Microchip Technology - PIC17C756AT-16/L Datasheet

Welcome to E-XFL.COM

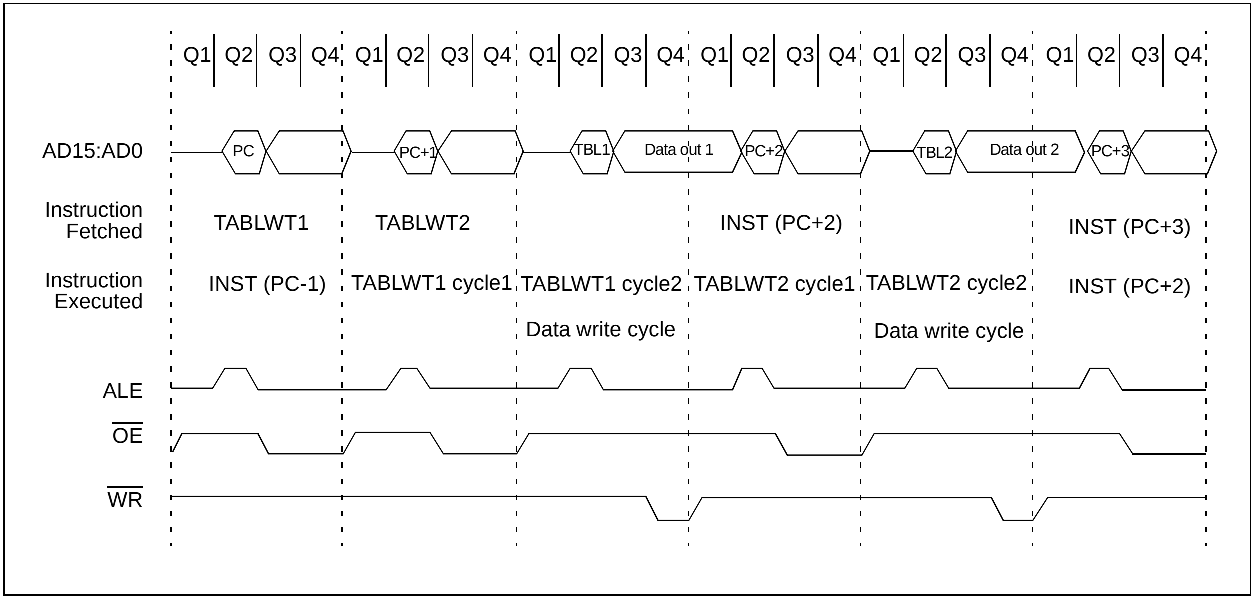
What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	16MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	50
Program Memory Size	32KB (16K x 16)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	902 x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 12x10b
Oscillator Type	External
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	68-LCC (J-Lead)
Supplier Device Package	68-PLCC (24.23x24.23)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic17c756at-16-l

FIGURE 8-6: CONSECUTIVE TABLWT WRITE TIMING (EXTERNAL MEMORY)



Example 9-4 shows the sequence to do a 16 x 16 signed multiply. Equation 9-2 shows the algorithm used. The 32-bit result is stored in four registers, RES3:RES0. To account for the sign bits of the arguments, each argument pairs most significant bit (MSb) is tested and the appropriate subtractions are done.

EQUATION 9-2: 16 x 16 SIGNED MULTIPLICATION ALGORITHM

$$
\begin{aligned}
&\mathit{RES3{:}RES0} \\
&\quad = \mathit{ARG1H{:}ARG1L} \bullet \mathit{ARG2H{:}ARG2L} \\
&\quad = (\mathit{ARG1H} \bullet \mathit{ARG2H} \bullet 2^{16}) && + \\
&\qquad (\mathit{ARG1H} \bullet \mathit{ARG2L} \bullet 2^{8}) && + \\
&\qquad (\mathit{ARG1L} \bullet \mathit{ARG2H} \bullet 2^{8}) && + \\
&\qquad (\mathit{ARG1L} \bullet \mathit{ARG2L}) && + \\
&\qquad (-1 \bullet \mathit{ARG2H}{<}7{>} \bullet \mathit{ARG1H{:}ARG1L} \bullet 2^{16}) && + \\
&\qquad (-1 \bullet \mathit{ARG1H}{<}7{>} \bullet \mathit{ARG2H{:}ARG2L} \bullet 2^{16})
\end{aligned}
$$

EXAMPLE 9-4: 16 x 16 SIGNED MULTIPLY ROUTINE

```
   MOVFP    ARG1L, WREG
   MULWF    ARG2L       ; ARG1L * ARG2L ->
                        ;   PRODH:PRODL
   MOVPF    PRODH, RES1 ;
   MOVPF    PRODL, RES0 ;
;
   MOVFP    ARG1H, WREG
   MULWF    ARG2H       ; ARG1H * ARG2H ->
                        ;   PRODH:PRODL
   MOVPF    PRODH, RES3 ;
   MOVPF    PRODL, RES2 ;
;
   MOVFP    ARG1L, WREG
   MULWF    ARG2H       ; ARG1L * ARG2H ->
                        ;   PRODH:PRODL
   MOVFP    PRODL, WREG ;
   ADDWF    RES1, F     ; Add cross
   MOVFP    PRODH, WREG ;    products
   ADDWFC   RES2, F     ;
   CLRF     WREG, F     ;
   ADDWFC   RES3, F     ;
;
   MOVFP    ARG1H, WREG ;
   MULWF    ARG2L       ; ARG1H * ARG2L ->
                        ;   PRODH:PRODL
   MOVFP    PRODL, WREG ;
   ADDWF    RES1, F     ; Add cross
   MOVFP    PRODH, WREG ;    products
   ADDWFC   RES2, F     ;
   CLRF     WREG, F     ;
   ADDWFC   RES3, F     ;
;
   BTFSS    ARG2H, 7    ; ARG2H:ARG2L neg?
   GOTO     SIGN_ARG1   ; no, check ARG1
   MOVFP    ARG1L, WREG ;
   SUBWF    RES2        ;
   MOVFP    ARG1H, WREG ;
   SUBWFB   RES3
;
SIGN_ARG1
   BTFSS    ARG1H, 7    ; ARG1H:ARG1L neg?
   GOTO     CONT_CODE   ; no, done
   MOVFP    ARG2L, WREG ;
   SUBWF    RES2        ;
   MOVFP    ARG2H, WREG ;
   SUBWFB   RES3
;
CONT_CODE
     :
```

© 1998-2013 Microchip Technology Inc.

13.1.3.1 PWM Periods

The period of the PWM1 output is determined by Timer1 and its period register (PR1). The period of the PWM2 and PWM3 outputs can be individually software configured to use either Timer1 or Timer2 as the time-base. For PWM2, when TM2PW2 bit (PW2DCL<5>) is clear, the time base is determined by TMR1 and PR1 and when TM2PW2 is set, the time base is determined by Timer2 and PR2. For PWM3, when TM2PW3 bit (PW3DCL<5>) is clear, the time base is determined by TMR1 and PR1, and when TM2PW3 is set, the time base is determined by Timer2 and PR2.

Running two different PWM outputs on two different timers allows different PWM periods. Running all PWMs from Timer1 allows the best use of resources by freeing Timer2 to operate as an 8-bit timer. Timer1 and Timer2 cannot be used as a 16-bit timer if any PWM is being used.

The PWM periods can be calculated as follows:

$$\text{period of PWM1} = [(\text{PR1}) + 1] \times 4T_{OSC}$$

$$\text{period of PWM2} = [(\text{PR1}) + 1] \times 4T_{OSC} \text{ or } [(\text{PR2}) + 1] \times 4T_{OSC}$$

$$\text{period of PWM3} = [(\text{PR1}) + 1] \times 4T_{OSC} \text{ or } [(\text{PR2}) + 1] \times 4T_{OSC}$$

The duty cycle of PWMx is determined by the 10-bit value DCx<9:0>. The upper 8-bits are from register PWxDCH and the lower 2-bits are from PWxDCL<7:6> (PWxDCH:PWxDCL<7:6>). Table 13-4 shows the maximum PWM frequency (FPWM), given the value in the period register.

The number of bits of resolution that the PWM can achieve depends on the operation frequency of the device as well as the PWM frequency (FPWM).

Maximum PWM resolution (bits) for a given PWM frequency:

$$= \frac{\log\left(\frac{F_{OSC}}{F_{PWM}}\right)}{\log(2)} \text{ bits}$$

where: FPWM = 1 / period of PWM

The PWMx duty cycle is as follows:

$$\text{PWMx Duty Cycle} = (\text{DCx}) \times T_{OSC}$$

where DCx represents the 10-bit value from PWxDCH:PWxDCL.

If DCx = 0, then the duty cycle is zero. If PRx = PWxDCH, then the PWM output will be low for one to four Q-clocks (depending on the state of the PWxDCL<7:6> bits). For a duty cycle to be 100%, the PWxDCH value must be greater then the PRx value.

The duty cycle registers for both PWM outputs are double buffered. When the user writes to these registers, they are stored in master latches. When TMR1 (or TMR2) overflows and a new PWM period begins, the master latch values are transferred to the slave latches and the PWMx pin is forced high.

Note: For PW1DCH, PW1DCL, PW2DCH, PW2DCL, PW3DCH and PW3DCL registers, a write operation writes to the "master latches", while a read operation reads the "slave latches". As a result, the user may not read back what was just written to the duty cycle registers (until transferred to slave latch).

The user should also avoid any "read-modify-write" operations on the duty cycle registers, such as: `ADDWF PW1DCH`. This may cause duty cycle outputs that are unpredictable.

TABLE 13-4: PWM FREQUENCY vs. RESOLUTION AT 33 MHz

PWM Frequency	Frequency (kHz)				
	32.2	**64.5**	**90.66**	**128.9**	**515.6**
PRx Value	0xFF	0x7F	0x5A	0x3F	0x0F
High Resolution	10-bit	9-bit	8.5-bit	8-bit	6-bit
Standard Resolution	8-bit	7-bit	6.5-bit	6-bit	4-bit

13.1.3.2 PWM INTERRUPTS

The PWM modules make use of the TMR1 and/or TMR2 interrupts. A timer interrupt is generated when TMR1 or TMR2 equals its period register and on the following increment is cleared to zero. This interrupt also marks the beginning of a PWM cycle. The user can write new duty cycle values before the timer rollover. The TMR1 interrupt is latched into the TMR1IF bit and the TMR2 interrupt is latched into the TMR2IF bit. These flags must be cleared in software.

© 1998-2013 Microchip Technology Inc.

13.2 Timer3

Timer3 is a 16-bit timer consisting of the TMR3H and TMR3L registers. TMR3H is the high byte of the timer and TMR3L is the low byte. This timer has an associated 16-bit period register (PR3H/CA1H:PR3L/CA1L). This period register can be software configured to be a another 16-bit capture register.

When the TMR3CS bit (TCON1<2>) is clear, the timer increments every instruction cycle (FOSC/4). When TMR3CS is set, the counter increments on every falling edge of the RB5/TCLK3 pin. In either mode, the TMR3ON bit must be set for the timer/counter to increment. When TMR3ON is clear, the timer will not increment or set flag bit TMR3IF.

Timer3 has two modes of operation, depending on the CA1/$\overline{\text{PR3}}$ bit (TCON2<3>). These modes are:

- Three capture and one period register mode
- Four capture register mode

The PIC17C7XX has up to four 16-bit capture registers that capture the 16-bit value of TMR3 when events are detected on capture pins. There are four capture pins (RB0/CAP1, RB1/CAP2, RG4/CAP3, and RE3/CAP4), one for each capture register pair. The capture pins are multiplexed with the I/O pins. An event can be:

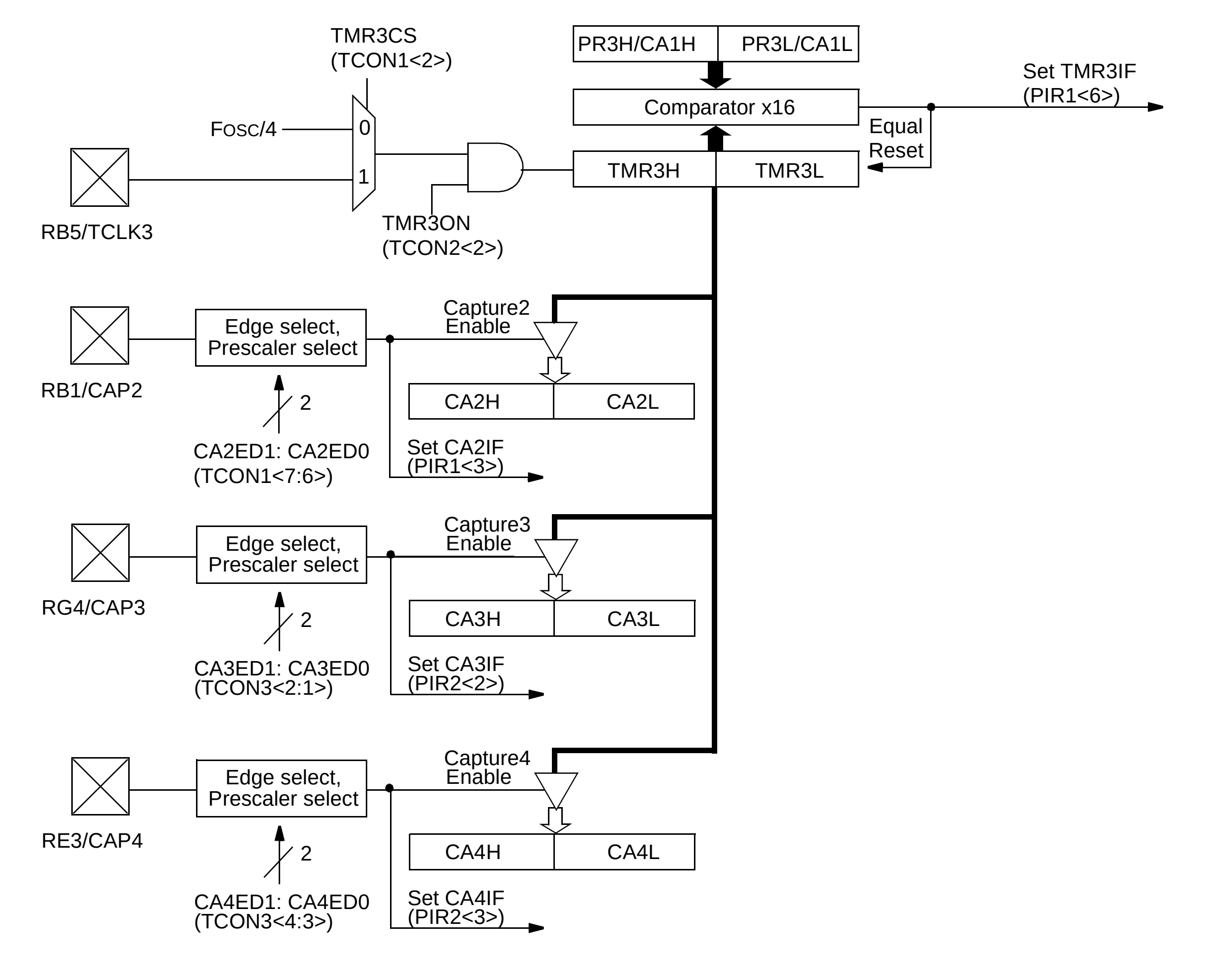
- A rising edge
- A falling edge
- Every 4th rising edge
- Every 16th rising edge

Each 16-bit capture register has an interrupt flag associated with it. The flag is set when a capture is made. The capture modules are truly part of the Timer3 block. Figure 13-5 and Figure 13-6 show the block diagrams for the two modes of operation.

13.2.1 THREE CAPTURE AND ONE PERIOD REGISTER MODE

In this mode, registers PR3H/CA1H and PR3L/CA1L constitute a 16-bit period register. A block diagram is shown in Figure 13-5. The timer increments until it equals the period register and then resets to 0000h on the next timer clock. TMR3 Interrupt Flag bit (TMR3IF) is set at this point. This interrupt can be disabled by clearing the TMR3 Interrupt Enable bit (TMR3IE). TMR3IF must be cleared in software.

FIGURE 13-5: TIMER3 WITH THREE CAPTURE AND ONE PERIOD REGISTER BLOCK DIAGRAM



© 1998-2013 Microchip Technology Inc.

14.2.2 USART ASYNCHRONOUS RECEIVER

The receiver block diagram is shown in Figure 14-2. The data comes in the RX/DT pin and drives the data recovery block. The data recovery block is actually a high speed shifter operating at 16 times the baud rate, whereas the main receive serial shifter operates at the bit rate or at FOSC.

Once Asynchronous mode is selected, reception is enabled by setting bit CREN (RCSTA<4>).

The heart of the receiver is the receive (serial) shift register (RSR). After sampling the STOP bit, the received data in the RSR is transferred to the RCREG (if it is empty). If the transfer is complete, the interrupt bit, RCIF, is set. The actual interrupt can be enabled/disabled by setting/clearing the RCIE bit. RCIF is a read only bit which is cleared by the hardware. It is cleared when RCREG has been read and is empty. RCREG is a double buffered register (i.e., it is a two-deep FIFO). It is possible for two bytes of data to be received and transferred to the RCREG FIFO and a third byte begin shifting to the RSR. On detection of the STOP bit of the third byte, if the RCREG is still full, then the overrun error bit, OERR (RCSTA<1>) will be set. The word in the RSR will be lost. RCREG can be read twice to retrieve the two bytes in the FIFO. The OERR bit has to be cleared in software which is done by resetting the receive logic (CREN is set). If the OERR bit is set, transfers from the RSR to RCREG are inhibited, so it is essential to clear the OERR bit if it is set. The framing error bit FERR (RCSTA<2>) is set if a STOP bit is not detected.

Note:	The FERR and the 9th receive bit are buffered the same way as the receive data. Reading the RCREG register will allow the RX9D and FERR bits to be loaded with values for the next received data. Therefore, it is essential for the user to read the RCSTA register before reading RCREG, in order not to lose the old FERR and RX9D information.

14.2.3 SAMPLING

The data on the RX/DT pin is sampled three times by a majority detect circuit to determine if a high or a low level is present at the RX/DT pin. The sampling is done on the seventh, eighth and ninth falling edges of a x16 clock (Figure 14-5).

The x16 clock is a free running clock and the three sample points occur at a frequency of every 16 falling edges.

FIGURE 14-5: RX PIN SAMPLING SCHEME

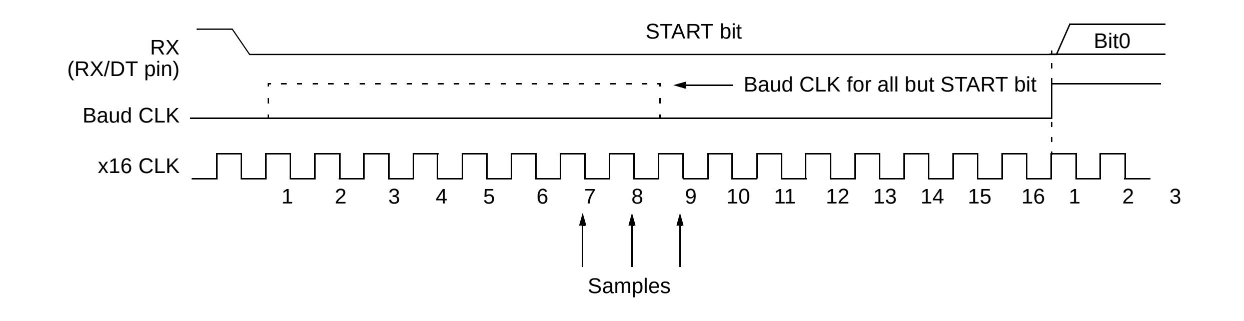
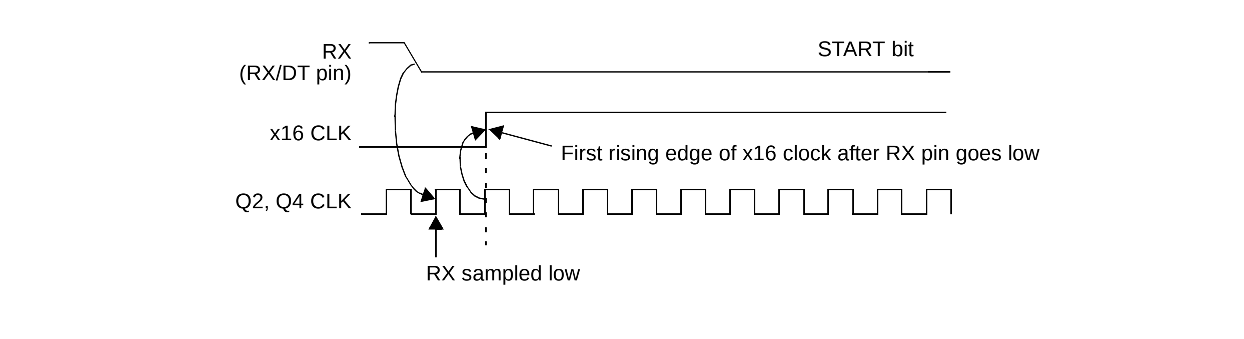


FIGURE 14-6: START BIT DETECT



The SSPSTAT register gives the status of the data transfer. This information includes detection of a START or STOP bit, specifies if the received byte was data or address if the next byte is the completion of 10-bit address and if this will be a read or write data transfer.

The SSPBUF is the register to which transfer data is written to or read from. The SSPSR register shifts the data in or out of the device. In receive operations, the SSPBUF and SSPSR create a doubled buffered receiver. This allows reception of the next byte to begin before reading the last byte of received data. When the complete byte is received, it is transferred to the SSPBUF register and flag bit SSPIF is set. If another complete byte is received before the SSPBUF register is read, a receiver overflow has occurred and bit SSPOV (SSPCON1<6>) is set and the byte in the SSPSR is lost.

The SSPADD register holds the slave address. In 10-bit mode, the user needs to write the high byte of the address (`1111 0 A9 A8 0`). Following the high byte address match, the low byte of the address needs to be loaded (A7:A0).

15.2.1 SLAVE MODE

In Slave mode, the SCL and SDA pins must be configured as inputs. The MSSP module will override the input state with the output data when required (slave-transmitter).

When an address is matched or the data transfer after an address match is received, the hardware automatically will generate the acknowledge ($\overline{\text{ACK}}$) pulse and then load the SSPBUF register with the received value currently in the SSPSR register.

There are certain conditions that will cause the MSSP module not to give this $\overline{\text{ACK}}$ pulse. These are if either (or both):

a) The buffer full bit BF (SSPSTAT<0>) was set before the transfer was received.

b) The overflow bit SSPOV (SSPCON1<6>) was set before the transfer was received.

If the BF bit is set, the SSPSR register value is not loaded into the SSPBUF, but bit SSPIF and SSPOV are set. Table 15-2 shows what happens when a data transfer byte is received, given the status of bits BF and SSPOV. The shaded cells show the condition where user software did not properly clear the overflow condition. Flag bit BF is cleared by reading the SSPBUF register, while bit SSPOV is cleared through software.

The SCL clock input must have a minimum high and low time for proper operation. The high and low times of the I^2C specification, as well as the requirement of the MSSP module, are shown in timing parameter #100 and parameter #101 of the Electrical Specifications.

© 1998-2013 Microchip Technology Inc.

FIGURE 15-21: START CONDITION FLOW CHART

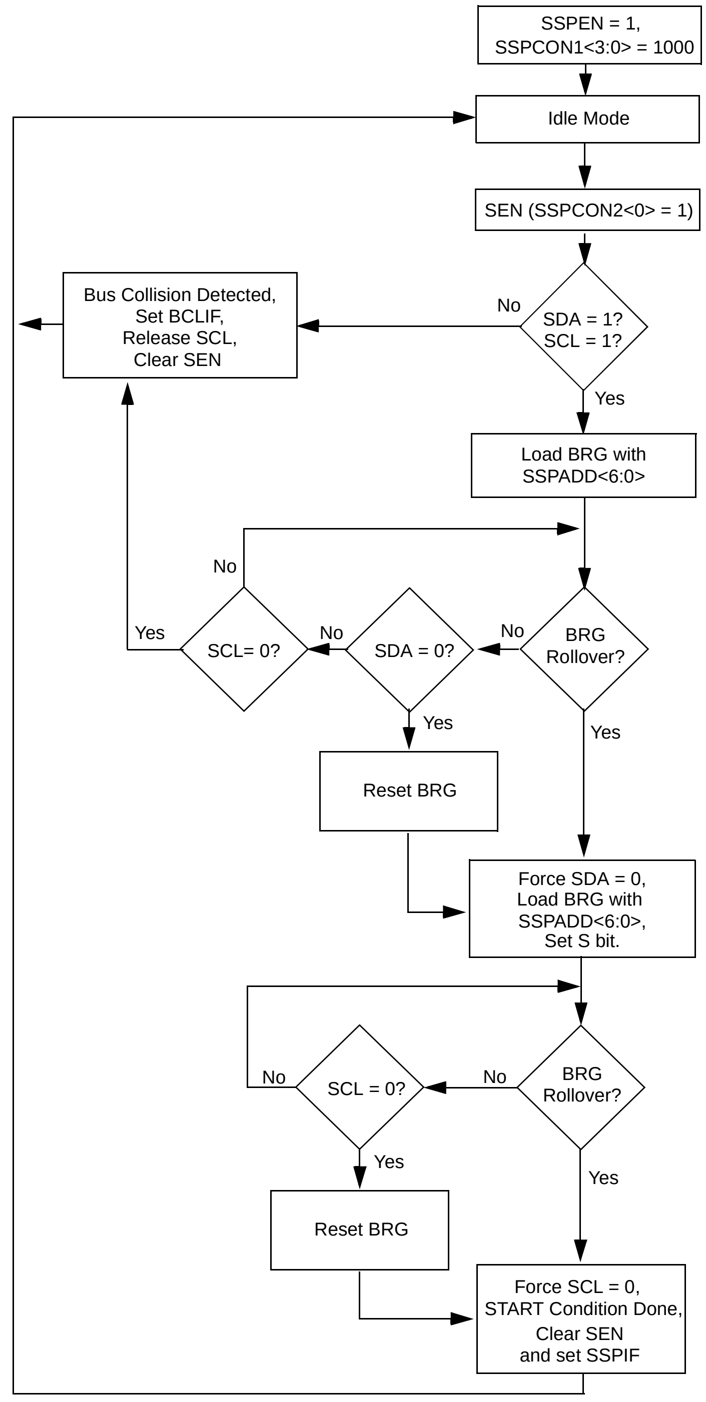
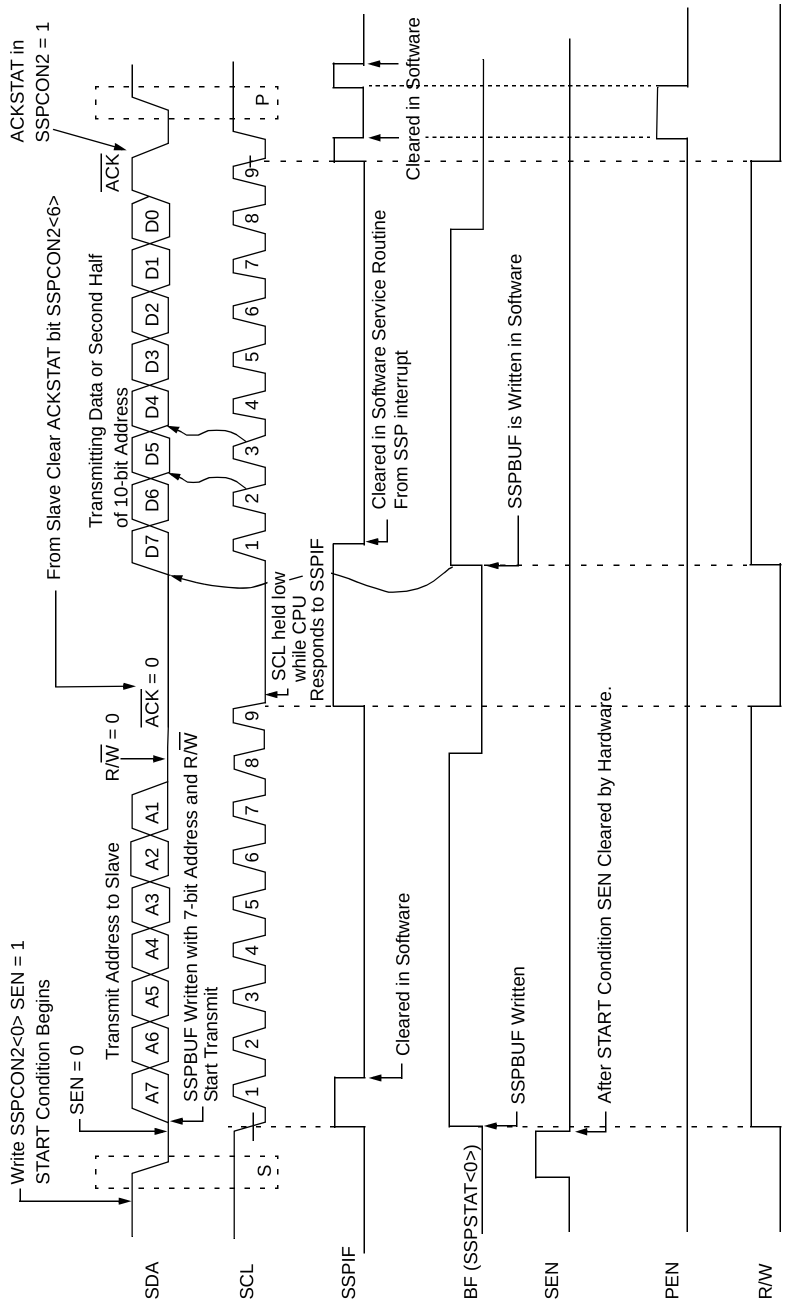


FIGURE 15-26: I^2C MASTER MODE TIMING (TRANSMISSION, 7 OR 10-BIT ADDRESS)



16.3 Configuring Analog Port Pins

The ADCON1, and DDR registers control the operation of the A/D port pins. The port pins that are desired as analog inputs must have their corresponding DDR bits set (input). If the DDR bit is cleared (output), the digital output level (VOH or VOL) will be converted.

The A/D operation is independent of the state of the CHS2:CHS0 bits and the DDR bits.

Note 1: When reading the port register, any pin configured as an analog input channel will read as cleared (a low level). Pins configured as digital inputs, will convert an analog input. Analog levels on a digitally configured input will not affect the conversion accuracy.

2: Analog levels on any pin that is defined as a digital input (including the AN15:AN0 pins), may cause the input buffer to consume current that is out of the devices specification.

16.4 A/D Conversions

Example 16-2 shows how to perform an A/D conversion. The PORTF and lower four PORTG pins are configured as analog inputs. The analog references (VREF+ and VREF-) are the device AVDD and AVSS. The A/D interrupt is enabled, and the A/D conversion clock is FRC. The conversion is performed on the RG3/AN0 pin (channel 0).

Note: The GO/$\overline{\text{DONE}}$ bit should **NOT** be set in the same instruction that turns on the A/D.

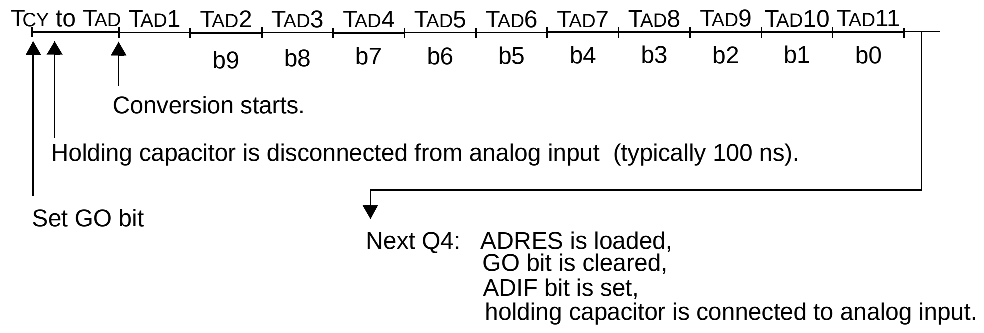
Clearing the GO/$\overline{\text{DONE}}$ bit during a conversion will abort the current conversion. The A/D result register pair will NOT be updated with the partially completed A/D conversion sample. That is, the ADRESH:ADRESL registers will continue to contain the value of the last completed conversion (or the last value written to the ADRESH:ADRESL registers). After the A/D conversion is aborted, a 2TAD wait is required before the next acquisition is started. After this 2TAD wait, acquisition on the selected channel is automatically started.

In Figure 16-4, after the GO bit is set, the first time segment has a minimum of TCY and a maximum of TAD.

EXAMPLE 16-2: A/D CONVERSION

```
     MOVLB   5                  ; Bank 5
     CLRF    ADCON1, F          ; Configure A/D inputs, All analog, TAD = Fosc/8, left just.
     MOVLW   0x01               ; A/D is on, Channel 0 is selected
     MOVWF   ADCON0             ;
     MOVLB   4                  ; Bank 4
     BCF     PIR2, ADIF         ; Clear A/D interrupt flag bit
     BSF     PIE2, ADIE         ; Enable A/D interrupts
     BSF     INTSTA, PEIE       ; Enable peripheral interrupts
     BCF     CPUSTA, GLINTD     ; Enable all interrupts
;
;  Ensure that the required sampling time for the selected input channel has elapsed.
;  Then the conversion may be started.
;
     MOVLB   5                  ; Bank 5
     BSF     ADCON0, GO         ; Start A/D Conversion
       :                        ;   The ADIF bit will be set and the GO/DONE bit
       :                        ;   is cleared upon completion of the A/D Conversion
```

FIGURE 16-4: A/D CONVERSION TAD CYCLES


TCY to TAD
TAD1
TAD2
TAD3
TAD4
TAD5
TAD6
TAD7
TAD8
TAD9
TAD10
TAD11
b9
b8
b7
b6
b5
b4
b3
b2
b1
b0
Conversion starts.
Holding capacitor is disconnected from analog input (typically 100 ns).
Set GO bit
Next Q4: ADRES is loaded,
GO bit is cleared,
ADIF bit is set,
holding capacitor is connected to analog input.

© 1998-2013 Microchip Technology Inc.

FIGURE 16-5: FLOW CHART OF A/D OPERATION

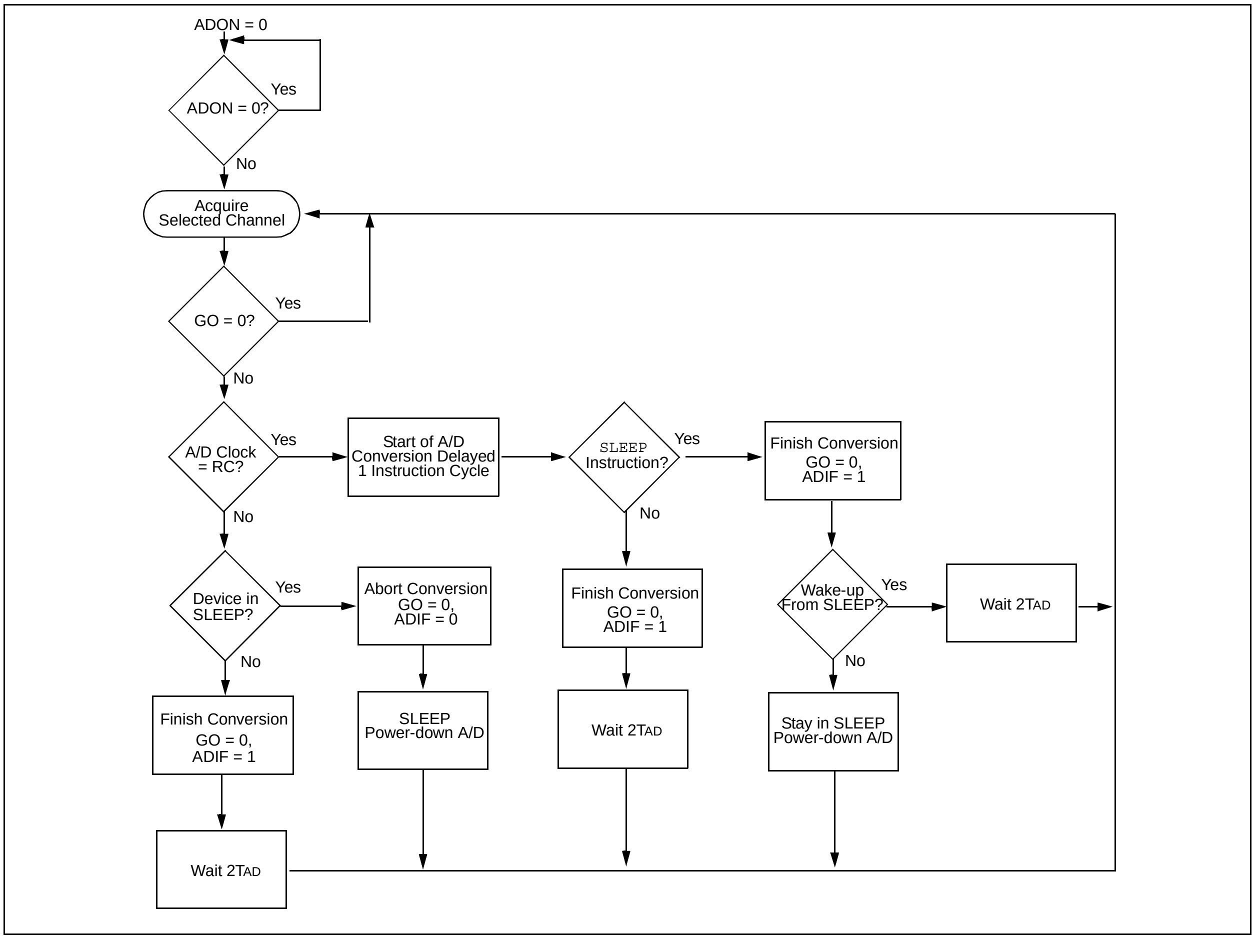

ADON = 0
ADON = 0?
Yes
No
Acquire Selected Channel
GO = 0?
Yes
No
A/D Clock = RC?
Yes
No
Start of A/D Conversion Delayed 1 Instruction Cycle
SLEEP Instruction?
Yes
No
Finish Conversion GO = 0, ADIF = 1
Device in SLEEP?
Yes
No
Abort Conversion GO = 0, ADIF = 0
Finish Conversion GO = 0, ADIF = 1
Wake-up From SLEEP?
Yes
No
Wait 2TAD
Finish Conversion GO = 0, ADIF = 1
SLEEP Power-down A/D
Wait 2TAD
Stay in SLEEP Power-down A/D
Wait 2TAD

TABLE 18-2: PIC17CXXX INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	16-bit Opcode				Status Affected	Notes
				MSb			LSb		
BYTE-ORIENTED FILE REGISTER OPERATIONS									
ADDWF	**f,d**	ADD WREG to f	1	0000	111d	ffff	ffff	OV,C,DC,Z	
ADDWFC	**f,d**	ADD WREG and Carry bit to f	1	0001	000d	ffff	ffff	OV,C,DC,Z	
ANDWF	**f,d**	AND WREG with f	1	0000	101d	ffff	ffff	Z	
CLRF	**f,s**	Clear f, or Clear f and Clear WREG	1	0010	100s	ffff	ffff	None	3
COMF	**f,d**	Complement f	1	0001	001d	ffff	ffff	Z	
CPFSEQ	**f**	Compare f with WREG, skip if f = WREG	1 (2)	0011	0001	ffff	ffff	None	6,8
CPFSGT	**f**	Compare f with WREG, skip if f > WREG	1 (2)	0011	0010	ffff	ffff	None	2,6,8
CPFSLT	**f**	Compare f with WREG, skip if f < WREG	1 (2)	0011	0000	ffff	ffff	None	2,6,8
DAW	**f,s**	Decimal Adjust WREG Register	1	0010	111s	ffff	ffff	C	3
DECF	**f,d**	Decrement f	1	0000	011d	ffff	ffff	OV,C,DC,Z	
DECFSZ	**f,d**	Decrement f, skip if 0	1 (2)	0001	011d	ffff	ffff	None	6,8
DCFSNZ	**f,d**	Decrement f, skip if not 0	1 (2)	0010	011d	ffff	ffff	None	6,8
INCF	**f,d**	Increment f	1	0001	010d	ffff	ffff	OV,C,DC,Z	
INCFSZ	**f,d**	Increment f, skip if 0	1 (2)	0001	111d	ffff	ffff	None	6,8
INFSNZ	**f,d**	Increment f, skip if not 0	1 (2)	0010	010d	ffff	ffff	None	6,8
IORWF	**f,d**	Inclusive OR WREG with f	1	0000	100d	ffff	ffff	Z	
MOVFP	**f,p**	Move f to p	1	011p	pppp	ffff	ffff	None	
MOVPF	**p,f**	Move p to f	1	010p	pppp	ffff	ffff	Z	
MOVWF	**f**	Move WREG to f	1	0000	0001	ffff	ffff	None	
MULWF	**f**	Multiply WREG with f	1	0011	0100	ffff	ffff	None	
NEGW	**f,s**	Negate WREG	1	0010	110s	ffff	ffff	OV,C,DC,Z	1,3
NOP	**—**	No Operation	1	0000	0000	0000	0000	None	
RLCF	**f,d**	Rotate left f through Carry	1	0001	101d	ffff	ffff	C	
RLNCF	**f,d**	Rotate left f (no carry)	1	0010	001d	ffff	ffff	None	
RRCF	**f,d**	Rotate right f through Carry	1	0001	100d	ffff	ffff	C	
RRNCF	**f,d**	Rotate right f (no carry)	1	0010	000d	ffff	ffff	None	
SETF	**f,s**	Set f	1	0010	101s	ffff	ffff	None	3
SUBWF	**f,d**	Subtract WREG from f	1	0000	010d	ffff	ffff	OV,C,DC,Z	1
SUBWFB	**f,d**	Subtract WREG from f with Borrow	1	0000	001d	ffff	ffff	OV,C,DC,Z	1
SWAPF	**f,d**	Swap f	1	0001	110d	ffff	ffff	None	
TABLRD	**t,i,f**	Table Read	2 (3)	1010	10ti	ffff	ffff	None	7
TABLWT	**t,i,f**	Table Write	2	1010	11ti	ffff	ffff	None	5
TLRD	**t,f**	Table Latch Read	1	1010	00tx	ffff	ffff	None	
TLWT	**t,f**	Table Latch Write	1	1010	01tx	ffff	ffff	None	

Legend: Refer to Table 18-1 for opcode field descriptions.

Note 1: 2’s Complement method.

2: Unsigned arithmetic.

3: If s = '1', only the file is affected: If s = '0', both the WREG register and the file are affected; If only the Working register (WREG) is required to be affected, then f = WREG must be specified.

4: During an LCALL, the contents of PCLATH are loaded into the MSB of the PC and kkkk kkkk is loaded into the LSB of the PC (PCL).

5: Multiple cycle instruction for EPROM programming when table pointer selects internal EPROM. The instruction is terminated by an interrupt event. When writing to external program memory, it is a two-cycle instruction.

6: Two-cycle instruction when condition is true, else single cycle instruction.

7: Two-cycle instruction except for TABLRD to PCL (program counter low byte), in which case it takes 3 cycles.

8: A “skip” means that instruction fetched during execution of current instruction is not executed, instead a NOP is executed.

© 1998-2013 Microchip Technology Inc.

ADDWFC ADD WREG and Carry bit to f

Syntax:	[*label*] ADDWFC f,d
Operands:	0 ≤ f ≤ 255 d ∈ [0,1]
Operation:	(WREG) + (f) + C → (dest)
Status Affected:	OV, C, DC, Z
Encoding:	`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:	[*label*] ANDLW k
Operands:	0 ≤ k ≤ 255
Operation:	(WREG) .AND. (k) → (WREG)
Status Affected:	Z
Encoding:	`1011` `0101` `kkkk` `kkkk`
Description:	The contents of WREG are AND'ed 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

© 1998-2013 Microchip Technology Inc.

ANDWF AND WREG with f

Syntax: [*label*] ANDWF f,d

Operands: 0 ≤ f ≤ 255
d ∈ [0,1]

Operation: (WREG) .AND. (f) → (dest)

Status Affected: Z

Encoding:

`0000`	`101d`	`ffff`	`ffff`

Description: The contents of WREG are AND'ed with register 'f'. If 'd' is 0 the result is stored in WREG. If 'd' is 1 the result is stored 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: `ANDWF REG, 1`

Before Instruction

WREG	=	0x17
REG	=	0xC2

After Instruction

WREG	=	0x17
REG	=	0x02

BCF Bit Clear f

Syntax: [*label*] BCF f,b

Operands: 0 ≤ f ≤ 255
0 ≤ b ≤ 7

Operation: 0 → (f<b>)

Status Affected: None

Encoding:

`1000`	`1bbb`	`ffff`	`ffff`

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

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write register 'f'

Example: `BCF FLAG_REG, 7`

Before Instruction

FLAG_REG = 0xC7

After Instruction

FLAG_REG = 0x47

TABLRD Table Read

Example1: `TABLRD 1, 1, REG ;`

Before Instruction

REG	=	0x53
TBLATH	=	0xAA
TBLATL	=	0x55
TBLPTR	=	0xA356
MEMORY(TBLPTR)	=	0x1234

After Instruction (table write completion)

REG	=	0xAA
TBLATH	=	0x12
TBLATL	=	0x34
TBLPTR	=	0xA357
MEMORY(TBLPTR)	=	0x5678

Example2: `TABLRD 0, 0, REG ;`

Before Instruction

REG	=	0x53
TBLATH	=	0xAA
TBLATL	=	0x55
TBLPTR	=	0xA356
MEMORY(TBLPTR)	=	0x1234

After Instruction (table write completion)

REG	=	0x55
TBLATH	=	0x12
TBLATL	=	0x34
TBLPTR	=	0xA356
MEMORY(TBLPTR)	=	0x1234

TABLWT Table Write

Syntax: [*label*] TABLWT t,i,f

Operands:
$0 \le f \le 255$
$i \in [0,1]$
$t \in [0,1]$

Operation:
If t = 0,
f → TBLATL;
If t = 1,
f → TBLATH;
TBLAT → Prog Mem (TBLPTR);
If i = 1,
TBLPTR + 1 → TBLPTR
If i = 0,
TBLPTR is unchanged

Status Affected: None

Encoding:

1010	11ti	ffff	ffff

Description:

1. Load value in ’f’ into 16-bit table latch (TBLAT)
   If t = 1: load into high byte;
   If t = 0: load into low byte
2. The contents of TBLAT are written to the program memory location pointed to by TBLPTR.
   If TBLPTR points to external program memory location, then the instruction takes two-cycle.
   If TBLPTR points to an internal EPROM location, then the instruction is terminated when an interrupt is received.

> **Note:** The $\overline{\text{MCLR}}$/VPP pin must be at the programming voltage for successful programming of internal memory.
> If $\overline{\text{MCLR}}$/VPP = VDD
> the programming sequence of internal memory will be interrupted. A short write will occur (2 TCY). The internal memory location will not be affected.

3. The TBLPTR can be automatically incremented
   If i = 1; TBLPTR is not incremented
   If i = 0; TBLPTR is incremented

Words: 1

Cycles: 2 (many if write is to on-chip EPROM program memory)

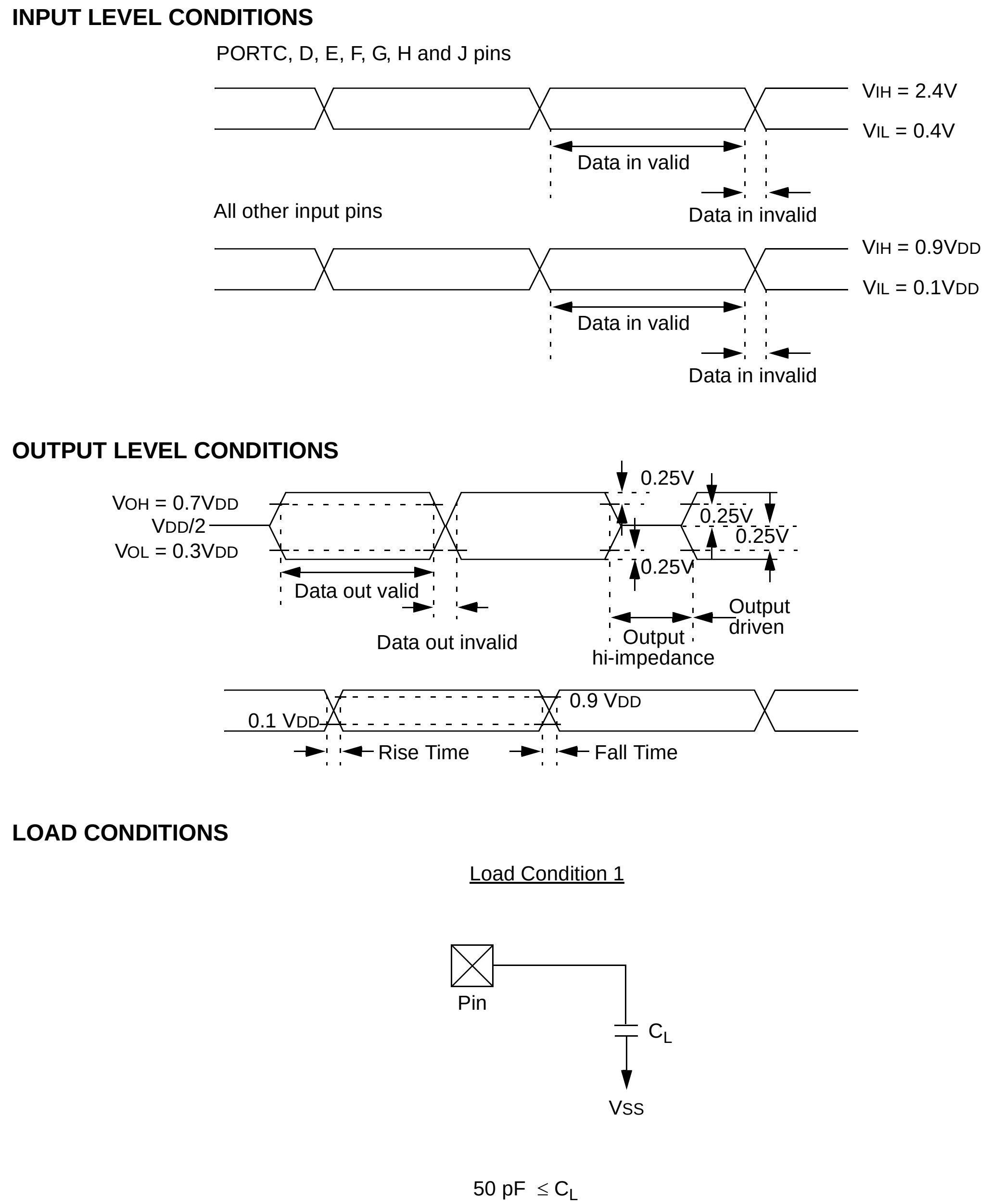
Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write register TBLATH or TBLATL
No operation	No operation (Table Pointer on Address bus)	No operation	No operation (Table Latch on Address bus, $\overline{\text{WR}}$ goes low)

© 1998-2013 Microchip Technology Inc.

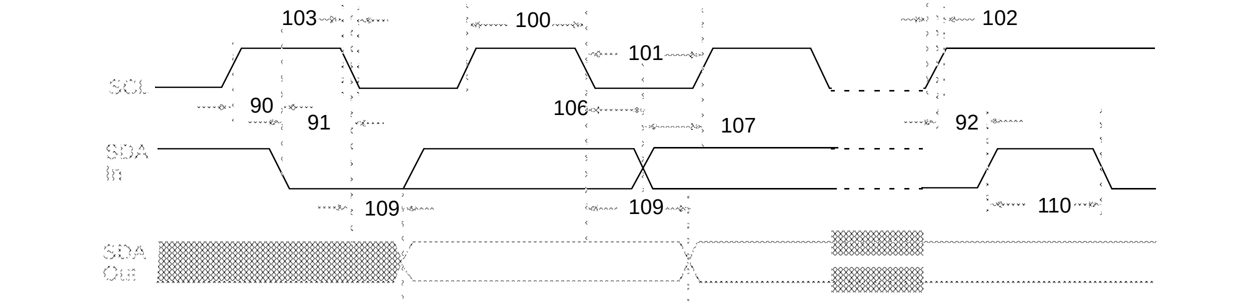
FIGURE 20-5: PARAMETER MEASUREMENT INFORMATION

All timings are measured between high and low measurement points as indicated below.



© 1998-2013 Microchip Technology Inc.

FIGURE 20-18: I²C BUS DATA TIMING



Note: Refer to Figure 20-5 for load conditions.

TABLE 20-13: I²C BUS DATA REQUIREMENTS

Param No.	Sym	Characteristic		Min	Max	Units	Conditions
100	Thigh	Clock high time	100 kHz mode	2(Tosc)(BRG + 1)	—	ms	
			400 kHz mode	2(Tosc)(BRG + 1)	—	ms	
			1 MHz mode(1)	2(Tosc)(BRG + 1)	—	ms	
101	Tlow	Clock low time	100 kHz mode	2(Tosc)(BRG + 1)	—	ms	
			400 kHz mode	2(Tosc)(BRG + 1)	—	ms	
			1 MHz mode(1)	2(Tosc)(BRG + 1)	—	ms	
102	Tr	SDA and SCL rise time	100 kHz mode	—	1000	ns	Cb is specified to be from 10 to 400 pF
			400 kHz mode	20 + 0.1Cb	300	ns	
			1 MHz mode(1)	—	300	ns	
103	Tf	SDA and SCL fall time	100 kHz mode	—	300	ns	Cb is specified to be from 10 to 400 pF
			400 kHz mode	20 + 0.1Cb	300	ns	
			1 MHz mode(1)	—	10	ns	
90	Tsu:sta	START condition setup time	100 kHz mode	2(Tosc)(BRG + 1)	—	ms	Only relevant for Repeated Start condition
			400 kHz mode	2(Tosc)(BRG + 1)	—	ms	
			1 MHz mode(1)	2(Tosc)(BRG + 1)	—	ms	
91	Thd:sta	START condition hold time	100 kHz mode	2(Tosc)(BRG + 1)	—	ms	After this period, the first clock pulse is generated
			400 kHz mode	2(Tosc)(BRG + 1)	—	ms	
			1 MHz mode(1)	2(Tosc)(BRG + 1)	—	ms	
106	Thd:dat	Data input hold time	100 kHz mode	0	—	ns	
			400 kHz mode	0	0.9	ms	
			1 MHz mode(1)	0	—	ns	
107	Tsu:dat	Data input setup time	100 kHz mode	250	—	ns	**(Note 2)**
			400 kHz mode	100	—	ns	
			1 MHz mode(1)	100	—	ns	
92	Tsu:sto	STOP condition setup time	100 kHz mode	2(Tosc)(BRG + 1)	—	ms	
			400 kHz mode	2(Tosc)(BRG + 1)	—	ms	
			1 MHz mode(1)	2(Tosc)(BRG + 1)	—	ms	
109	Taa	Output valid from clock	100 kHz mode	—	3500	ns	
			400 kHz mode	—	1000	ns	
			1 MHz mode(1)	—	400	ns	

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

2: A fast mode (400 KHz) I²C bus device can be used in a standard mode I²C bus system, but the parameter # 107 ≥ 250 ns must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCL signal. If such a device does stretch the LOW period of the SCL signal, it must output the next data bit to the SDA line. Parameter #102 + #107 = 1000 + 250 = 1250 ns (for 100 kHz mode) before the SCL line is released.

3: C_b is specified to be from 10-400pF. The minimum specifications are characterized with C_b=10pF. The rise time spec (t_r) is characterized with R_p=R_p min. The minimum fall time specification (t_f) is characterized with C_b=10pF,and R_p=R_p max. These are only valid for fast mode operation (VDD=4.5-5.5V) and where the SPM bit (SSPSTAT<7>) =1.)

4: Max specifications for these parameters are valid for falling edge only. Specs are characterized with R_p=R_p min and C_b=400pF for standard mode, 200pF for fast mode, and 10pF for 1MHz mode.

© 1998-2013 Microchip Technology Inc.

FIGURE 21-15: TYPICAL, MINIMUM AND MAXIMUM WDT PERIOD vs. VDD (-40°C TO +125°C)

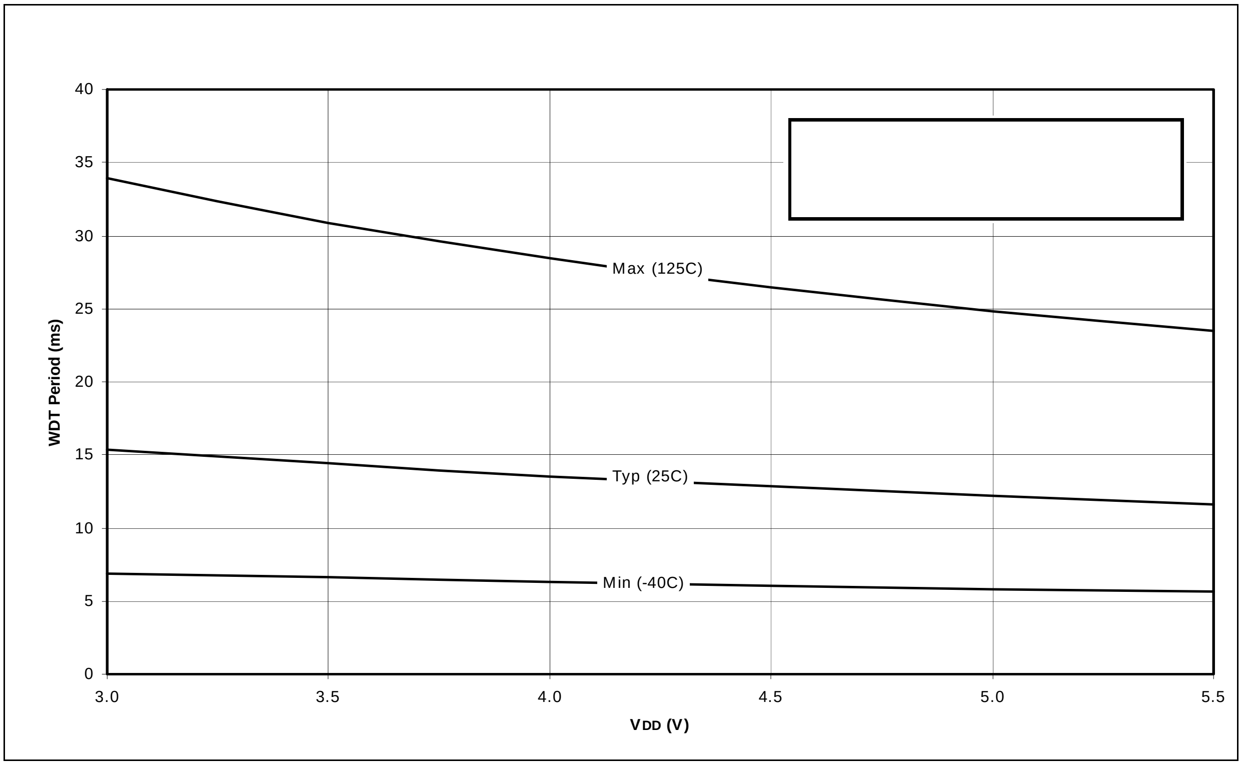
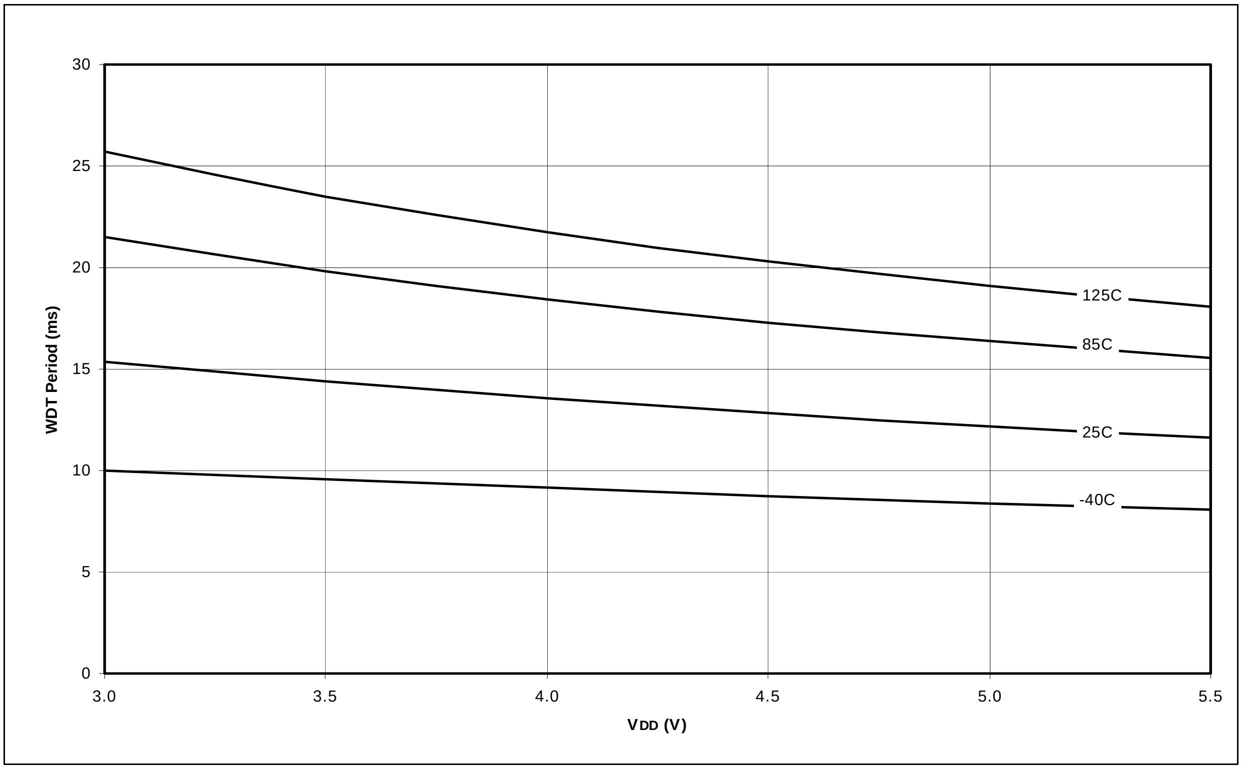


FIGURE 21-16: TYPICAL WDT PERIOD vs. VDD OVER TEMPERATURE (-40°C TO +125°C)



© 1998-2013 Microchip Technology Inc.