



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	Obsolete
Core Processor	PIC
Core Size	8-Bit
Speed	8MHz
Connectivity	I ² C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	66
Program Memory Size	32KB (16K x 16)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	902 x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	A/D 16x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	80-TQFP
Supplier Device Package	80-TQFP (12x12)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic17lc766t-08i-pt

TABLE 3-1: PINOUT DESCRIPTIONS (CONTINUED)

Name	PIC17C75X			PIC17C76X		I/O/P Type	Buffer Type	Description
	DIP No.	PLCC No.	TQFP No.	PLCC No.	QFP No.			
RC0/AD0	2	3	58	3	72	I/O	TTL	<p>PORTC is a bi-directional I/O Port.</p> <p>This is also the least significant byte (LSB) of the 16-bit wide system bus in Microprocessor mode or Extended Microcontroller mode. In multiplexed system bus configuration, these pins are address output as well as data input or output.</p>
RC1/AD1	63	67	55	83	69	I/O	TTL	
RC2/AD2	62	66	54	82	68	I/O	TTL	
RC3/AD3	61	65	53	81	67	I/O	TTL	
RC4/AD4	60	64	52	80	66	I/O	TTL	
RC5/AD5	58	63	51	79	65	I/O	TTL	
RC6/AD6	58	62	50	78	64	I/O	TTL	
RC7/AD7	57	61	49	77	63	I/O	TTL	
RD0/AD8	10	11	2	15	4	I/O	TTL	<p>PORTD is a bi-directional I/O Port.</p> <p>This is also the most significant byte (MSB) of the 16-bit system bus in Microprocessor mode or Extended Microcontroller mode. In multiplexed system bus configuration, these pins are address output as well as data input or output.</p>
RD1/AD9	9	10	1	14	3	I/O	TTL	
RD2/AD10	8	9	64	9	78	I/O	TTL	
RD3/AD11	7	8	63	8	77	I/O	TTL	
RD4/AD12	6	7	62	7	76	I/O	TTL	
RD5/AD13	5	6	61	6	75	I/O	TTL	
RD6/AD14	4	5	60	5	74	I/O	TTL	
RD7/AD15	3	4	59	4	73	I/O	TTL	
RE0/ALE	11	12	3	16	5	I/O	TTL	<p>PORTE is a bi-directional I/O Port.</p> <p>In Microprocessor mode or Extended Microcontroller mode, RE0 is the Address Latch Enable (ALE) output. Address should be latched on the falling edge of ALE output.</p> <p>In Microprocessor or Extended Microcontroller mode, RE1 is the Output Enable (\overline{OE}) control output (active low).</p> <p>In Microprocessor or Extended Microcontroller mode, RE2 is the Write Enable (\overline{WR}) control output (active low).</p> <p>RE3 can also be the Capture4 input pin.</p>
RE1/ \overline{OE}	12	13	4	17	6	I/O	TTL	
RE2/ \overline{WR}	13	14	5	18	7	I/O	TTL	
RE3/CAP4	14	15	6	19	8	I/O	ST	
RF0/AN4	26	28	18	36	24	I/O	ST	<p>PORTF is a bi-directional I/O Port.</p> <p>RF0 can also be analog input 4.</p> <p>RF1 can also be analog input 5.</p> <p>RF2 can also be analog input 6.</p> <p>RF3 can also be analog input 7.</p> <p>RF4 can also be analog input 8.</p> <p>RF5 can also be analog input 9.</p> <p>RF6 can also be analog input 10.</p> <p>RF7 can also be analog input 11.</p>
RF1/AN5	25	27	17	35	23	I/O	ST	
RF2/AN6	24	26	16	30	18	I/O	ST	
RF3/AN7	23	25	15	29	17	I/O	ST	
RF4/AN8	22	24	14	28	16	I/O	ST	
RF5/AN9	21	23	13	27	15	I/O	ST	
RF6/AN10	20	22	12	26	14	I/O	ST	
RF7/AN11	19	21	11	25	13	I/O	ST	

Legend: I = Input only; O = Output only; I/O = Input/Output;
P = Power; — = Not Used; TTL = TTL input; ST = Schmitt Trigger input

Note 1: The output is only available by the peripheral operation.
Note 2: Open drain input/output pin. Pin forced to input upon any device RESET.

PIC17C7XX

In Figure 5-5, Figure 5-6 and Figure 5-7, the TPWRT timer time-out is greater than the TOST timer time-out, as would be the case in higher frequency crystals. For lower frequency crystals (i.e., 32 kHz), TOST may be greater.

FIGURE 5-5: TIME-OUT SEQUENCE ON POWER-UP ($\overline{\text{MCLR}}$ TIED TO V_{DD})

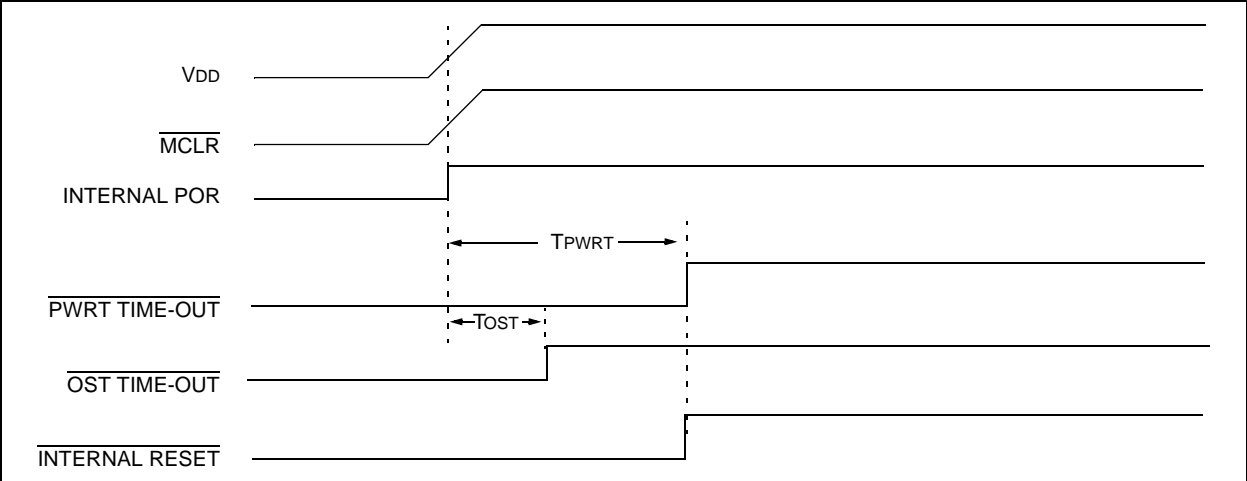


FIGURE 5-6: TIME-OUT SEQUENCE ON POWER-UP ($\overline{\text{MCLR}}$ NOT TIED TO V_{DD})

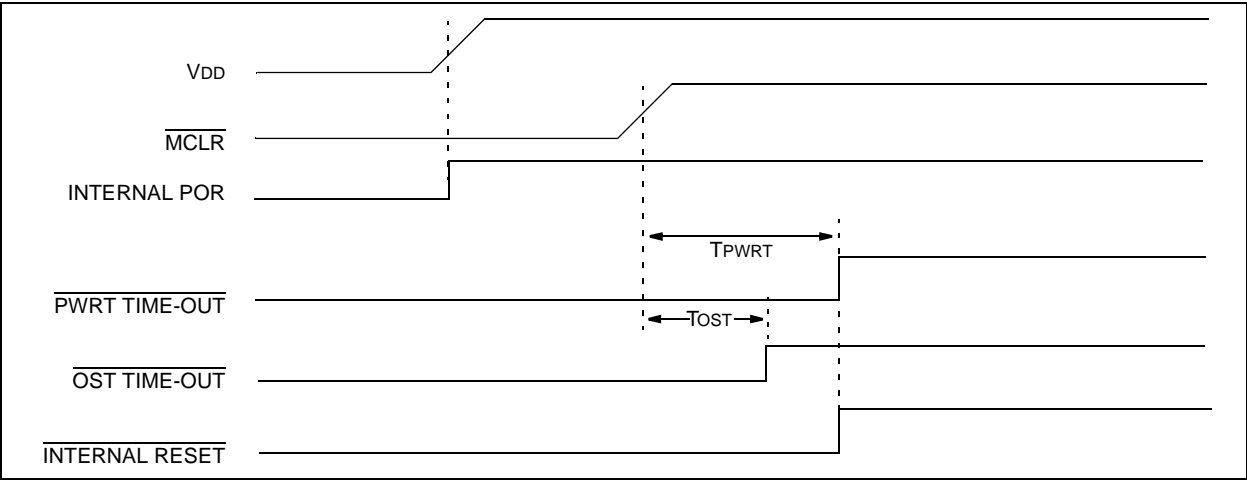
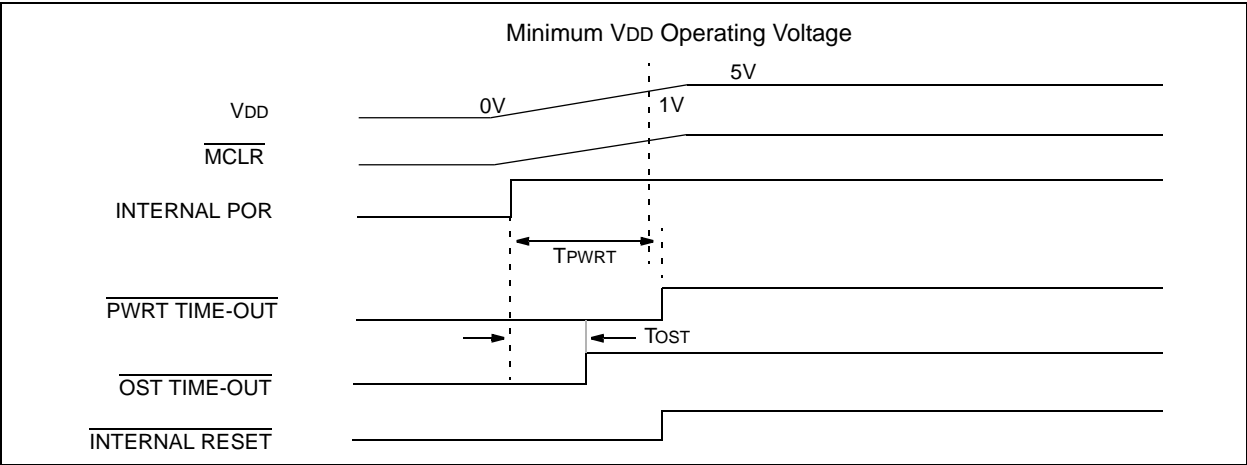


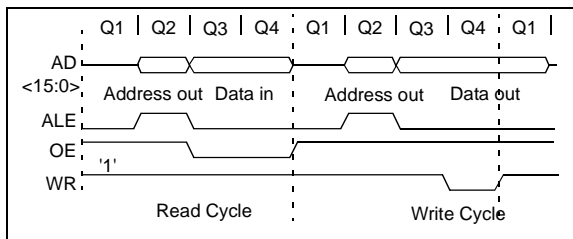
FIGURE 5-7: SLOW RISE TIME ($\overline{\text{MCLR}}$ TIED TO V_{DD})



7.1.2 EXTERNAL MEMORY INTERFACE

When either Microprocessor or Extended Microcontroller mode is selected, PORTC, PORTD and PORTE are configured as the system bus. PORTC and PORTD are the multiplexed address/data bus and PORTE<2:0> is for the control signals. External components are needed to demultiplex the address and data. This can be done as shown in Figure 7-4. The waveforms of address and data are shown in Figure 7-3. For complete timings, please refer to the electrical specification section.

FIGURE 7-3: EXTERNAL PROGRAM MEMORY ACCESS WAVEFORMS



The system bus requires that there is no bus conflict (minimal leakage), so the output value (address) will be capacitively held at the desired value.

As the speed of the processor increases, external EPROM memory with faster access time must be used. Table 7-2 lists external memory speed requirements for a given PIC17C7XX device frequency.

In Extended Microcontroller mode, when the device is executing out of internal memory, the control signals will continue to be active. That is, they indicate the action that is occurring in the internal memory. The external memory access is ignored.

The following selection is for use with Microchip EPROMs. For interfacing to other manufacturers memory, please refer to the electrical specifications of the desired PIC17C7XX device, as well as the desired memory device to ensure compatibility.

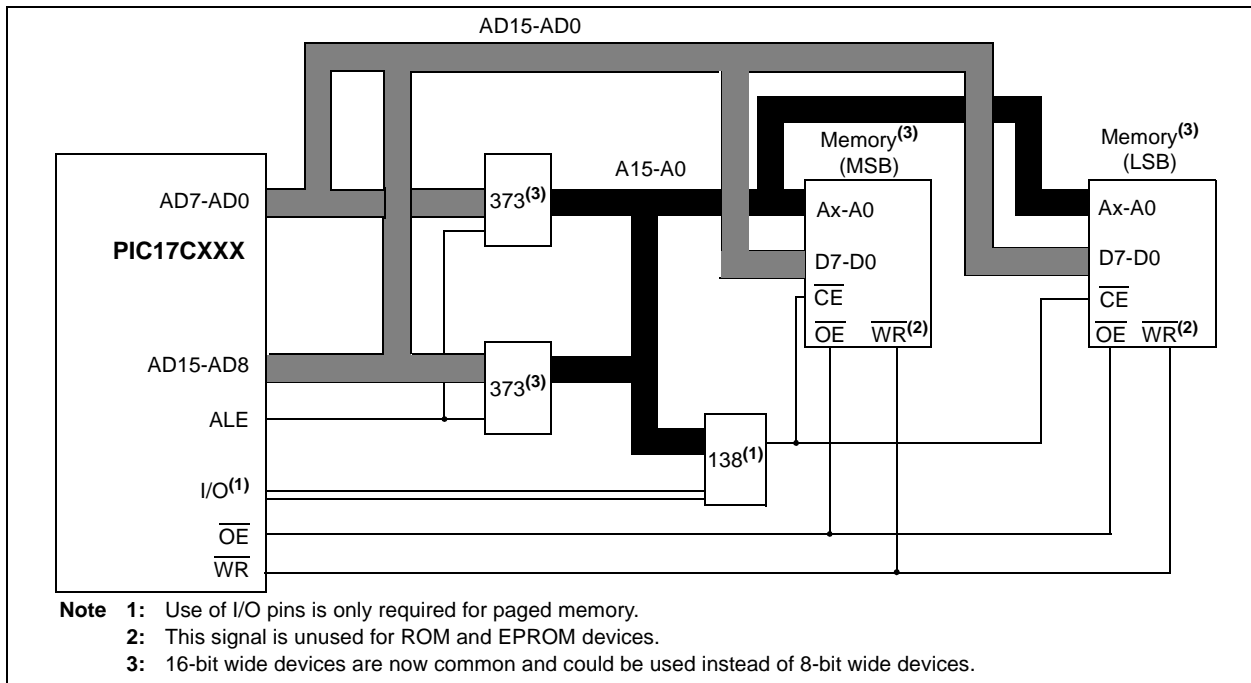
TABLE 7-2: EPROM MEMORY ACCESS TIME ORDERING SUFFIX

PIC17C7XX Oscillator Frequency	Instruction Cycle Time (Tcy)	EPROM Suffix
8 MHz	500 ns	-25
16 MHz	250 ns	-15
20 MHz	200 ns	-10
25 MHz	160 ns	-70

Note: The access times for this requires the use of fast SRAMs.

The electrical specifications now include timing specifications for the memory interface with PIC17LCXXX devices. These specifications reflect the capability of the device by characterization. Please validate your design with these timings.

FIGURE 7-4: TYPICAL EXTERNAL PROGRAM MEMORY CONNECTION DIAGRAM



8.1 Table Writes to Internal Memory

A table write operation to internal memory causes a long write operation. The long write is necessary for programming the internal EPROM. Instruction execution is halted while in a long write cycle. The long write will be terminated by any enabled interrupt. To ensure that the EPROM location has been well programmed, a minimum programming time is required (see specification #D114). Having only one interrupt enabled to terminate the long write ensures that no unintentional interrupts will prematurely terminate the long write.

The sequence of events for programming an internal program memory location should be:

1. Disable all interrupt sources, except the source to terminate EPROM program write.
2. Raise MCLR/VPP pin to the programming voltage.
3. Clear the WDT.
4. Do the table write. The interrupt will terminate the long write.
5. Verify the memory location (table read).

Note 1: Programming requirements must be met. See timing specification in electrical specifications for the desired device. Violating these specifications (including temperature) may result in EPROM locations that are not fully programmed and may lose their state over time.

2: If the VPP requirement is not met, the table write is a 2-cycle write and the program memory is unchanged.

8.1.1 TERMINATING LONG WRITES

An interrupt source or RESET are the only events that terminate a long write operation. Terminating the long write from an interrupt source requires that the interrupt enable and flag bits are set. The GLINTD bit only enables the vectoring to the interrupt address.

If the T0CKI, RA0/INT, or TMR0 interrupt source is used to terminate the long write, the interrupt flag of the highest priority enabled interrupt, will terminate the long write and automatically be cleared.

Note 1: If an interrupt is pending, the TABLWRT is aborted (a NOP is executed). The highest priority pending interrupt, from the T0CKI, RA0/INT, or TMR0 sources that is enabled, has its flag cleared.

2: If the interrupt is not being used for the program write timing, the interrupt should be disabled. This will ensure that the interrupt is not lost, nor will it terminate the long write prematurely.

If a peripheral interrupt source is used to terminate the long write, the interrupt enable and flag bits must be set. The interrupt flag will not be automatically cleared upon the vectoring to the interrupt vector address.

The GLINTD bit determines whether the program will branch to the interrupt vector when the long write is terminated. If GLINTD is clear, the program will vector, if GLINTD is set, the program will not vector to the interrupt address.

TABLE 8-1: INTERRUPT - TABLE WRITE INTERACTION

Interrupt Source	GLINTD	Enable Bit	Flag Bit	Action
RA0/INT, TMR0, T0CKI	0	1	1	Terminate long table write (to internal program memory), branch to interrupt vector (branch clears flag bit).
	0	1	0	None.
	1	0	x	None.
	1	1	1	Terminate long table write, do not branch to interrupt vector (flag is automatically cleared).
Peripheral	0	1	1	Terminate long table write, branch to interrupt vector.
	0	1	0	None.
	1	0	x	None.
	1	1	1	Terminate long table write, do not branch to interrupt vector (flag remains set).

PIC17C7XX

10.3 PORTC and DDRC Registers

PORTC is an 8-bit bi-directional port. The corresponding data direction register is DDRC. A '1' in DDRC configures the corresponding port pin as an input. A '0' in the DDRC register configures the corresponding port pin as an output. Reading PORTC reads the status of the pins, whereas writing to PORTC will write to the port latch. PORTC is multiplexed with the system bus. When operating as the system bus, PORTC is the low order byte of the address/data bus (AD7:AD0). The timing for the system bus is shown in the Electrical Specifications section.

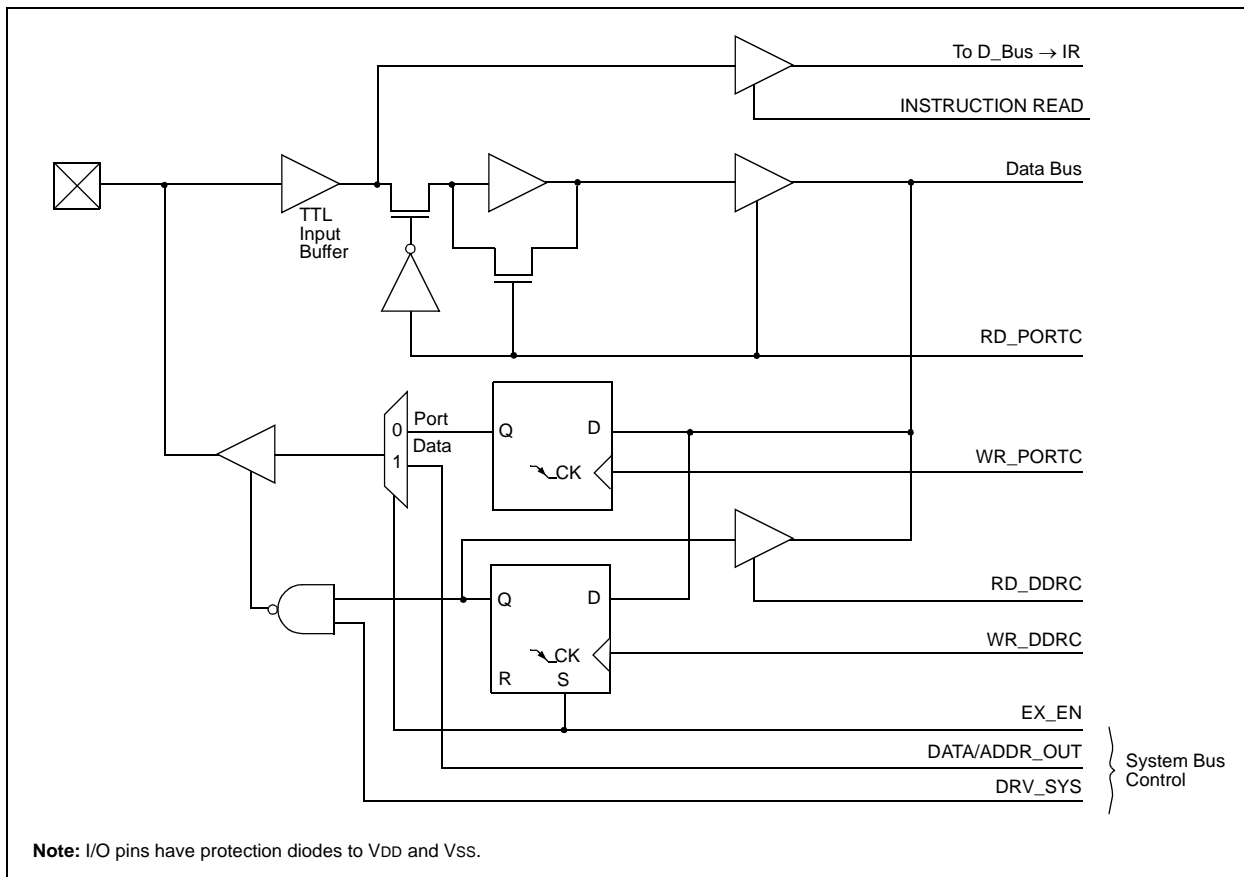
Note: This port is configured as the system bus when the device's configuration bits are selected to Microprocessor or Extended Microcontroller modes. In the two other microcontroller modes, this port is a general purpose I/O.

Example 10-3 shows an instruction sequence to initialize PORTC. The Bank Select Register (BSR) must be selected to Bank 1 for the port to be initialized. The following example uses the MOVLB instruction to load the BSR register for bank selection.

EXAMPLE 10-3: INITIALIZING PORTC

```
MOVLB 1      ; Select Bank 1
CLRF  PORTC, F ; Initialize PORTC data
              ; latches before setting
              ; the data direction reg
MOVLW 0xCF    ; Value used to initialize
              ; data direction
MOVWF DDRC    ; Set RC<3:0> as inputs
              ; RC<5:4> as outputs
              ; RC<7:6> as inputs
```

FIGURE 10-9: BLOCK DIAGRAM OF RC7:RC0 PORT PINS



11.0 OVERVIEW OF TIMER RESOURCES

The PIC17C7XX has four timer modules. Each module can generate an interrupt to indicate that an event has occurred. These timers are called:

- Timer0 - 16-bit timer with programmable 8-bit prescaler
- Timer1 - 8-bit timer
- Timer2 - 8-bit timer
- Timer3 - 16-bit timer

For enhanced time base functionality, four input Captures and three Pulse Width Modulation (PWM) outputs are possible. The PWMs use the Timer1 and Timer2 resources and the input Captures use the Timer3 resource.

11.1 Timer0 Overview

The Timer0 module is a simple 16-bit overflow counter. The clock source can be either the internal system clock ($F_{osc}/4$) or an external clock.

When Timer0 uses an external clock source, it has the flexibility to allow user selection of the incrementing edge, rising or falling.

The Timer0 module also has a programmable prescaler. The T0PS3:T0PS0 bits (T0STA<4:1>) determine the prescale value. TMR0 can increment at the following rates: 1:1, 1:2, 1:4, 1:8, 1:16, 1:32, 1:64, 1:128, 1:256.

Synchronization of the external clock occurs after the prescaler. When the prescaler is used, the external clock frequency may be higher than the device's frequency. The maximum external frequency on the T0CKI pin is 50 MHz, given the high and low time requirements of the clock.

11.2 Timer1 Overview

The Timer1 module is an 8-bit timer/counter with an 8-bit period register (PR1). When the TMR1 value rolls over from the period match value to 0h, the TMR1IF flag is set and an interrupt will be generated if enabled. In Counter mode, the clock comes from the RB4/TCLK12 pin, which can also be selected to be the clock for the Timer2 module.

TMR1 can be concatenated with TMR2 to form a 16-bit timer. The TMR1 register is the LSB and TMR2 is the MSB. When in the 16-bit timer mode, there is a corresponding 16-bit period register (PR2:PR1). When the TMR2:TMR1 value rolls over from the period match value to 0h, the TMR1IF flag is set and an interrupt will be generated, if enabled.

11.3 Timer2 Overview

The Timer2 module is an 8-bit timer/counter with an 8-bit period register (PR2). When the TMR2 value rolls over from the period match value to 0h, the TMR2IF flag is set and an interrupt will be generated, if enabled. In Counter mode, the clock comes from the RB4/TCLK12 pin, which can also provide the clock for the Timer1 module.

TMR2 can be concatenated with TMR1 to form a 16-bit timer. The TMR2 register is the MSB and TMR1 is the LSB. When in the 16-bit timer mode, there is a corresponding 16-bit period register (PR2:PR1). When the TMR2:TMR1 value rolls over from the period match value to 0h, the TMR1IF flag is set and an interrupt will be generated, if enabled.

11.4 Timer3 Overview

The Timer3 module is a 16-bit timer/counter with a 16-bit period register. When the TMR3H:TMR3L value rolls over to 0h, the TMR3IF bit is set and an interrupt will be generated, if enabled. In Counter mode, the clock comes from the RB5/TCLK3 pin.

When operating in the four Capture modes, the period registers become the second (of four) 16-bit capture registers.

11.5 Role of the Timer/Counters

The timer modules are general purpose, but have dedicated resources associated with them. Timer1 and Timer2 are the time bases for the three Pulse Width Modulation (PWM) outputs, while Timer3 is the time base for the four input captures.

This mode (3 Capture, 1 Period) is selected if control bit CA1/PR3 is clear. In this mode, the Capture1 register, consisting of high byte (PR3H/CA1H) and low byte (PR3L/CA1L), is configured as the period control register for TMR3. Capture1 is disabled in this mode and the corresponding interrupt bit, CA1IF, is never set. TMR3 increments until it equals the value in the period register and then resets to 0000h on the next timer clock.

All other Captures are active in this mode.

13.2.1.1 Capture Operation

The CAXED1 and CAXED0 bits determine the event on which capture will occur. The possible events are:

- Capture on every falling edge
- Capture on every rising edge
- Capture every 4th rising edge
- Capture every 16th rising edge

When a capture takes place, an interrupt flag is latched into the CAXIF bit. This interrupt can be enabled by setting the corresponding mask bit CAXIE. The Peripheral Interrupt Enable bit (PEIE) must be set and the Global Interrupt Disable bit (GLINTD) must be cleared for the interrupt to be acknowledged. The CAXIF interrupt flag bit is cleared in software.

When the capture prescale select is changed, the prescaler is not reset and an event may be generated. Therefore, the first capture after such a change will be ambiguous. However, it sets the time-base for the next capture. The prescaler is reset upon chip RESET.

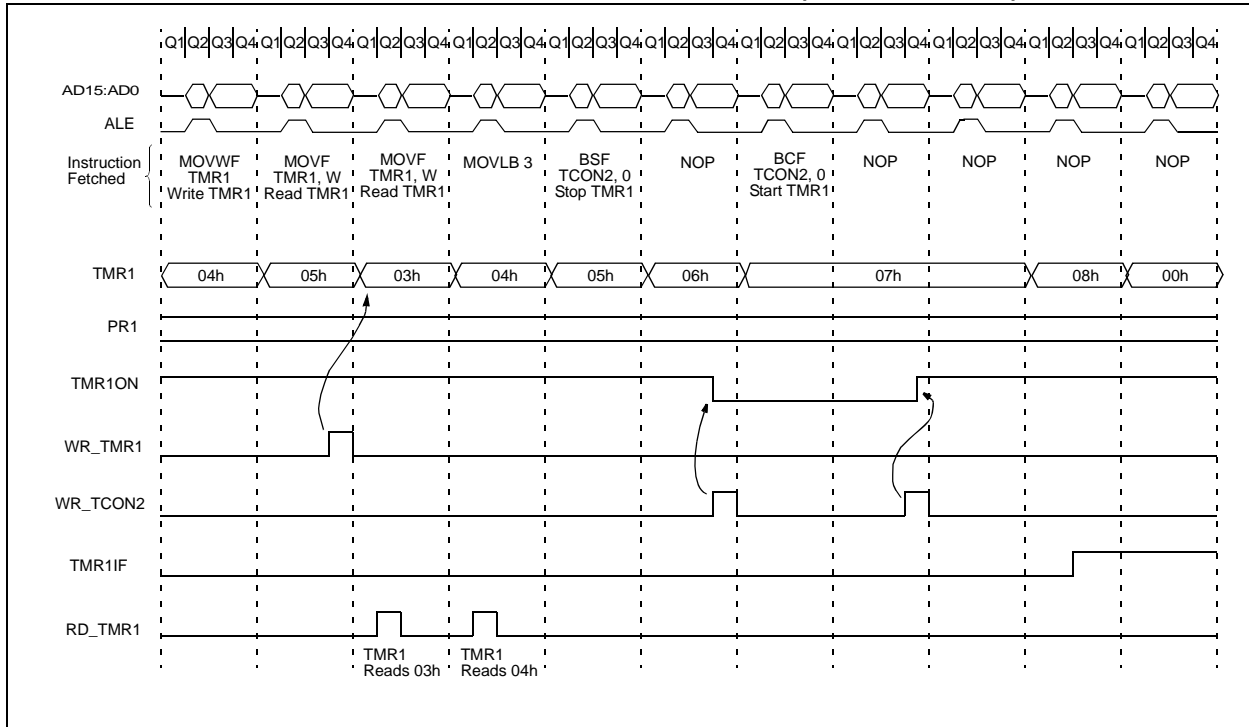
The capture pin, CAPx, is a multiplexed pin. When used as a port pin, the capture is not disabled. However, the user can simply disable the Capture interrupt by clearing CAXIE. If the CAPx pin is used as an output pin, the user can activate a capture by writing to the port pin. This may be useful during development phase to emulate a capture interrupt.

The input on the capture pin CAPx is synchronized internally to internal phase clocks. This imposes certain restrictions on the input waveform (see the Electrical Specification section for timing).

The capture overflow status flag bit is double buffered. The master bit is set if one captured word is already residing in the Capture register (CAXH:CAXL) and another “event” has occurred on the CAPx pin. The new event will not transfer the TMR3 value to the capture register, protecting the previous unread capture value. When the user reads both the high and the low bytes (in any order) of the Capture register, the master overflow bit is transferred to the slave overflow bit (CAXOVF) and then the master bit is reset. The user can then read TCONx to determine the value of CAXOVF.

The recommended sequence to read capture registers and capture overflow flag bits is shown in Example 13-1.

FIGURE 13-8: TIMER1, TIMER2 AND TIMER3 OPERATION (IN TIMER MODE)



PIC17C7XX

NOTES:

14.2 USART Asynchronous Mode

In this mode, the USART uses standard nonreturn-to-zero (NRZ) format (one START bit, eight or nine data bits, and one STOP bit). The most common data format is 8-bits. An on-chip dedicated 8-bit baud rate generator can be used to derive standard baud rate frequencies from the oscillator. The USART's transmitter and receiver are functionally independent but use the same data format and baud rate. The baud rate generator produces a clock x64 of the bit shift rate. Parity is not supported by the hardware, but can be implemented in software (and stored as the ninth data bit). Asynchronous mode is stopped during SLEEP.

The Asynchronous mode is selected by clearing the SYNC bit (TXSTA<4>).

The USART Asynchronous module consists of the following components:

- Baud Rate Generator
- Sampling Circuit
- Asynchronous Transmitter
- Asynchronous Receiver

14.2.1 USART ASYNCHRONOUS TRANSMITTER

The USART transmitter block diagram is shown in Figure 14-1. The heart of the transmitter is the transmit shift register (TSR). The shift register obtains its data from the read/write transmit buffer (TXREG). TXREG is loaded with data in software. The TSR is not loaded until the STOP bit has been transmitted from the previous load. As soon as the STOP bit is transmitted, the TSR is loaded with new data from the TXREG (if available). Once TXREG transfers the data to the TSR (occurs in one Tcy at the end of the current BRG cycle), the TXREG is empty and an interrupt bit, TXIF, is set. This interrupt can be enabled/disabled by setting/clearing the TXIE bit. TXIF will be set, regardless of TXIE and cannot be reset in software. It will reset only when new data is loaded into TXREG. While TXIF indicates the status of the TXREG, the TRMT (TXSTA<1>) bit shows the status of the TSR.

TRMT is a read only bit which is set when the TSR is empty. No interrupt logic is tied to this bit, so the user has to poll this bit in order to determine if the TSR is empty.

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

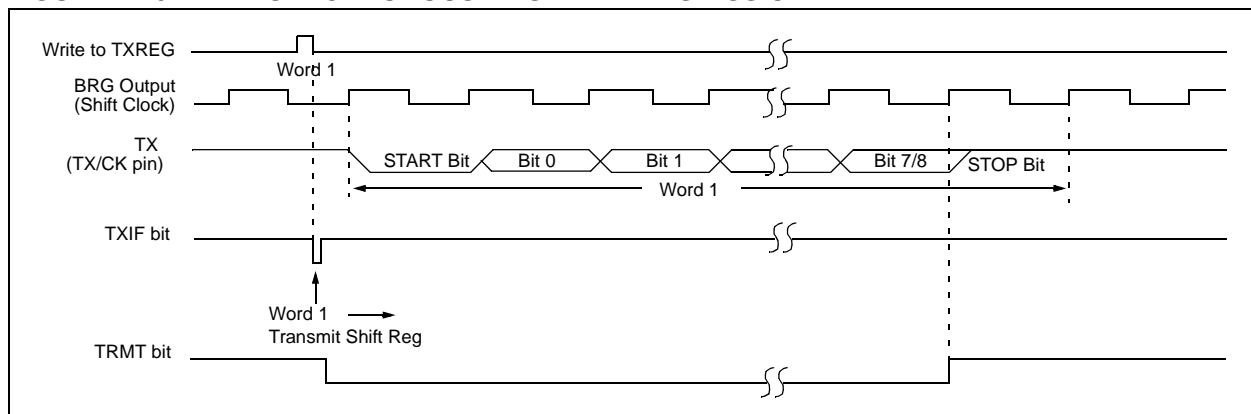
Transmission is enabled by setting the TXEN (TXSTA<5>) bit. The actual transmission will not occur until TXREG has been loaded with data and the baud rate generator (BRG) has produced a shift clock (Figure 14-3). The transmission can also be started by first loading TXREG and then setting TXEN. Normally, when transmission is first started, the TSR is empty, so a transfer to TXREG will result in an immediate transfer to TSR, resulting in an empty TXREG. A back-to-back transfer is thus possible (Figure 14-4). Clearing TXEN during a transmission will cause the transmission to be aborted. This will reset the transmitter and the TX/CK pin will revert to hi-impedance.

In order to select 9-bit transmission, the TX9 (TXSTA<6>) bit should be set and the ninth bit value should be written to TX9D (TXSTA<0>). The ninth bit value must be written before writing the 8-bit data to the TXREG. This is because a data write to TXREG can result in an immediate transfer of the data to the TSR (if the TSR is empty).

Steps to follow when setting up an Asynchronous Transmission:

1. Initialize the SPBRG register for the appropriate baud rate.
2. Enable the asynchronous serial port by clearing the SYNC bit and setting the SPEN bit.
3. If interrupts are desired, then set the TXIE bit.
4. If 9-bit transmission is desired, then set the TX9 bit.
5. If 9-bit transmission is selected, the ninth bit should be loaded in TX9D.
6. Load data to the TXREG register.
7. Enable the transmission by setting TXEN (starts transmission).

FIGURE 14-3: ASYNCHRONOUS MASTER TRANSMISSION



14.3 USART Synchronous Master Mode

In Master Synchronous mode, the data is transmitted in a half-duplex manner; i.e., transmission and reception do not occur at the same time: when transmitting data, the reception is inhibited and vice versa. The synchronous mode is entered by setting the SYNC (TXSTA<4>) bit. In addition, the SPEN (RCSTA<7>) bit is set in order to configure the I/O pins to CK (clock) and DT (data) lines, respectively. The Master mode indicates that the processor transmits the master clock on the CK line. The Master mode is entered by setting the CSRC (TXSTA<7>) bit.

14.3.1 USART SYNCHRONOUS MASTER TRANSMISSION

The USART transmitter block diagram is shown in Figure 14-1. The heart of the transmitter is the transmit (serial) shift register (TSR). The shift register obtains its data from the read/write transmit buffer TXREG. TXREG is loaded with data in software. The TSR is not loaded until the last bit has been transmitted from the previous load. As soon as the last bit is transmitted, the TSR is loaded with new data from TXREG (if available). Once TXREG transfers the data to the TSR (occurs in one Tcy at the end of the current BRG cycle), TXREG is empty and the TXIF bit is set. This interrupt can be enabled/disabled by setting/clearing the TXIE bit. TXIF will be set regardless of the state of bit TXIE and cannot be cleared in software. It will reset only when new data is loaded into TXREG. While TXIF indicates the status of TXREG, TRMT (TXSTA<1>) shows the status of the TSR. TRMT is a read only bit which is set when the TSR is empty. No interrupt logic is tied to this bit, so the user has to poll this bit in order to determine if the TSR is empty. The TSR is not mapped in data memory, so it is not available to the user.

Transmission is enabled by setting the TXEN (TXSTA<5>) bit. The actual transmission will not occur until TXREG has been loaded with data. The first data bit will be shifted out on the next available rising edge of the clock on the TX/CK pin. Data out is stable around the falling edge of the synchronous clock (Figure 14-9). The transmission can also be started by first loading TXREG and then setting TXEN. This is advantageous when slow baud rates are selected, since BRG is kept in RESET when the TXEN, CREN, and SREN bits are clear. Setting the TXEN bit will start the BRG, creating a shift clock immediately. Normally when transmission is first started, the TSR is empty, so a transfer to TXREG will result in an immediate transfer to the TSR, resulting in an empty TXREG. Back-to-back transfers are possible.

Clearing TXEN during a transmission will cause the transmission to be aborted and will reset the transmitter. The RX/DT and TX/CK pins will revert to hi-impedance. If either CREN or SREN are set during a transmission, the transmission is aborted and the RX/DT pin reverts to a hi-impedance state (for a reception). The TX/CK pin will remain an output if the CSRC bit is set (internal clock). The transmitter logic is not reset, although it is disconnected from the pins. In order to reset the transmitter, the user has to clear the TXEN bit. If the SREN bit is set (to interrupt an ongoing transmission and receive a single word), then after the single word is received, SREN will be cleared and the serial port will revert back to transmitting, since the TXEN bit is still set. The DT line will immediately switch from hi-impedance Receive mode to transmit and start driving. To avoid this, TXEN should be cleared.

In order to select 9-bit transmission, the TX9 (TXSTA<6>) bit should be set and the ninth bit should be written to TX9D (TXSTA<0>). The ninth bit must be written before writing the 8-bit data to TXREG. This is because a data write to TXREG can result in an immediate transfer of the data to the TSR (if the TSR is empty). If the TSR was empty and TXREG was written before writing the "new" TX9D, the "present" value of TX9D is loaded.

Steps to follow when setting up a Synchronous Master Transmission:

1. Initialize the SPBRG register for the appropriate baud rate (see Baud Rate Generator Section for details).
2. Enable the synchronous master serial port by setting the SYNC, SPEN, and CSRC bits.
3. Ensure that the CREN and SREN bits are clear (these bits override transmission when set).
4. If interrupts are desired, then set the TXIE bit (the GLINTD bit must be clear and the PEIE bit must be set).
5. If 9-bit transmission is desired, then set the TX9 bit.
6. If 9-bit transmission is selected, the ninth bit should be loaded in TX9D.
7. Start transmission by loading data to the TXREG register.
8. Enable the transmission by setting TXEN.

Writing the transmit data to the TXREG, then enabling the transmit (setting TXEN), allows transmission to start sooner than doing these two events in the reverse order.

Note: To terminate a transmission, either clear the SPEN bit, or the TXEN bit. This will reset the transmit logic, so that it will be in the proper state when transmit is re-enabled.

PIC17C7XX

15.2.18.3 Bus Collision During a STOP Condition

Bus collision occurs during a STOP condition if:

- After the SDA pin has been de-asserted and allowed to float high, SDA is sampled low after the BRG has timed out.
- After the SCL pin is de-asserted, SCL is sampled low before SDA goes high.

The STOP condition begins with SDA asserted low. When SDA is sampled low, the SCL pin is allowed to float. When the pin is sampled high (clock arbitration), the baud rate generator is loaded with SSPADD<6:0> and counts down to '0'. After the BRG times out, SDA is sampled. If SDA is sampled low, a bus collision has occurred. This is due to another master attempting to drive a data '0'. If the SCL pin is sampled low before SDA is allowed to float high, a bus collision occurs. This is another case of another master attempting to drive a data '0' (Figure 15-40).

FIGURE 15-40: BUS COLLISION DURING A STOP CONDITION (CASE 1)

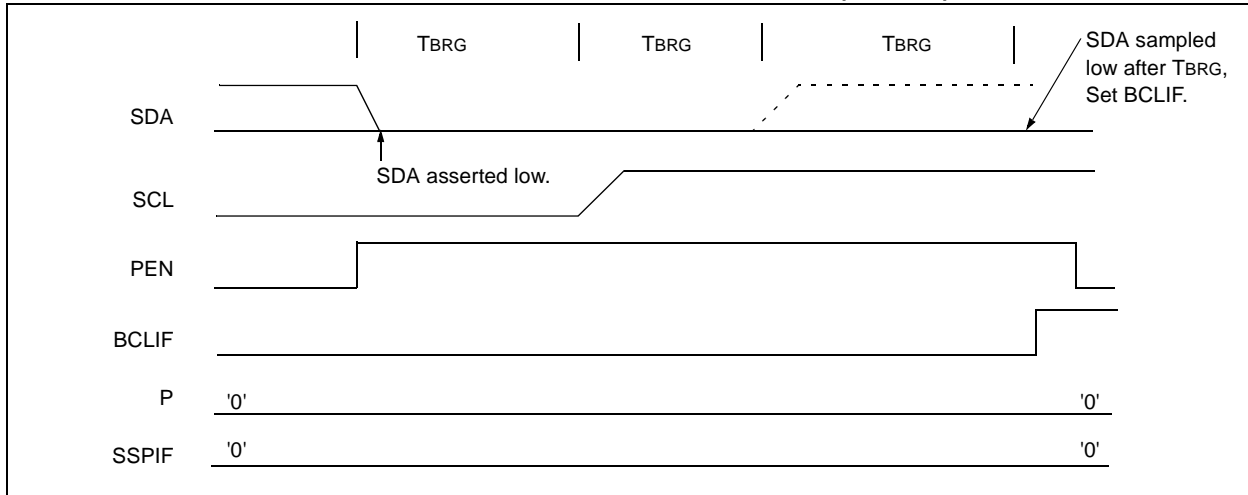
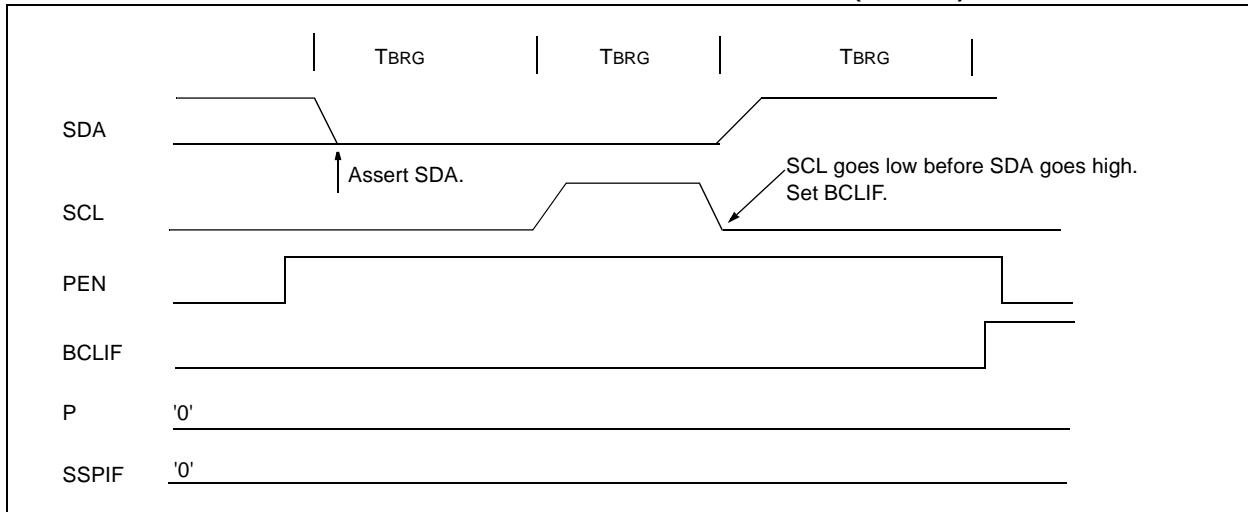


FIGURE 15-41: BUS COLLISION DURING A STOP CONDITION (CASE 2)



17.0 SPECIAL FEATURES OF THE CPU

What sets a microcontroller apart from other processors are special circuits to deal with the needs of real-time applications. The PIC17CXXX family has a host of such features intended to maximize system reliability, minimize cost through elimination of external components, provide power saving operating modes and offer code protection. These are:

- Oscillator Selection (Section 4.0)
- RESET (Section 5.0)
 - Power-on Reset (POR)
 - Power-up Timer (PWRT)
 - Oscillator Start-up Timer (OST)
 - Brown-out Reset (BOR)
- Interrupts (Section 6.0)
- Watchdog Timer (WDT)
- SLEEP mode
- Code protection

The PIC17CXXX has a Watchdog Timer which can be shut-off only through EPROM bits. It runs off its own RC oscillator for added reliability. There are two timers that offer necessary delays on POR and BOR. One is the Oscillator Start-up Timer (OST), intended to keep the chip in RESET until the crystal oscillator is stable. The other is the Power-up Timer (PWRT), which provides a fixed delay of 96 ms (nominal) on power-up only, designed to keep the part in RESET while the power supply stabilizes. With these two timers on-chip, most applications need no external RESET circuitry.

The SLEEP mode is designed to offer a very low current power-down mode. The user can wake from SLEEP through external RESET, Watchdog Timer Reset, or through an interrupt. Several oscillator options are also made available to allow the part to fit the application. The RC oscillator option saves system cost, while the LF crystal option saves power. Configuration bits are used to select various options. This configuration word has the format shown in Figure 17-1.

REGISTER 17-1: CONFIGURATION WORDS

High (H) Table Read Addr. FE0Fh - FE08h	U-x	R/P-1	R/P-1	U-x	U-x	U-x	U-x	U-x	U-x
	—	PM2	BODEN	—	—	—	—	—	—
	bit 15	bit 8	bit 7						bit 0
Low (L) Table Read Addr. FE07h - FE00h	U-x	U-x	R/P-1	U-x	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
	—	—	PM1	—	PM0	WDTPS1	WDTPS0	FOSC1	FOSC0
	bit 15	bit 8	bit 7						bit 0
bits 7H, 6L, 4L	PM2, PM1, PM0: Processor Mode Select bits 111 = Microprocessor mode 110 = Microcontroller mode 101 = Extended Microcontroller mode 000 = Code Protected Microcontroller mode								
bit 6H	BODEN: Brown-out Detect Enable 1 = Brown-out Detect circuitry is enabled 0 = Brown-out Detect circuitry is disabled								
bits 3L:2L	WDTPS1:WDTPS0: WDT Postscaler Select bits 11 = WDT enabled, postscaler = 1 10 = WDT enabled, postscaler = 256 01 = WDT enabled, postscaler = 64 00 = WDT disabled, 16-bit overflow timer								
bits 1L:0L	FOSC1:FOSC0: Oscillator Select bits 11 = EC oscillator 10 = XT oscillator 01 = RC oscillator 00 = LF oscillator								
Shaded bits (—)	Reserved								

ADDWFC		ADD WREG and Carry bit to f							
Syntax:	[<i>label</i>] ADDWFC f,d								
Operands:	$0 \leq f \leq 255$ $d \in [0,1]$								
Operation:	$(WREG) + (f) + C \rightarrow (dest)$								
Status Affected:	OV, C, DC, Z								
Encoding:	<table border="1"><tr><td>0001</td><td>000d</td><td>ffff</td><td>ffff</td></tr></table>				0001	000d	ffff	ffff	
0001	000d	ffff	ffff						
Description:	Add WREG, the Carry Flag and data memory location 'f'. If 'd' is 0, the result is placed in WREG. If 'd' is 1, the result is placed in data memory location 'f'.								
Words:	1								
Cycles:	1								
Q Cycle Activity:									
	Q1	Q2	Q3	Q4					
	Decode	Read register 'f'	Process Data	Write to destination					

Example: ADDWFC REG 0

Before Instruction

Carry bit = 1
REG = 0x02
WREG = 0x4D

After Instruction

Carry bit = 0
REG = 0x02
WREG = 0x50

ANDLW		And Literal with WREG						
Syntax:	[<i>label</i>] ANDLW k							
Operands:	$0 \leq k \leq 255$							
Operation:	(WREG) .AND. (k) → (WREG)							
Status Affected:	Z							
Encoding:	<table border="1"><tr><td>1011</td><td>0101</td><td>kkkk</td><td>kkkk</td></tr></table>				1011	0101	kkkk	kkkk
1011	0101	kkkk	kkkk					
Description:	The contents of WREG are AND'd with the 8-bit literal 'k'. The result is placed in WREG.							
Words:	1							
Cycles:	1							
Q Cycle Activity:								
	Q1	Q2	Q3	Q4				
	Decode	Read literal 'k'	Process Data	Write to WREG				

Example: ANDLW 0x5F

Before Instruction

WREG = 0xA3

After Instruction

WREG = 0x03

IORWF		Inclusive OR WREG with f						
Syntax:	[<i>label</i>] IORWF f,d							
Operands:	0 ≤ f ≤ 255 d ∈ [0,1]							
Operation:	(WREG) .OR. (f) → (dest)							
Status Affected:	\overline{Z}							
Encoding:	<table><tr><td>0000</td><td>100d</td><td>ffff</td><td>ffff</td></tr></table>				0000	100d	ffff	ffff
0000	100d	ffff	ffff					
Description:	Inclusive OR WREG with register 'f'. If 'd' is 0, the result is placed in WREG. If 'd' is 1, the result is placed back in register 'f'.							
Words:	1							
Cycles:	1							
Q Cycle Activity:								
	Q1	Q2	Q3	Q4				
	Decode	Read register 'f'	Process Data	Write to destination				

Example: IORWF RESULT, 0

Before Instruction

RESULT = 0x13
WREG = 0x91

After Instruction

RESULT = 0x13
WREG = 0x93

LCALL		Long Call						
Syntax:	[<i>label</i>] LCALL k							
Operands:	0 ≤ k ≤ 255							
Operation:	PC + 1 → TOS; k → PCL, (PCLATH) → PCH							
Status Affected:	None							
Encoding:	<table><tr><td>1011</td><td>0111</td><td>kkkk</td><td>kkkk</td></tr></table>				1011	0111	kkkk	kkkk
1011	0111	kkkk	kkkk					
Description:	<p>LCALL allows an unconditional subroutine call to anywhere within the 64K program memory space.</p> <p>First, the return address (PC + 1) is pushed onto the stack. A 16-bit destination address is then loaded into the program counter. The lower 8-bits of the destination address are embedded in the instruction. The upper 8-bits of PC are loaded from PC high holding latch, PCLATH.</p>							
Words:	1							
Cycles:	2							
Q Cycle Activity:								
	Q1	Q2	Q3	Q4				
	Decode	Read literal 'k'	Process Data	Write register PCL				
	No operation	No operation	No operation	No operation				

Example: MOVLW HIGH(SUBROUTINE)
MOVWF WREG, PCLATH
LCALL LOW(SUBROUTINE)

Before Instruction

SUBROUTINE = 16-bit Address
PC = ?

After Instruction

PC = Address (SUBROUTINE)

PIC17C7XX

FIGURE 20-17: I²C BUS START/STOP BITS TIMING

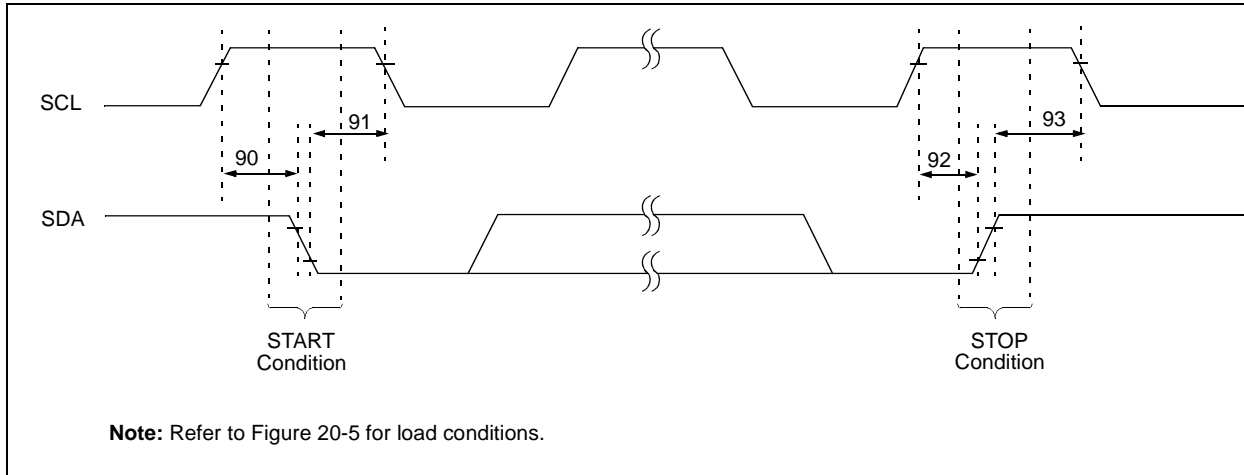


TABLE 20-12: I²C BUS START/STOP BITS REQUIREMENTS

Param. No.	Sym	Characteristic		Min	Typ	Max	Units	Conditions
90	Tsu:sta	START condition Setup time	100 kHz mode	$2(T_{osc})(BRG + 1)$	—	—	ns	Only relevant for Repeated Start condition
			400 kHz mode	$2(T_{osc})(BRG + 1)$	—	—		
			1 MHz mode ⁽¹⁾	$2(T_{osc})(BRG + 1)$	—	—		
91	Thd:sta	START condition Hold time	100 kHz mode	$2(T_{osc})(BRG + 1)$	—	—	ns	After this period, the first clock pulse is generated
			400 kHz mode	$2(T_{osc})(BRG + 1)$	—	—		
			1 MHz mode ⁽¹⁾	$2(T_{osc})(BRG + 1)$	—	—		
92	Tsu:sto	STOP condition Setup time	100 kHz mode	$2(T_{osc})(BRG + 1)$	—	—	ns	
			400 kHz mode	$2(T_{osc})(BRG + 1)$	—	—		
			1 MHz mode ⁽¹⁾	$2(T_{osc})(BRG + 1)$	—	—		
93	Thd:sto	STOP condition Hold time	100 kHz mode	$2(T_{osc})(BRG + 1)$	—	—	ns	
			400 kHz mode	$2(T_{osc})(BRG + 1)$	—	—		
			1 MHz mode ⁽¹⁾	$2(T_{osc})(BRG + 1)$	—	—		

Note 1: Maximum pin capacitance = 10 pF for all I²C pins.

PIC17C7XX

FIGURE 20-23: A/D CONVERSION TIMING

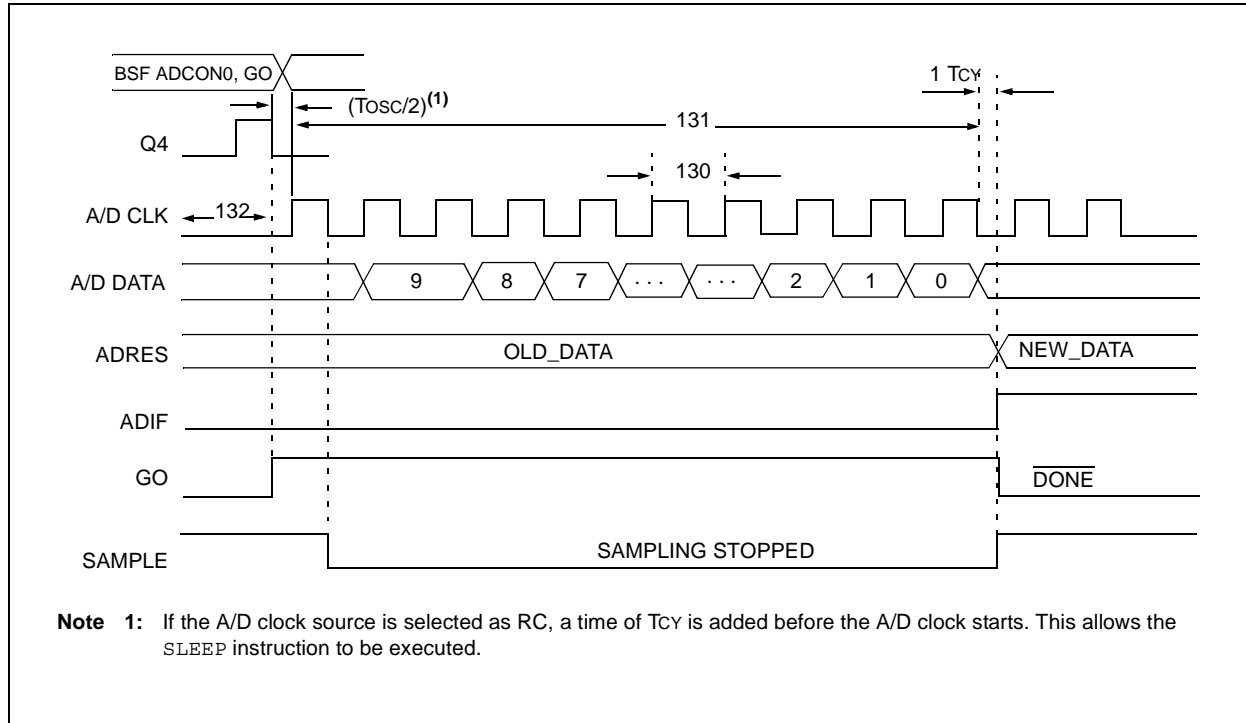


TABLE 20-19: A/D CONVERSION REQUIREMENTS

Param. No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
130	TAD	A/D clock period	PIC17CXXX	1.6	—	—	μs
			PIC17LCXXX	3.0	—	—	μs
			PIC17CXXX	2.0	4.0	6.0	μs
			PIC17LCXXX	3.0	6.0	9.0	μs
131	TCNV	Conversion time (not including acquisition time) (Note 1)	11	—	12	Tad	
132	TACQ	Acquisition time	(Note 2)	20	—	μs	The minimum time is the amplifier settling time. This may be used if the "new" input voltage has not changed by more than 1LSb (i.e., 5 mV @ 5.12V) from the last sampled voltage (as stated on CHOLD).
			10	—	—	μs	
134	TGO	Q4 to ADCLK start	—	Tosc/2	—	—	If the A/D clock source is selected as RC, a time of Tcy is added before the A/D clock starts. This allows the SLEEP instruction to be executed.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated.

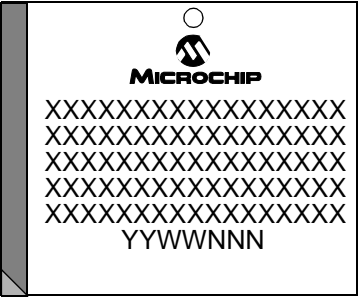
Note 1: ADRES register may be read on the following Tcy cycle.

Note 2: See Section 16.1 for minimum conditions when input voltage has changed more than 1 LSb.

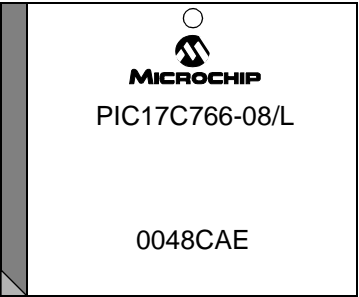
PIC17C7XX

Package Marking Information (Cont.)

84-Lead PLCC



Example



PIC17C7XX

NOTES: