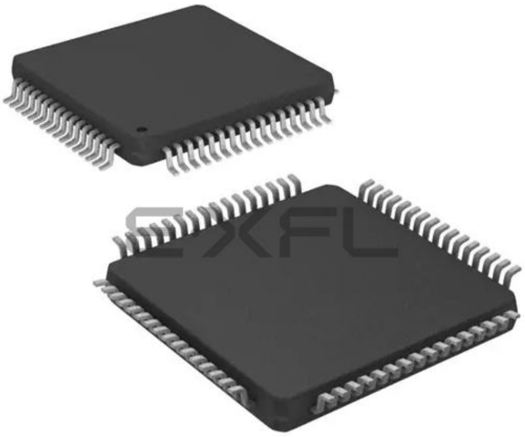
Microchip Technology - PIC17C756AT-16I/PT Datasheet



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	Obsolete
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	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic17c756at-16i-pt

5.0 RESET

The PIC17CXXX differentiates between various kinds of RESET:

- Power-on Reset (POR)
- Brown-out Reset
- $\overline{\text{MCLR}}$ Reset
- WDT Reset

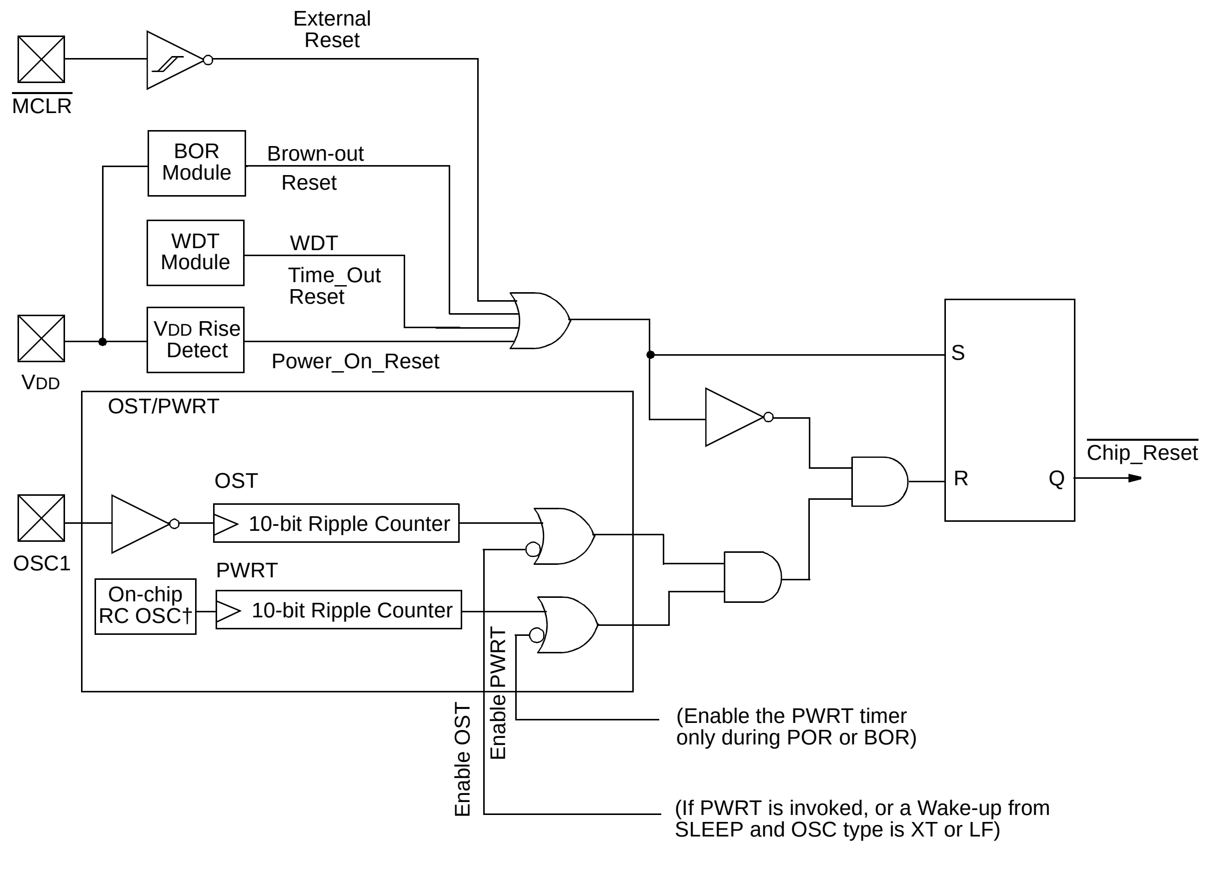
Some registers are not affected in any RESET condition, their status is unknown on POR and unchanged in any other RESET. Most other registers are forced to a "RESET state". The $\overline{\text{TO}}$ and $\overline{\text{PD}}$ bits are set or cleared differently in different RESET situations, as indicated in Table 5-3. These bits, in conjunction with the $\overline{\text{POR}}$ and $\overline{\text{BOR}}$ bits, are used in software to determine the nature of the RESET. See Table 5-4 for a full description of the RESET states of all registers.

When the device enters the "RESET state", the Data Direction registers (DDR) are forced set, which will make the I/O hi-impedance inputs. The RESET state of some peripheral modules may force the I/O to other operations, such as analog inputs or the system bus.

Note:	While the device is in a RESET state, the internal phase clock is held in the Q1 state. Any processor mode that allows external execution will force the RE0/ALE pin as a low output and the RE1/$\overline{\text{OE}}$ and RE2/$\overline{\text{WR}}$ pins as high outputs.

A simplified block diagram of the On-Chip Reset Circuit is shown in Figure 5-1.

FIGURE 5-1: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT



† This RC oscillator is shared with the WDT when not in a power-up sequence.

5.1 Power-on Reset (POR), Power-up Timer (PWRT), Oscillator Start-up Timer (OST) and Brown-out Reset (BOR)

5.1.1 POWER-ON RESET (POR)

The Power-on Reset circuit holds the device in RESET until VDD is above the trip point (in the range of 1.4V - 2.3V). The devices produce an internal RESET for both rising and falling VDD. To take advantage of the POR, just tie the MCLR/VPP pin directly (or through a resistor) to VDD. This will eliminate external RC components usually needed to create Power-on Reset. A minimum rise time for VDD is required. See Electrical Specifications for details.

Figure 5-2 and Figure 5-3 show two possible POR circuits.

FIGURE 5-2: USING ON-CHIP POR

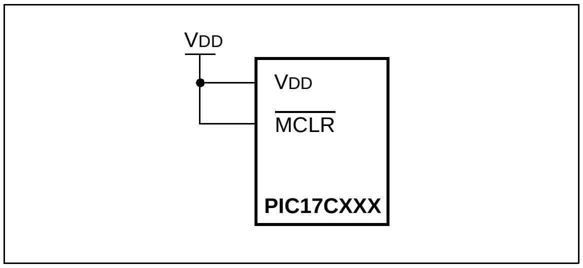
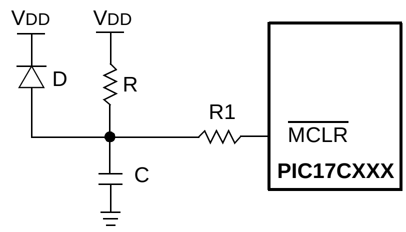


FIGURE 5-3: EXTERNAL POWER-ON RESET CIRCUIT (FOR SLOW VDD POWER-UP)



Note 1: An external Power-on Reset circuit is required only if VDD power-up time is too slow. The diode D helps discharge the capacitor quickly when VDD powers down.

2: R < 40 kΩ is recommended to ensure that the voltage drop across R does not exceed 0.2V (max. leakage current spec. on the MCLR/VPP pin is 5 μA). A larger voltage drop will degrade VIH level on the MCLR/VPP pin.

3: R1 = 100Ω to 1 kΩ will limit any current flowing into MCLR from external capacitor C in the event of MCLR/VPP pin breakdown due to Electrostatic Discharge (ESD) or Electrical Overstress (EOS).

5.1.2 POWER-UP TIMER (PWRT)

The Power-up Timer provides a fixed 96 ms time-out (nominal) on power-up. This occurs from the rising edge of the internal POR signal if VDD and MCLR are tied, or after the first rising edge of MCLR (detected high). The Power-up Timer operates on an internal RC oscillator. The chip is kept in RESET as long as the PWRT is active. In most cases, the PWRT delay allows VDD to rise to an acceptable level.

The power-up time delay will vary from chip to chip and with VDD and temperature. See DC parameters for details.

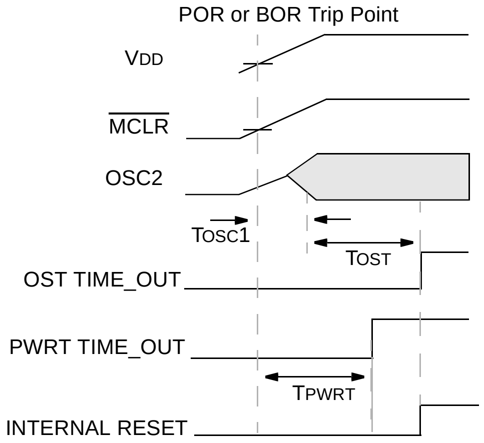
5.1.3 OSCILLATOR START-UP TIMER (OST)

The Oscillator Start-up Timer (OST) provides a 1024 oscillator cycle (1024TOSC) delay whenever the PWRT is invoked, or a wake-up from SLEEP event occurs in XT or LF mode. The PWRT and OST operate in parallel.

The OST counts the oscillator pulses on the OSC1/CLKIN pin. The counter only starts incrementing after the amplitude of the signal reaches the oscillator input thresholds. This delay allows the crystal oscillator or resonator to stabilize before the device exits RESET. The length of the time-out is a function of the crystal/resonator frequency.

Figure 5-4 shows the operation of the OST circuit. In this figure, the oscillator is of such a low frequency that although enabled simultaneously, the OST does not time-out until after the Power-up Timer time-out.

FIGURE 5-4: OSCILLATOR START-UP TIME (LOW FREQUENCY)



This figure shows in greater detail the timings involved with the oscillator start-up timer. In this example, the low frequency crystal start-up time is larger than power-up time (TPWRT).

TOSC1 = time for the crystal oscillator to react to an oscillation level detectable by the Oscillator Start-up Timer (OST).

TOST = 1024TOSC.

6.1 Interrupt Status Register (INTSTA)

The Interrupt Status/Control register (INTSTA) contains the flag and enable bits for non-peripheral interrupts.

The PEIF bit is a read only, bit wise OR of all the peripheral flag bits in the PIR registers (Figure 6-4 and Figure 6-5).

Note:	All interrupt flag bits get set by their specified condition, even if the corresponding interrupt enable bit is clear (interrupt disabled), or the GLINTD bit is set (all interrupts disabled).

Care should be taken when clearing any of the INTSTA register enable bits when interrupts are enabled (GLINTD is clear). If any of the INTSTA flag bits (T0IF, INTF, T0CKIF, or PEIF) are set in the same instruction cycle as the corresponding interrupt enable bit is cleared, the device will vector to the RESET address (0x00).

Prior to disabling any of the INTSTA enable bits, the GLINTD bit should be set (disabled).

REGISTER 6-1: INTSTA REGISTER (ADDRESS: 07h, UNBANKED)

R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PEIF	T0CKIF	T0IF	INTF	PEIE	T0CKIE	T0IE	INTE
bit 7							bit 0

bit 7 **PEIF**: Peripheral Interrupt Flag bit
This bit is the OR of all peripheral interrupt flag bits AND'ed with their corresponding enable bits. The interrupt logic forces program execution to address (20h) when a peripheral interrupt is pending.
1 = A peripheral interrupt is pending
0 = No peripheral interrupt is pending

bit 6 **T0CKIF**: External Interrupt on T0CKI Pin Flag bit
This bit is cleared by hardware, when the interrupt logic forces program execution to address (18h).
1 = The software specified edge occurred on the RA1/T0CKI pin
0 = The software specified edge did not occur on the RA1/T0CKI pin

bit 5 **T0IF**: TMR0 Overflow Interrupt Flag bit
This bit is cleared by hardware, when the interrupt logic forces program execution to address (10h).
1 = TMR0 overflowed
0 = TMR0 did not overflow

bit 4 **INTF**: External Interrupt on INT Pin Flag bit
This bit is cleared by hardware, when the interrupt logic forces program execution to address (08h).
1 = The software specified edge occurred on the RA0/INT pin
0 = The software specified edge did not occur on the RA0/INT pin

bit 3 **PEIE**: Peripheral Interrupt Enable bit
This bit acts as a global enable bit for the peripheral interrupts that have their corresponding enable bits set.
1 = Enable peripheral interrupts
0 = Disable peripheral interrupts

bit 2 **T0CKIE**: External Interrupt on T0CKI Pin Enable bit
1 = Enable software specified edge interrupt on the RA1/T0CKI pin
0 = Disable interrupt on the RA1/T0CKI pin

bit 1 **T0IE**: TMR0 Overflow Interrupt Enable bit
1 = Enable TMR0 overflow interrupt
0 = Disable TMR0 overflow interrupt

bit 0 **INTE**: External Interrupt on RA0/INT Pin Enable bit
1 = Enable software specified edge interrupt on the RA0/INT pin
0 = Disable software specified edge interrupt on the RA0/INT pin

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
- n = Value at POR Reset	’1’ = Bit is set	’0’ = Bit is cleared	x = Bit is unknown

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 $\overline{\text{MCLR}}$/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 `TABLWT` 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).

NOTES:

13.1 Timer1 and Timer2

13.1.1 TIMER1, TIMER2 IN 8-BIT MODE

Both Timer1 and Timer2 will operate in 8-bit mode when the T16 bit is clear. These two timers can be independently configured to increment from the internal instruction cycle clock (TCY), or from an external clock source on the RB4/TCLK12 pin. The timer clock source is configured by the TMRxCS bit (x = 1 for Timer1, or = 2 for Timer2). When TMRxCS is clear, the clock source is internal and increments once every instruction cycle (FOSC/4). When TMRxCS is set, the clock source is the RB4/TCLK12 pin and the counters will increment on every falling edge of the RB4/TCLK12 pin.

The timer increments from 00h until it equals the Period register (PRx). It then resets to 00h at the next increment cycle. The timer interrupt flag is set when the timer is reset. TMR1 and TMR2 have individual interrupt flag bits. The TMR1 interrupt flag bit is latched into TMR1IF and the TMR2 interrupt flag bit is latched into TMR2IF.

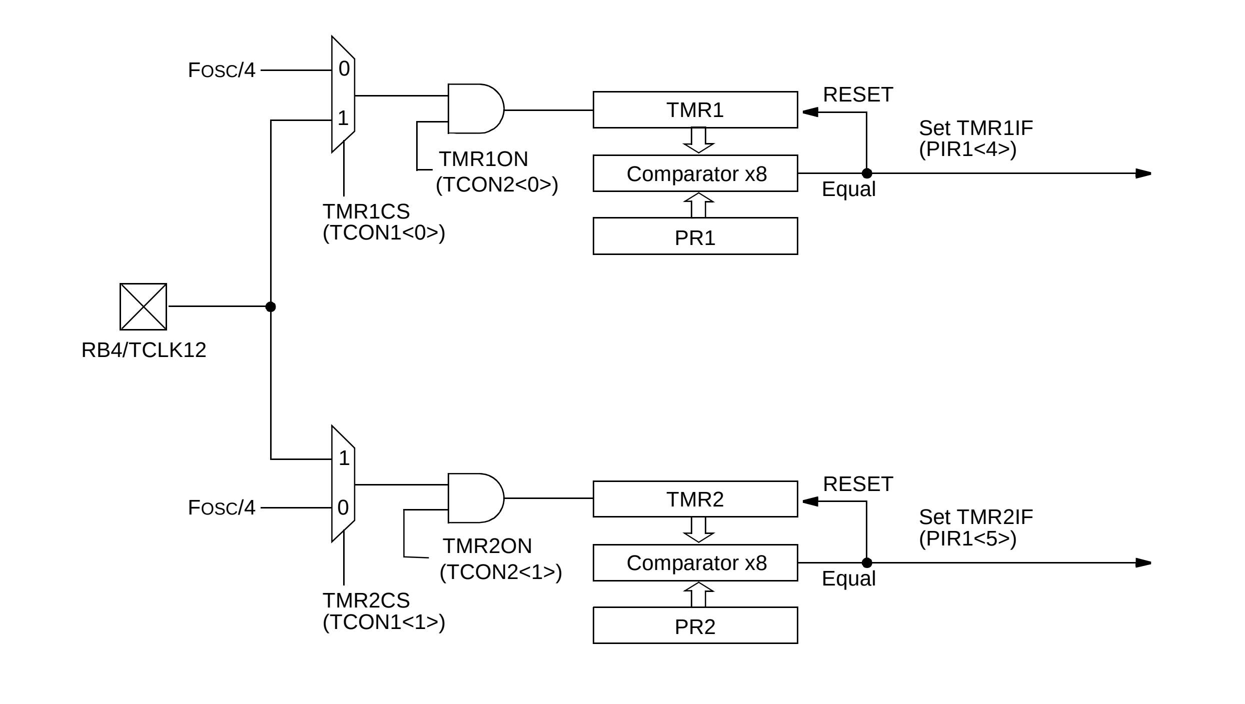
Each timer also has a corresponding interrupt enable bit (TMRxIE). The timer interrupt can be enabled/ disabled by setting/clearing this bit. For peripheral interrupts to be enabled, the Peripheral Interrupt Enable bit must be set (PEIE = '1') and global interrupt must be enabled (GLINTD = '0').

The timers can be turned on and off under software control. When the timer on control bit (TMRxON) is set, the timer increments from the clock source. When TMRxON is cleared, the timer is turned off and cannot cause the timer interrupt flag to be set.

13.1.1.1 External Clock Input for Timer1 and Timer2

When TMRxCS is set, the clock source is the RB4/ TCLK12 pin, and the counter will increment on every falling edge on the RB4/TCLK12 pin. The TCLK12 input is synchronized with internal phase clocks. This causes a delay from the time a falling edge appears on TCLK12 to the time TMR1 or TMR2 is actually incremented. For the external clock input timing requirements, see the Electrical Specification section.

FIGURE 13-1: TIMER1 AND TIMER2 IN TWO 8-BIT TIMER/COUNTER MODE



13.1.3.3 External Clock Source

The PWMs will operate, regardless of the clock source of the timer. The use of an external clock has ramifications that must be understood. Because the external TCLK12 input is synchronized internally (sampled once per instruction cycle), the time TCLK12 changes to the time the timer increments, will vary by as much as 1TCY (one instruction cycle). This will cause jitter in the duty cycle as well as the period of the PWM output.

This jitter will be ±1TCY, unless the external clock is synchronized with the processor clock. Use of one of the PWM outputs as the clock source to the TCLK12 input, will supply a synchronized clock.

In general, when using an external clock source for PWM, its frequency should be much less than the device frequency (FOSC).

13.1.3.4 Maximum Resolution/Frequency for External Clock Input

The use of an external clock for the PWM time base (Timer1 or Timer2) limits the PWM output to a maximum resolution of 8-bits. The PWxDCL<7:6> bits must be kept cleared. Use of any other value will distort the PWM output. All resolutions are supported when internal clock mode is selected. The maximum attainable frequency is also lower. This is a result of the timing requirements of an external clock input for a timer (see the Electrical Specification section). The maximum PWM frequency, when the timers clock source is the RB4/TCLK12 pin, is shown in Table 13-4 (Standard Resolution mode).

TABLE 13-5: REGISTERS/BITS ASSOCIATED WITH PWM

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	$\overline{\text{MCLR}}$, WDT
16h, Bank 3	TCON1	CA2ED1	CA2ED0	CA1ED1	CA1ED0	T16	TMR3CS	TMR2CS	TMR1CS	0000 0000	0000 0000
17h, Bank 3	TCON2	CA2OVF	CA1OVF	PWM2ON	PWM1ON	CA1/$\overline{\text{PR3}}$	TMR3ON	TMR2ON	TMR1ON	0000 0000	0000 0000
16h, Bank 7	TCON3	—	CA4OVF	CA3OVF	CA4ED1	CA4ED0	CA3ED1	CA3ED0	PWM3ON	-000 0000	-000 0000
10h, Bank 2	TMR1	Timer1's Register								xxxx xxxx	uuuu uuuu
11h, Bank 2	TMR2	Timer2's Register								xxxx xxxx	uuuu uuuu
16h, Bank 1	PIR1	RBIF	TMR3IF	TMR2IF	TMR1IF	CA2IF	CA1IF	TX1IF	RC1IF	x000 0010	u000 0010
17h, Bank 1	PIE1	RBIE	TMR3IE	TMR2IE	TMR1IE	CA2IE	CA1IE	TX1IE	RC1IE	0000 0000	0000 0000
07h, Unbanked	INTSTA	PEIF	T0CKIF	T0IF	INTF	PEIE	T0CKIE	T0IE	INTE	0000 0000	0000 0000
06h, Unbanked	CPUSTA	—	—	STKAV	GLINTD	$\overline{\text{TO}}$	$\overline{\text{PD}}$	POR	BOR	--11 11qq	--11 qquu
14h, Bank 2	PR1	Timer1 Period Register								xxxx xxxx	uuuu uuuu
15h, Bank 2	PR2	Timer2 Period Register								xxxx xxxx	uuuu uuuu
10h, Bank 3	PW1DCL	DC1	DC0	—	—	—	—	—	—	xx-- ----	uu-- ----
11h, Bank 3	PW2DCL	DC1	DC0	TM2PW2	—	—	—	—	—	xx0- ----	uu0- ----
10h, Bank 7	PW3DCL	DC1	DC0	TM2PW3	—	—	—	—	—	xx0- ----	uu0- ----
12h, Bank 3	PW1DCH	DC9	DC8	DC7	DC6	DC5	DC4	DC3	DC2	xxxx xxxx	uuuu uuuu
13h, Bank 3	PW2DCH	DC9	DC8	DC7	DC6	DC5	DC4	DC3	DC2	xxxx xxxx	uuuu uuuu
11h, Bank 7	PW3DCH	DC9	DC8	DC7	DC6	DC5	DC4	DC3	DC2	xxxx xxxx	uuuu uuuu

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0', q = value depends on conditions.
Shaded cells are not used by PWM Module.

13.2.3 READING THE CAPTURE REGISTERS

The Capture overflow status flag bits are double buffered. The master bit is set if one captured word is already residing in the Capture register 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.

An example of an instruction sequence to read capture registers and capture overflow flag bits is shown in Example 13-1. Depending on the capture source, different registers will need to be read.

EXAMPLE 13-1: SEQUENCE TO READ CAPTURE REGISTERS

```
    MOVLB 3                 ; Select Bank 3
    MOVPF CA2L, LO_BYTE     ; Read Capture2 low byte, store in LO_BYTE
    MOVPF CA2H, HI_BYTE     ; Read Capture2 high byte, store in HI_BYTE
    MOVPF TCON2, STAT_VAL   ; Read TCON2 into file STAT_VAL
```

TABLE 13-6: REGISTERS ASSOCIATED WITH CAPTURE

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	$\overline{\text{MCLR}}$, WDT
16h, Bank 3	TCON1	CA2ED1	CA2ED0	CA1ED1	CA1ED0	T16	TMR3CS	TMR2CS	TMR1CS	0000 0000	0000 0000
17h, Bank 3	TCON2	CA2OVF	CA1OVF	PWM2ON	PWM1ON	CA1/$\overline{\text{PR3}}$	TMR3ON	TMR2ON	TMR1ON	0000 0000	0000 0000
16h, Bank 7	TCON3	—	CA4OVF	CA3OVF	CA4ED1	CA4ED0	CA3ED1	CA3ED0	PWM3ON	-000 0000	-000 0000
12h, Bank 2	TMR3L	Holding Register for the Low Byte of the 16-bit TMR3 Register								xxxx xxxx	uuuu uuuu
13h, Bank 2	TMR3H	Holding Register for the High Byte of the 16-bit TMR3 Register								xxxx xxxx	uuuu uuuu
16h, Bank 1	PIR1	RBIF	TMR3IF	TMR2IF	TMR1IF	CA2IF	CA1IF	TX1IF	RC1IF	x000 0010	u000 0010
17h, Bank 1	PIE1	RBIE	TMR3IE	TMR2IE	TMR1IE	CA2IE	CA1IE	TX1IE	RC1IE	0000 0000	0000 0000
10h, Bank 4	PIR2	SSPIF	BCLIF	ADIF	—	CA4IF	CA3IF	TX2IF	RC2IF	000- 0010	000- 0010
11h, Bank 4	PIE2	SSPIE	BCLIE	ADIE	—	CA4IE	CA3IE	TX2IE	RC2IE	000- 0000	000- 0000
07h, Unbanked	INTSTA	PEIF	T0CKIF	T0IF	INTF	PEIE	T0CKIE	T0IE	INTE	0000 0000	0000 0000
06h, Unbanked	CPUSTA	—	—	STKAV	GLINTD	$\overline{\text{TO}}$	$\overline{\text{PD}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$	--11 11qq	--11 qquu
16h, Bank 2	PR3L/CA1L	Timer3 Period Register, Low Byte/Capture1 Register, Low Byte								xxxx xxxx	uuuu uuuu
17h, Bank 2	PR3H/CA1H	Timer3 Period Register, High Byte/Capture1 Register, High Byte								xxxx xxxx	uuuu uuuu
14h, Bank 3	CA2L	Capture2 Low Byte								xxxx xxxx	uuuu uuuu
15h, Bank 3	CA2H	Capture2 High Byte								xxxx xxxx	uuuu uuuu
12h, Bank 7	CA3L	Capture3 Low Byte								xxxx xxxx	uuuu uuuu
13h, Bank 7	CA3H	Capture3 High Byte								xxxx xxxx	uuuu uuuu
14h, Bank 7	CA4L	Capture4 Low Byte								xxxx xxxx	uuuu uuuu
15h, Bank 7	CA4H	Capture4 High Byte								xxxx xxxx	uuuu uuuu

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0', q = value depends on condition.
Shaded cells are not used by Capture.

NOTES:

TABLE 14-10: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE TRANSMISSION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	$\overline{\text{MCLR}}$, WDT
16h, Bank 1	PIR1	RBIF	TMR3IF	TMR2IF	TMR1IF	CA2IF	CA1IF	TX1IF	RC1IF	x000 0010	u000 0010
17h, Bank 1	PIE1	RBIE	TMR3IE	TMR2IE	TMR1IE	CA2IE	CA1IE	TX1IE	RC1IE	0000 0000	0000 0000
13h, Bank 0	RCSTA1	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00u
15h, Bank 0	TXSTA1	CSRC	TX9	TXEN	SYNC	—	—	TRMT	TX9D	0000 --1x	0000 --1u
16h, Bank 0	TXREG1	TX7	TX6	TX5	TX4	TX3	TX2	TX1	TX0	xxxx xxxx	uuuu uuuu
17h, Bank 0	SPBRG1	Baud Rate Generator Register								0000 0000	0000 0000
10h, Bank 4	PIR2	SSPIF	BCLIF	ADIF	—	CA4IF	CA3IF	TX2IF	RC2IF	000- 0010	000- 0010
11h, Bank 4	PIE2	SSPIE	BCLIE	ADIE	—	CA4IE	CA3IE	TX2IE	RC2IE	000- 0000	000- 0000
13h, Bank 4	RCSTA2	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00u
16h, Bank 4	TXREG2	TX7	TX6	TX5	TX4	TX3	TX2	TX1	TX0	xxxx xxxx	uuuu uuuu
15h, Bank 4	TXSTA2	CSRC	TX9	TXEN	SYNC	—	—	TRMT	TX9D	0000 --1x	0000 --1u
17h, Bank 4	SPBRG2	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, u = unchanged, - = unimplemented, read as a '0'. Shaded cells are not used for synchronous slave transmission.

TABLE 14-11: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE RECEPTION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	$\overline{\text{MCLR}}$, WDT
16h, Bank1	PIR1	RBIF	TMR3IF	TMR2IF	TMR1IF	CA2IF	CA1IF	TX1IF	RC1IF	x000 0010	u000 0010
17h, Bank1	PIE1	RBIE	TMR3IE	TMR2IE	TMR1IE	CA2IE	CA1IE	TX1IE	RC1IE	0000 0000	0000 0000
13h, Bank0	RCSTA1	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00u
14h, Bank0	RCREG1	RX7	RX6	RX5	RX4	RX3	RX2	RX1	RX0	xxxx xxxx	uuuu uuuu
15h, Bank 0	TXSTA1	CSRC	TX9	TXEN	SYNC	—	—	TRMT	TX9D	0000 --1x	0000 --1u
17h, Bank 0	SPBRG1	Baud Rate Generator Register								0000 0000	0000 0000
10h, Bank 4	PIR2	SSPIF	BCLIF	ADIF	—	CA4IF	CA3IF	TX2IF	RC2IF	000- 0010	000- 0010
11h, Bank 4	PIE2	SSPIE	BCLIE	ADIE	—	CA4IE	CA3IE	TX2IE	RC2IE	000- 0000	000- 0000
13h, Bank 4	RCSTA2	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00u
14h, Bank 4	RCREG2	RX7	RX6	RX5	RX4	RX3	RX2	RX1	RX0	xxxx xxxx	uuuu uuuu
15h, Bank 4	TXSTA2	CSRC	TX9	TXEN	SYNC	—	—	TRMT	TX9D	0000 --1x	0000 --1u
17h, Bank 4	SPBRG2	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, u = unchanged, - = unimplemented, read as a '0'. Shaded cells are not used for synchronous slave reception.

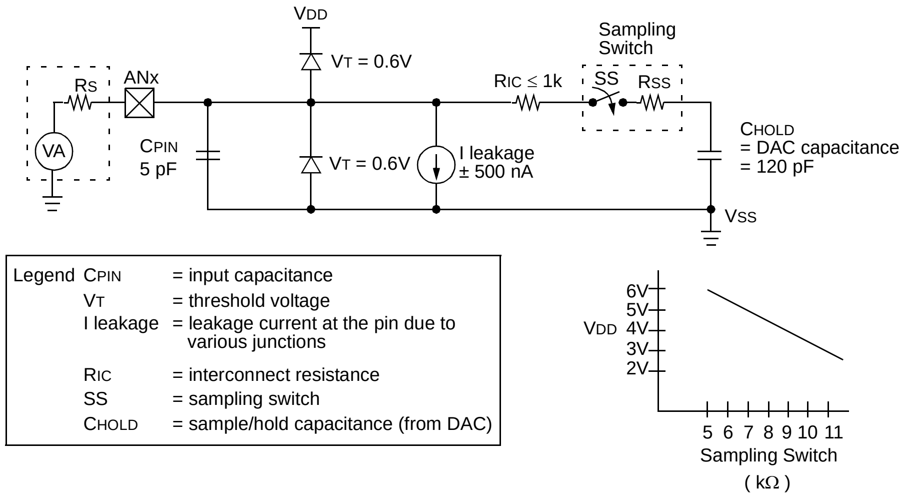
15.4 Example Program

Example 15-2 shows MPLAB® C17 ’C’ code for using the I^{2}C module in Master mode to communicate with a 24LC01B serial EEPROM. This example uses the PIC® MCU ‘C’ libraries included with MPLAB C17.

EXAMPLE 15-2: INTERFACING TO A 24LC01B SERIAL EEPROM (USING MPLAB C17)

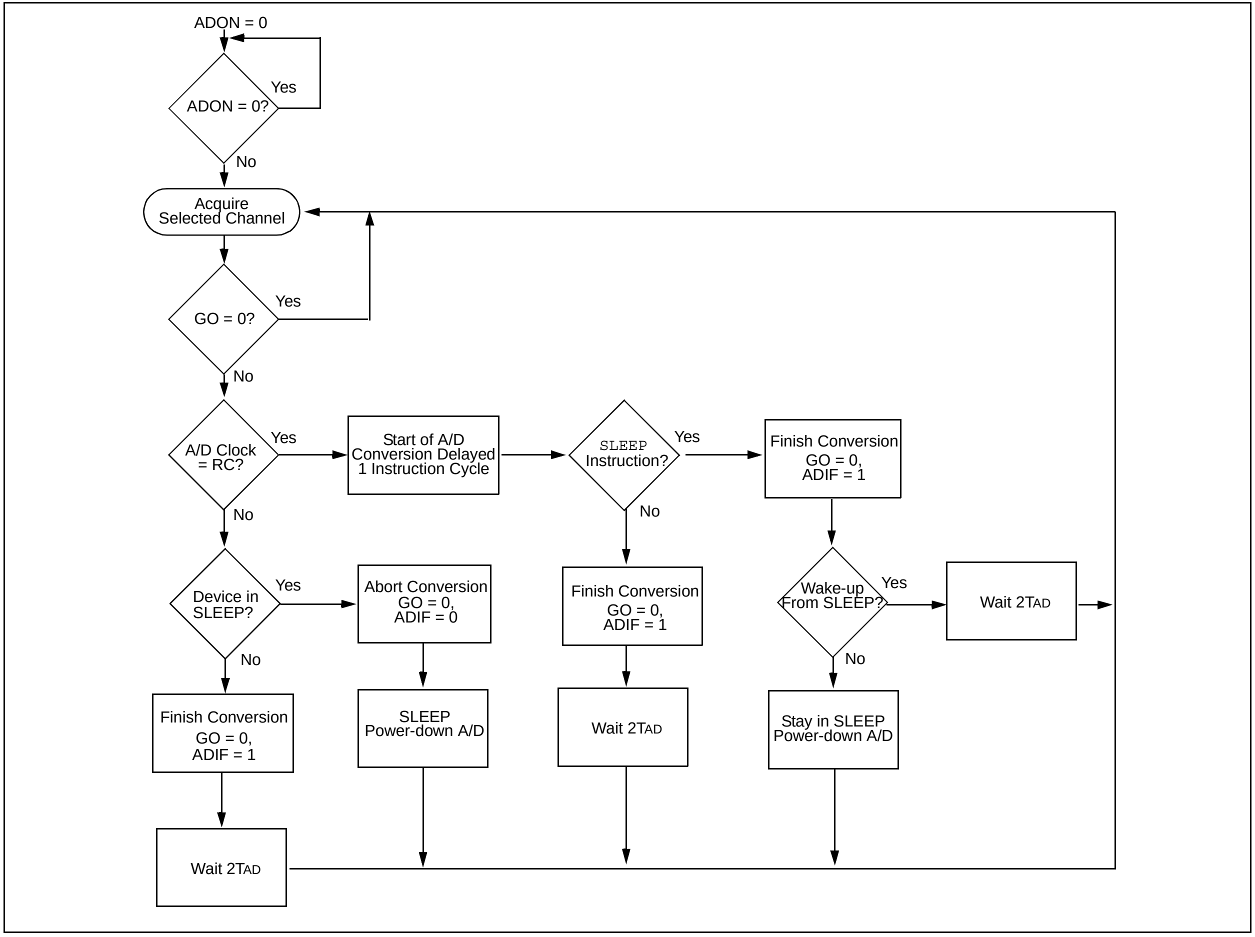
```
// Include necessary header files
#include <p17c756.h>     // Processor header file
#include <delays.h>      // Delay routines header file
#include <stdlib.h>      // Standard Library header file
#include <i2c16.h>       // I2C routines header file

#define CONTROL 0xa0     // Control byte definition for 24LC01B

// Function declarations
void main(void);
void WritePORTD(static unsigned char data);
void ByteWrite(static unsigned char address,static unsigned char data);
unsigned char ByteRead(static unsigned char address);
void ACKPoll(void);

// Main program
void main(void)
{
static unsigned char address;    // I2C address of 24LC01B
static unsigned char datao;      // Data written to 24LC01B
static unsigned char datai;      // Data read from 24LC01B

    address = 0;                 // Preset address to 0
    OpenI2C(MASTER,SLEW_ON);     // Configure I2C Module Master mode, Slew rate control on
    SSPADD = 39;                 // Configure clock for 100KHz

    while(address<128)           // Loop 128 times, 24LC01B is 128x8
    {
        datao = PORTB;
        do
        {
            ByteWrite(address,datao);   // Write data to EEPROM
            ACKPoll();                  // Poll the 24LC01B for state
            datai = ByteRead(address);  // Read data from EEPROM into SSPBUF
        } while(datai != datao);        // Loop as long as data not correctly
                                        //    written to 24LC01B

        address++;                      // Increment address
    }
    while(1)                            // Done writing 128 bytes to 24LC01B, Loop forever
    {
        Nop();
    }
}
```

FIGURE 16-3: ANALOG INPUT MODEL



Legend		
	CPIN	= input capacitance
	VT	= threshold voltage
	I leakage	= leakage current at the pin due to various junctions
	RIC	= interconnect resistance
	SS	= sampling switch
	CHOLD	= sample/hold capacitance (from DAC)

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

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'

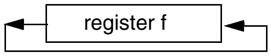
Example: `BCF     FLAG_REG,  7`

Before Instruction
FLAG_REG = 0xC7

After Instruction
FLAG_REG = 0x47

RLNCF Rotate Left f (no carry)

Syntax:	[label] RLNCF f,d
Operands:	0 ≤ f ≤ 255 d ∈ [0,1]
Operation:	f<n> → d<n+1>; f<7> → d<0>
Status Affected:	None
Encoding:	`0010` `001d` `ffff` `ffff`
Description:	The contents of register 'f' are rotated one bit to the left. If 'd' is 0, the result is placed 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: `RLNCF REG, 1`

Before Instruction

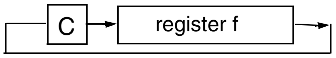
C	=	`0`
REG	=	`1110 1011`

After Instruction

C	=	
REG	=	`1101 0111`

RRCF Rotate Right f through Carry

Syntax:	[label] RRCF f,d
Operands:	0 ≤ f ≤ 255 d ∈ [0,1]
Operation:	f<n> → d<n-1>; f<0> → C; C → d<7>
Status Affected:	C
Encoding:	`0001` `100d` `ffff` `ffff`
Description:	The contents of register 'f' are rotated one bit to the right through the Carry Flag. 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: `RRCF REG1,0`

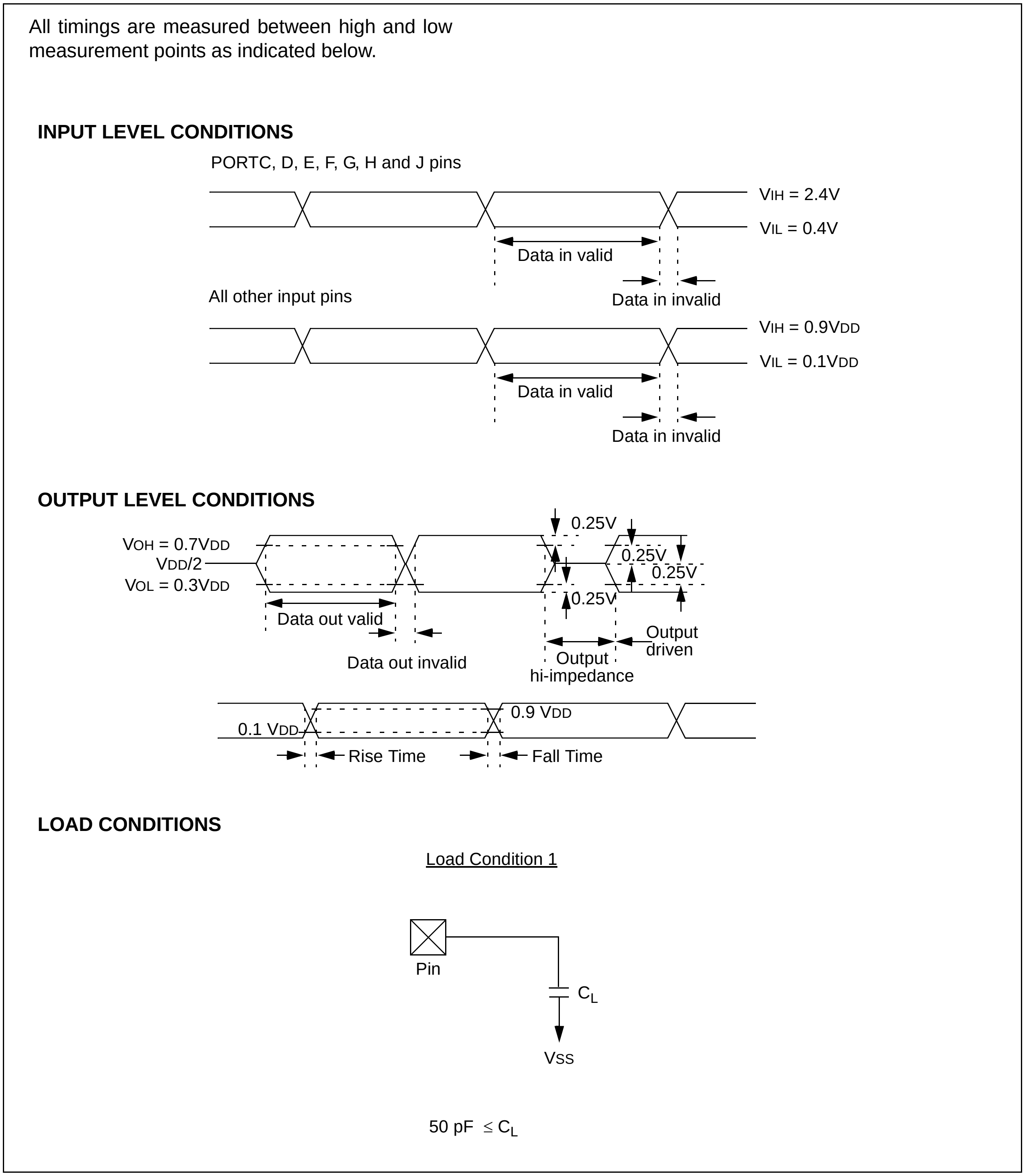
Before Instruction

REG1	=	`1110 0110`
C	=	`0`

After Instruction

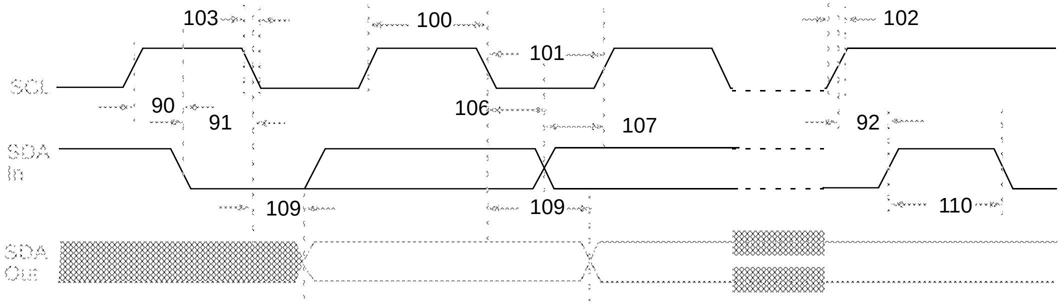
REG1	=	`1110 0110`
WREG	=	`0111 0011`
C	=	`0`

FIGURE 20-5: PARAMETER MEASUREMENT INFORMATION



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

FIGURE 20-18: I^{2}C BUS DATA TIMING



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

TABLE 20-13: I^{2}C 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 I^{2}C pins.

2: A fast mode (400 KHz) I^{2}C bus device can be used in a standard mode I^{2}C 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.

FIGURE 21-17: TYPICAL, MINIMUM AND MAXIMUM VOH vs. IOH (VDD = 5V, -40°C TO +125°C)

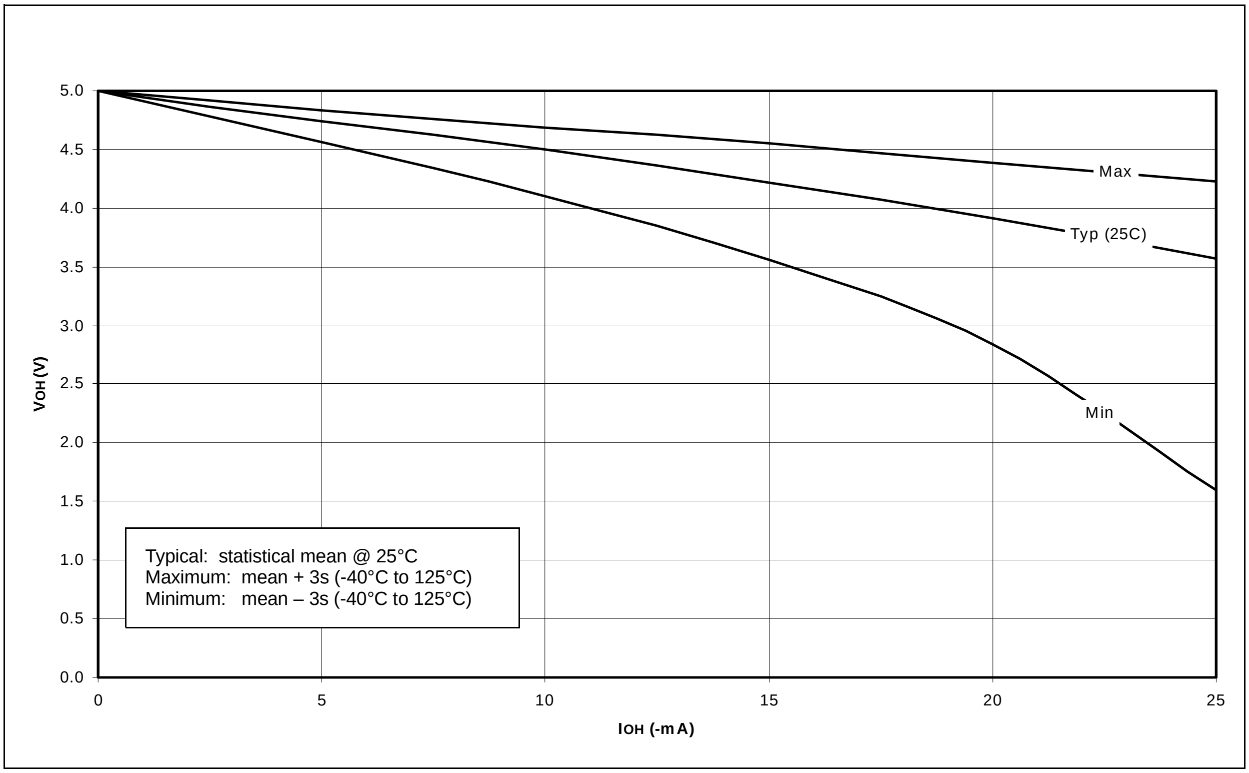
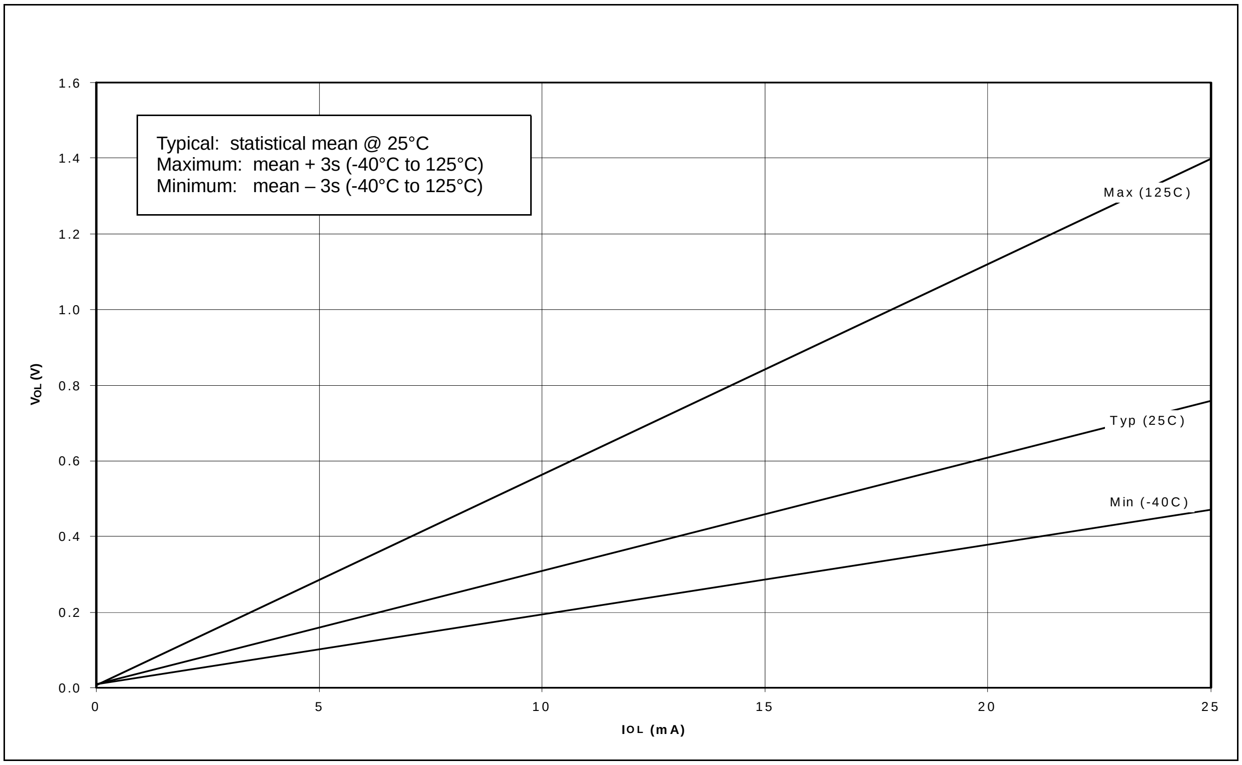


FIGURE 21-18: TYPICAL, MINIMUM AND MAXIMUM VOL vs. IOL (VDD = 5V, -40°C TO +125°C)



Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE**.** Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
= ISO/TS 16949 =

Trademarks

The Microchip name and logo, the Microchip logo, dsPIC, FlashFlex, KEELOQ, KEELOQ logo, MPLAB, PIC, PICmicro, PICSTART, PIC^{32} logo, rfPIC, SST, SST Logo, SuperFlash and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

FilterLab, Hampshire, HI-TECH C, Linear Active Thermistor, MTP, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

Analog-for-the-Digital Age, Application Maestro, BodyCom, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, HI-TIDE, In-Circuit Serial Programming, ICSP, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, mTouch, Omniscient Code Generation, PICC, PICC-18, PICDEM, PICDEM.net, PICkit, PICtail, REAL ICE, rfLAB, Select Mode, SQI, Serial Quad I/O, Total Endurance, TSHARC, UniWinDriver, WiperLock, ZENA and Z-Scale are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

GestIC and ULPP are registered trademarks of Microchip Technology Germany II GmbH & Co. & KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

Printed on recycled paper.

ISBN: 9781620769317

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.