interval.

PIC16(L)F722/3/4/6/7

REGISTER 14-1: CPSCON0: CAPACITIVE SENSING CONTROL REGISTER 0

R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R-0	R/W-0
CPSON	—	—	—	CPSRNG1	CPSRNG0	CPSOUT	T0XCS
bit 7							bit 0

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

- bit 7 **CPSON:** Capacitive Sensing Module Enable bit
1 = Capacitive sensing module is operating
0 = Capacitive sensing module is shut off and consumes no operating current
- bit 6-4 **Unimplemented:** Read as '0'
- bit 3-2 **CPSRNG<1:0>:** Capacitive Sensing Oscillator Range bits
00 = Oscillator is Off.
01 = Oscillator is in low range. Charge/discharge current is nominally 0.1 μ A.
10 = Oscillator is in medium range. Charge/discharge current is nominally 1.2 μ A.
11 = Oscillator is in high range. Charge/discharge current is nominally 18 μ A.
- bit 1 **CPSOUT:** Capacitive Sensing Oscillator Status bit
1 = Oscillator is sourcing current (Current flowing out the pin)
0 = Oscillator is sinking current (Current flowing into the pin)
- bit 0 **T0XCS:** Timer0 External Clock Source Select bit
If T0CS = 1
The T0XCS bit controls which clock external to the core/Timer0 module supplies Timer0:
1 = Timer0 Clock Source is the capacitive sensing oscillator
0 = Timer0 Clock Source is the T0CKI pin
If T0CS = 0
Timer0 clock source is controlled by the core/Timer0 module and is Fosc/4.

TABLE 15-3: SUMMARY OF REGISTERS ASSOCIATED WITH CAPTURE

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
ANSELB	—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0	--11 1111	--11 1111
APFCON	—	—	—	—	—	—	SSSEL	CCP2SEL	---- --00	---- --00
CCP1CON	—	—	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000
CCP2CON	—	—	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	--00 0000
CCPRxL	Capture/Compare/PWM Register X Low Byte								xxxx xxxx	uuuu uuuu
CCPRxH	Capture/Compare/PWM Register X High Byte								xxxx xxxx	uuuu uuuu
INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000x
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
PIE2	—	—	—	—	—	—	—	CCP2IE	---- --0	---- --0
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
PIR2	—	—	—	—	—	—	—	CCP2IF	---- --0	---- --0
T1CON	TMR1CS1	TMR1CS0	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	—	TMR1ON	0000 00-0	uuuu uu-u
T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/DONE	T1GVAL	T1GSS1	T1GSS0	0000 0x00	0000 0x00
TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111	1111 1111
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	1111 1111	1111 1111

Legend: — = Unimplemented locations, read as '0', u = unchanged, x = unknown. Shaded cells are not used by the Capture.

15.3.2 PWM PERIOD

The PWM period is specified by the PR2 register of Timer2. The PWM period can be calculated using the formula of Equation 15-1.

EQUATION 15-1: PWM PERIOD

$$PWM\ Period = [(PR2) + 1] \cdot 4 \cdot T_{OSC} \cdot (TMR2\ Prescale\ Value)$$

Note: $T_{OSC} = 1/F_{OSC}$

When TMR2 is equal to PR2, the following three events occur on the next increment cycle:

- TMR2 is cleared
- The CCPx pin is set. (Exception: If the PWM duty cycle = 0%, the pin will not be set.)
- The PWM duty cycle is latched from CCPRxL into CCPRxH.

Note: The Timer2 postscaler (refer to **Section 13.1 “Timer2 Operation”**) is not used in the determination of the PWM frequency.

15.3.3 PWM DUTY CYCLE

The PWM duty cycle is specified by writing a 10-bit value to multiple registers: CCPRxL register and DCxB<1:0> bits of the CCPxCON register. The CCPRxL contains the eight MSbs and the DCxB<1:0> bits of the CCPxCON register contain the two LSbs. CCPRxL and DCxB<1:0> bits of the CCPxCON register can be written to at any time. The duty cycle value is not latched into CCPRxH until after the period completes (i.e., a match between PR2 and TMR2 registers occurs). While using the PWM, the CCPRxH register is read-only.

Equation 15-2 is used to calculate the PWM pulse width.

Equation 15-3 is used to calculate the PWM duty cycle ratio.

EQUATION 15-2: PULSE WIDTH

$$Pulse\ Width = (CCPRxL:CCPxCON<5:4>) \cdot T_{OSC} \cdot (TMR2\ Prescale\ Value)$$

Note: $T_{OSC} = 1/F_{OSC}$

EQUATION 15-3: DUTY CYCLE RATIO

$$Duty\ Cycle\ Ratio = \frac{(CCPRxL:CCPxCON<5:4>)}{4(PR2 + 1)}$$

The CCPRxH register and a 2-bit internal latch are used to double buffer the PWM duty cycle. This double buffering is essential for glitchless PWM operation.

The 8-bit timer TMR2 register is concatenated with either the 2-bit internal system clock (F_{OSC}), or two bits of the prescaler, to create the 10-bit time base. The system clock is used if the Timer2 prescaler is set to 1:1.

When the 10-bit time base matches the CCPRxH and 2-bit latch, then the CCPx pin is cleared (refer to Figure 15-3).

16.3.2.3 AUSART Synchronous Slave Reception

The operation of the Synchronous Master and Slave modes is identical (**Section 16.3.1.4 “Synchronous Master Reception”**), with the following exceptions:

- Sleep
- CREN bit is always set, therefore the receiver is never Idle
- SREN bit, which is a “don’t care” in Slave mode

A character may be received while in Sleep mode by setting the CREN bit prior to entering Sleep. Once the word is received, the RSR register will transfer the data to the RCREG register. If the RCIE interrupt enable bit of the PIE1 register is set, the interrupt generated will wake the device from Sleep and execute the next instruction. If the GIE bit is also set, the program will branch to the interrupt vector.

16.3.2.4 Synchronous Slave Reception Setup:

1. Set the SYNC and SPEN bits and clear the CSRC bit.
2. If interrupts are desired, set the RCIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
3. If 9-bit reception is desired, set the RX9 bit.
4. Verify address detection is disabled by clearing the ADDEN bit of the RCSTA register.
5. Set the CREN bit to enable reception.
6. The RCIF bit of the PIR1 register will be set when reception is complete. An interrupt will be generated if the RCIE bit of the PIE1 register was set.
7. If 9-bit mode is enabled, retrieve the Most Significant bit from the RX9D bit of the RCSTA register.
8. Retrieve the eight Least Significant bits from the receive FIFO by reading the RCREG register.
9. If an overrun error occurs, clear the error by either clearing the CREN bit of the RCSTA register.

TABLE 16-9: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE RECEPTION

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
INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000x
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
RCREG	AUSART Receive Data Register								0000 0000	0000 0000
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	1111 1111	1111 1111
TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010

Legend: x = unknown, – = unimplemented read as ‘0’. Shaded cells are not used for Synchronous Slave Reception.

17.1.2 SLAVE MODE

For any SPI device acting as a slave, the data is transmitted and received as external clock pulses appear on SCK pin. This external clock must meet the minimum high and low times as specified in the electrical specifications.

17.1.2.1 Slave Mode Operation

The SSP consists of a transmit/receive shift register (SSPSR) and a buffer register (SSPBUF). The SSPSR shifts the data in and out of the device, MSb first. The SSPBUF holds the data that was written to the SSPSR until the received data is ready.

The slave has no control as to when data will be clocked in or out of the device. All data that is to be transmitted, to a master or another slave, must be loaded into the SSPBUF register before the first clock pulse is received.

Once eight bits of data have been received:

- Received byte is moved to the SSPBUF register
- BF bit of the SSPSTAT register is set
- SSPIF bit of the PIR1 register is set

Any write to the SSPBUF register during transmission/reception of data will be ignored and the Write Collision Detect bit, WCOL of the SSPCON register, will be set. User software must clear the WCOL bit so that it can be determined if the following write(s) to the SSPBUF register completed successfully.

The user's firmware must read SSPBUF, clearing the BF flag, or the SSPOV bit of the SSPCON register will be set with the reception of the next byte and communication will be disabled.

A SPI module transmits and receives at the same time, occasionally causing dummy data to be transmitted/received. It is up to the user to determine which data is to be used and what can be discarded.

17.1.2.2 Enabling Slave I/O

To enable the serial port, the SSPEN bit of the SSPCON register must be set. If a Slave mode of operation is selected in the SSPM bits of the SSPCON register, the SDI, SDO, SCK pins will be assigned as serial port pins.

For these pins to function as serial port pins, they must have their corresponding data direction bits set or cleared in the associated TRIS register as follows:

- SDI configured as input
- SDO configured as output
- SCK configured as input

Optionally, a fourth pin, Slave Select (\overline{SS}) may be used in Slave mode. Slave Select may be configured to operate on one of the following pins via the SSSEL bit in the APFCON register.

- RA5/AN4/ \overline{SS}
- RA0/AN0/ \overline{SS}

Upon selection of a Slave Select pin, the appropriate bits must be set in the ANSELA and TRISA registers. Slave Select must be set as an input by setting the corresponding bit in TRISA, and digital I/O must be enabled on the \overline{SS} pin by clearing the corresponding bit of the ANSELA register.

17.1.2.3 Slave Mode Setup

When initializing the SSP module to SPI Slave mode, compatibility must be ensured with the master device. This is done by programming the appropriate control bits of the SSPCON and SSPSTAT registers. These control bits allow the following to be specified:

- SCK as clock input
- Idle state of SCK (CKP bit)
- Data input sample phase (SMP bit)
- Output data on rising/falling edge of SCK (CKE bit)

Figure 17-4 and Figure 17-5 show example waveforms of Slave mode operation.

17.1.2.4 Slave Select Operation

The \overline{SS} pin allows Synchronous Slave mode operation. The SPI must be in Slave mode with \overline{SS} pin control enabled ($SSPM<3:0> = 0100$). The associated TRIS bit for the \overline{SS} pin must be set, making \overline{SS} an input.

In Slave Select mode, when:

- $\overline{SS} = 0$, The device operates as specified in **Section 17.1.2 “Slave Mode”**.
- $\overline{SS} = 1$, The SPI module is held in Reset and the SDO pin will be tri-stated.

Note 1: When the SPI is in Slave mode with \overline{SS} pin control enabled ($SSPM<3:0> = 0100$), the SPI module will reset if the \overline{SS} pin is driven high.

2: If the SPI is used in Slave mode with CKE set, the \overline{SS} pin control must be enabled.

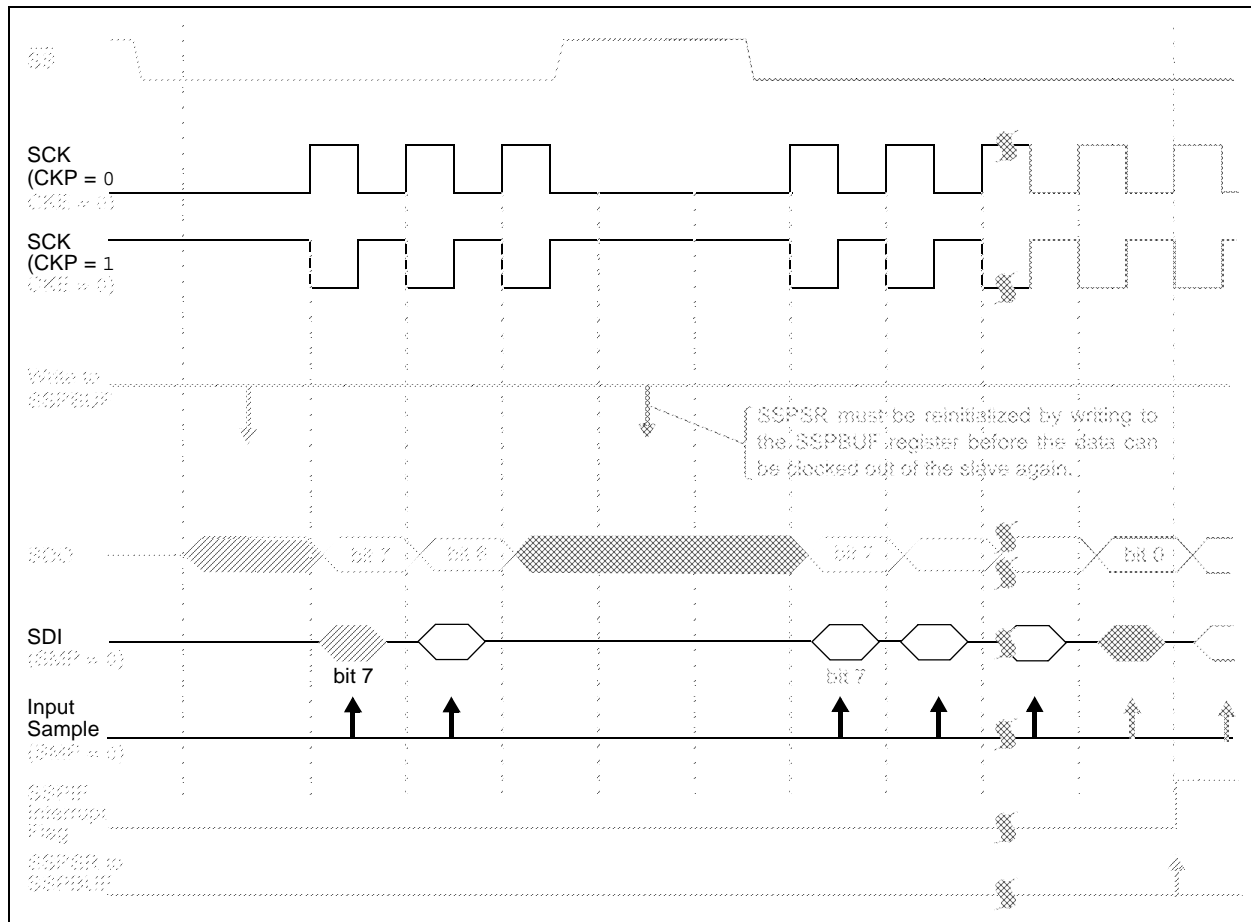
When the SPI module resets, the bit counter is cleared to '0'. This can be done by either forcing the \overline{SS} pin to a high level or clearing the SSPEN bit. Figure 17-6 shows the timing waveform for such a synchronization event.

Note: SSPSR must be reinitialized by writing to the SSPBUF register before the data can be clocked out of the slave again.

17.1.2.5 Sleep in Slave Mode

While in Sleep mode, the slave can transmit/receive data. The SPI Transmit/Receive Shift register operates asynchronously to the device on the externally supplied clock source. This allows the device to be placed in Sleep mode and data to be shifted into the SPI Transmit/Receive Shift register. When all eight bits have been received, the SSP Interrupt Flag bit will be set and if enabled, will wake the device from Sleep.

FIGURE 17-6: SLAVE SELECT SYNCHRONIZATION WAVEFORM



21.0 INSTRUCTION SET SUMMARY

The PIC16(L)F722/3/4/6/7 instruction set is highly orthogonal and is comprised of three basic categories:

- **Byte-oriented** operations
- **Bit-oriented** operations
- **Literal and control** operations

Each PIC16 instruction is a 14-bit word divided into an **opcode**, which specifies the instruction type and one or more **operands**, which further specify the operation of the instruction. The formats for each of the categories is presented in Figure 21-1, while the various opcode fields are summarized in Table 21-1.

Table 21-2 lists the instructions recognized by the MPASM™ assembler.

For **byte-oriented** instructions, 'f' represents a file register designator and 'd' represents a destination designator. The file register designator specifies which file register is to be used by the instruction.

The destination designator specifies where the result of the operation is to be placed. If 'd' is zero, the result is placed in the W register. If 'd' is one, the result is placed in the file register specified in the instruction.

For **bit-oriented** instructions, 'b' represents a bit field designator, which selects the bit affected by the operation, while 'f' represents the address of the file in which the bit is located.

For **literal and control** operations, 'k' represents an 8-bit or 11-bit constant, or literal value.

One instruction cycle consists of four oscillator periods; for an oscillator frequency of 4 MHz, this gives a nominal instruction execution time of 1 μs. All instructions are executed within a single instruction cycle, unless a conditional test is true, or the program counter is changed as a result of an instruction. When this occurs, the execution takes two instruction cycles, with the second cycle executed as a NOP.

All instruction examples use the format '0xhh' to represent a hexadecimal number, where 'h' signifies a hexadecimal digit.

21.1 Read-Modify-Write Operations

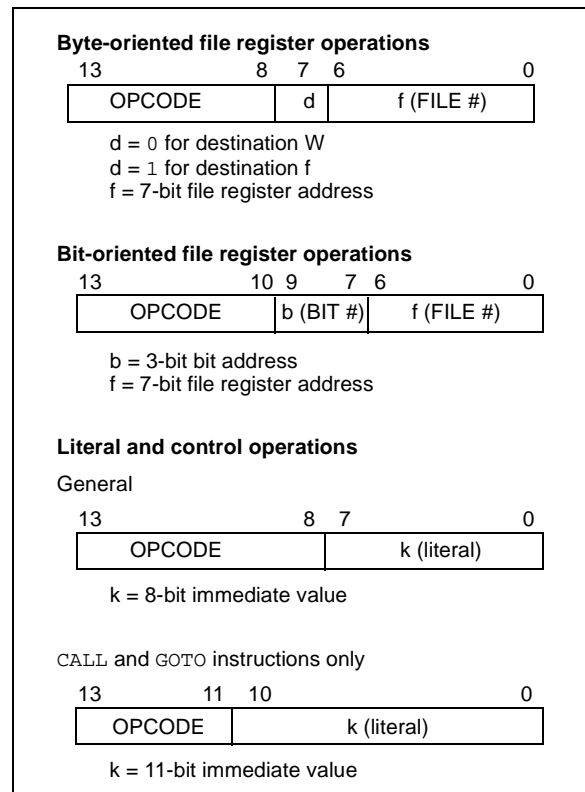
Any instruction that specifies a file register as part of the instruction performs a Read-Modify-Write (R-M-W) operation. The register is read, the data is modified, and the result is stored according to either the instruction, or the destination designator 'd'. A read operation is performed on a register even if the instruction writes to that register.

For example, a `CLRF PORTB` instruction will read PORTB, clear all the data bits, then write the result back to PORTB. This example would have the unintended consequence of clearing the condition that set the RBIF flag.

TABLE 21-1: OPCODE FIELD DESCRIPTIONS

Field	Description
f	Register file address (0x00 to 0x7F)
W	Working register (accumulator)
b	Bit address within an 8-bit file register
k	Literal field, constant data or label
x	Don't care location (= 0 or 1). The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools.
d	Destination select; d = 0: store result in W, d = 1: store result in file register f. Default is d = 1.
PC	Program Counter
TO	Time-out bit
C	Carry bit
DC	Digit carry bit
Z	Zero bit
PD	Power-down bit

FIGURE 21-1: GENERAL FORMAT FOR INSTRUCTIONS



21.2 Instruction Descriptions

ADDLW Add literal and W

Syntax: [*label*] ADDLW *k*

Operands: $0 \leq k \leq 255$

Operation: $(W) + k \rightarrow (W)$

Status Affected: C, DC, Z

Description: The contents of the W register are added to the 8-bit literal 'k' and the result is placed in the W register.

BCF Bit Clear f

Syntax: [*label*] BCF *f*,*b*

Operands: $0 \leq f \leq 127$
 $0 \leq b \leq 7$

Operation: $0 \rightarrow (f)$

Status Affected: None

Description: Bit 'b' in register 'f' is cleared.

ADDWF Add W and f

Syntax: [*label*] ADDWF *f*,*d*

Operands: $0 \leq f \leq 127$
 $d \in [0,1]$

Operation: $(W) + (f) \rightarrow (\text{destination})$

Status Affected: C, DC, Z

Description: Add the contents of the W register with register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.

BSF Bit Set f

Syntax: [*label*] BSF *f*,*b*

Operands: $0 \leq f \leq 127$
 $0 \leq b \leq 7$

Operation: $1 \rightarrow (f)$

Status Affected: None

Description: Bit 'b' in register 'f' is set.

ANDLW AND literal with W

Syntax: [*label*] ANDLW *k*

Operands: $0 \leq k \leq 255$

Operation: $(W) .\text{AND.} (k) \rightarrow (W)$

Status Affected: Z

Description: The contents of W register are AND'ed with the 8-bit literal 'k'. The result is placed in the W register.

BTFSC Bit Test f, Skip if Clear

Syntax: [*label*] BTFSC *f*,*b*

Operands: $0 \leq f \leq 127$
 $0 \leq b \leq 7$

Operation: skip if $(f) = 0$

Status Affected: None

Description: If bit 'b' in register 'f' is '1', the next instruction is executed.
If bit 'b', in register 'f', is '0', the next instruction is discarded, and a NOP is executed instead, making this a 2-cycle instruction.

ANDWF AND W with f

Syntax: [*label*] ANDWF *f*,*d*

Operands: $0 \leq f \leq 127$
 $d \in [0,1]$

Operation: $(W) .\text{AND.} (f) \rightarrow (\text{destination})$

Status Affected: Z

Description: AND the W register with register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.

22.2 MPLAB XC Compilers

The MPLAB XC Compilers are complete ANSI C compilers for all of Microchip's 8, 16, and 32-bit MCU and DSC devices. These compilers provide powerful integration capabilities, superior code optimization and ease of use. MPLAB XC Compilers run on Windows, Linux or MAC OS X.

For easy source level debugging, the compilers provide debug information that is optimized to the MPLAB X IDE.

The free MPLAB XC Compiler editions support all devices and commands, with no time or memory restrictions, and offer sufficient code optimization for most applications.

MPLAB XC Compilers include an assembler, linker and utilities. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. MPLAB XC Compiler uses the assembler to produce its object file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

22.3 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for PIC10/12/16/18 MCUs.

The MPASM Assembler generates relocatable object files for the MPLINK Object Linker, Intel® standard HEX files, MAP files to detail memory usage and symbol reference, absolute LST files that contain source lines and generated machine code, and COFF files for debugging.

The MPASM Assembler features include:

- Integration into MPLAB X IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multipurpose source files
- Directives that allow complete control over the assembly process

22.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler. It can link relocatable objects from precompiled libraries, using directives from a linker script.

The MPLIB Object Librarian manages the creation and modification of library files of precompiled code. When a routine from a library is called from a source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications.

The object linker/librarian features include:

- Efficient linking of single libraries instead of many smaller files
- Enhanced code maintainability by grouping related modules together
- Flexible creation of libraries with easy module listing, replacement, deletion and extraction

22.5 MPLAB Assembler, Linker and Librarian for Various Device Families

MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC DSC devices. MPLAB XC Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

23.2 DC Characteristics: PIC16(L)F722/3/4/6/7-I/E (Industrial, Extended)

PIC16LF722/3/4/6/7			Standard Operating Conditions (unless otherwise stated)				
			Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended				
PIC16F722/3/4/6/7			Standard Operating Conditions (unless otherwise stated)				
			Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended				
Param No.	Device Characteristics	Min.	Typ†	Max.	Units	Conditions	
						VDD	Note
	Supply Current (IDD) ^(1, 2)						
D009	LDO Regulator	—	350	—	μA	—	HS, EC OR INTOSC/INTOSCIO (8-16 MHz) Clock modes with all VCAP pins disabled
		—	50	—	μA	—	All VCAP pins disabled
		—	30	—	μA	—	VCAP enabled on RA0, RA5 or RA6
		—	5	—	μA	—	LP Clock mode and Sleep (requires FVR and BOR to be disabled)
D010		—	7.0	12	μA	1.8	FOSC = 32 kHz
		—	9.0	14	μA	3.0	LP Oscillator mode (Note 4), -40°C ≤ TA ≤ +85°C
D010		—	11	20	μA	1.8	FOSC = 32 kHz
		—	14	22	μA	3.0	LP Oscillator mode (Note 4), -40°C ≤ TA ≤ +85°C
		—	15	24	μA	5.0	
D011		—	7.0	12	μA	1.8	FOSC = 32 kHz
		—	9.0	18	μA	3.0	LP Oscillator mode -40°C ≤ TA ≤ +125°C
D011		—	11	21	μA	1.8	FOSC = 32 kHz
		—	14	25	μA	3.0	LP Oscillator mode (Note 4) -40°C ≤ TA ≤ +125°C
		—	15	27	μA	5.0	
D011		—	110	150	μA	1.8	FOSC = 1 MHz
		—	150	215	μA	3.0	XT Oscillator mode
D011		—	120	175	μA	1.8	FOSC = 1 MHz
		—	180	250	μA	3.0	XT Oscillator mode (Note 5)
		—	240	300	μA	5.0	
D012		—	230	300	μA	1.8	FOSC = 4 MHz
		—	400	600	μA	3.0	XT Oscillator mode
D012		—	250	350	μA	1.8	FOSC = 4 MHz
		—	420	650	μA	3.0	XT Oscillator mode (Note 5)
		—	500	750	μA	5.0	
D013		—	125	180	μA	1.8	FOSC = 1 MHz
		—	230	270	μA	3.0	EC Oscillator mode
D013		—	150	205	μA	1.8	FOSC = 1 MHz
		—	225	320	μA	3.0	EC Oscillator mode (Note 5)
		—	250	410	μA	5.0	

- Note 1:** The test conditions for all IDD measurements in active operation mode are: OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD; MCLR = VDD; WDT disabled.
- Note 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.
- Note 3:** For RC oscillator configurations, current through REXT is not included. The current through the resistor can be extended by the formula $I_R = V_{DD}/2R_{EXT}$ (mA) with REXT in kΩ.
- Note 4:** FVR and BOR are disabled.
- Note 5:** 0.1 μF capacitor on VCAP (RA0).

PIC16(L)F722/3/4/6/7

TABLE 23-2: OSCILLATOR PARAMETERS

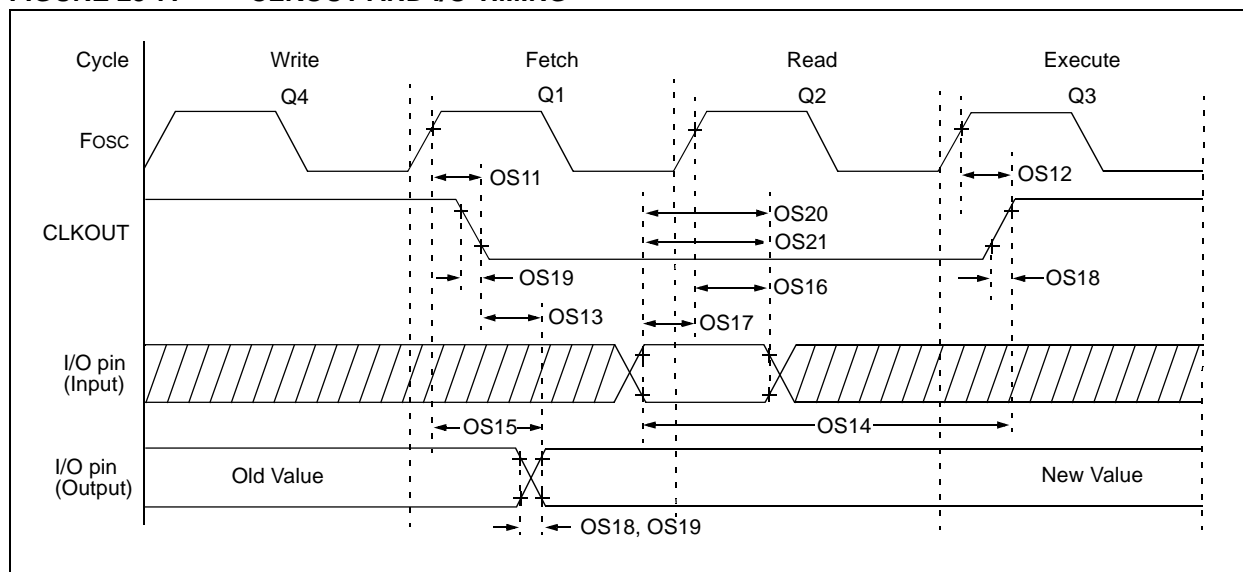
Standard Operating Conditions (unless otherwise stated)								
Operating Temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$								
Param No.	Sym.	Characteristic	Freq. Tolerance	Min.	Typ†	Max.	Units	Conditions
OS08	HFosc	Internal Calibrated HFINTOSC Frequency ⁽²⁾	$\pm 2\%$	—	16.0	—	MHz	$0^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$, $V_{DD} \geq 2.5\text{V}$
			$\pm 5\%$	—	16.0	—	MHz	$-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$
OS08A	MFosc	Internal Calibrated MFINTOSC Frequency ⁽²⁾	$\pm 2\%$	—	500	—	kHz	$0^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$, $V_{DD} \geq 2.5\text{V}$
			$\pm 5\%$	—	500	10	kHz	$-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$
OS10*	Tiosc st	HFINTOSC Wake-up from Sleep Start-up Time	—	—	5	8	μs	
		MFINTOSC Wake-up from Sleep Start-up Time	—	—	20	30	μs	

* 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 1:** Instruction cycle period (Tcy) equals four times the input oscillator time base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "min" values with an external clock applied to the OSC1 pin. When an external clock input is used, the "max" cycle time limit is "DC" (no clock) for all devices.
- 2:** To ensure these oscillator frequency tolerances, VDD and VSS must be capacitively decoupled as close to the device as possible. 0.1 μF and 0.01 μF values in parallel are recommended.
- 3:** By design.

FIGURE 23-7: CLKOUT AND I/O TIMING



PIC16(L)F722/3/4/6/7

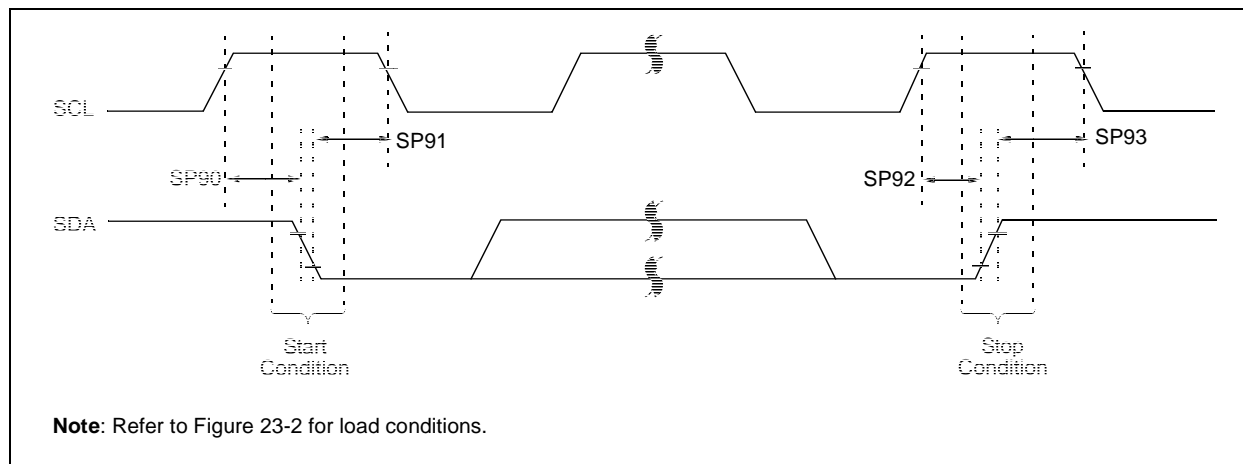
TABLE 23-11: SPI MODE REQUIREMENTS

Param No.	Symbol	Characteristic		Min.	Typ†	Max.	Units	Conditions
SP70*	TssL2sch, TssL2scl	$\overline{SS}\downarrow$ to SCK \downarrow or SCK \uparrow input		Tcy	—	—	ns	
SP71*	Tsch	SCK input high time (Slave mode)		Tcy + 20	—	—	ns	
SP72*	Tscl	SCK input low time (Slave mode)		Tcy + 20	—	—	ns	
SP73*	TdIV2sch, TdIV2scl	Setup time of SDI data input to SCK edge		100	—	—	ns	
SP74*	Tsch2diL, TscL2diL	Hold time of SDI data input to SCK edge		100	—	—	ns	
SP75*	TdoR	SDO data output rise time	3.0-5.5V	—	10	25	ns	
			1.8-5.5V	—	25	50	ns	
SP76*	TdoF	SDO data output fall time		—	10	25	ns	
SP77*	TssH2doZ	$\overline{SS}\uparrow$ to SDO output high-impedance		10	—	50	ns	
SP78*	Tscr	SCK output rise time (Master mode)	3.0-5.5V	—	10	25	ns	
			1.8-5.5V	—	25	50	ns	
SP79*	TscF	SCK output fall time (Master mode)		—	10	25	ns	
SP80*	Tsch2doV, TscL2doV	SDO data output valid after SCK edge	3.0-5.5V	—	—	50	ns	
			1.8-5.5V	—	—	145	ns	
SP81*	TdoV2sch, TdoV2scl	SDO data output setup to SCK edge		Tcy	—	—	ns	
SP82*	TssL2doV	SDO data output valid after $\overline{SS}\downarrow$ edge		—	—	50	ns	
SP83*	Tsch2ssH, TscL2ssH	$\overline{SS}\uparrow$ after SCK edge		1.5Tcy + 40	—	—	ns	

* 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.

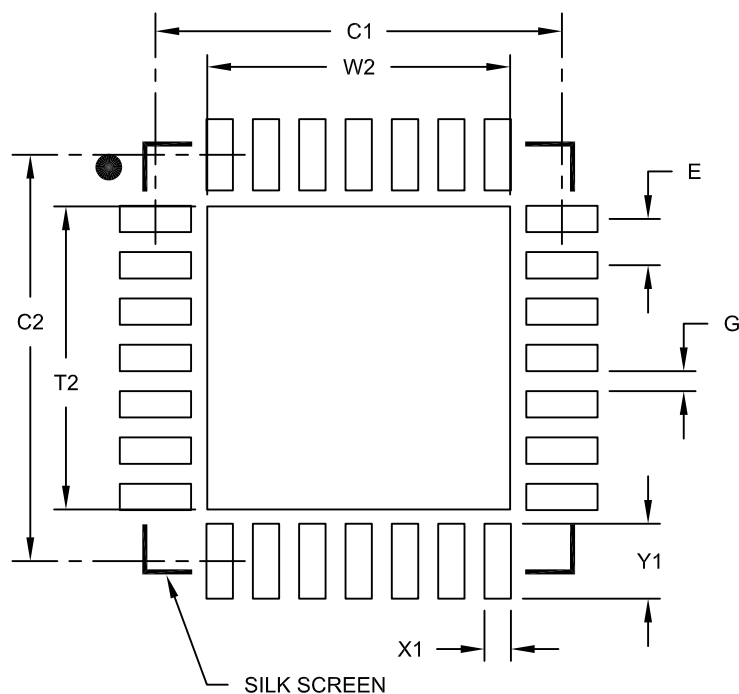
FIGURE 23-20: I²C BUS START/STOP BITS TIMING



PIC16(L)F722/3/4/6/7

28-Lead Plastic Quad Flat, No Lead Package (ML) – 6x6 mm Body [QFN] with 0.55 mm Contact Length

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



RECOMMENDED LAND PATTERN

Dimension	Units	MILLIMETERS		
		MIN	NOM	MAX
Contact Pitch	E	0.65 BSC		
Optional Center Pad Width	W2			4.25
Optional Center Pad Length	T2			4.25
Contact Pad Spacing	C1		5.70	
Contact Pad Spacing	C2		5.70	
Contact Pad Width (X28)	X1			0.37
Contact Pad Length (X28)	Y1			1.00
Distance Between Pads	G	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2105A