Microchip Technology - PIC17C766-16I/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	66
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 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/pic17c766-16i-pt

2.0 DEVICE VARIETIES

Each device has a variety of frequency ranges and packaging options. Depending on application and production requirements, the proper device option can be selected using the information in the PIC17C7XX Product Selection System section at the end of this data sheet. When placing orders, please use the "PIC17C7XX Product Identification System" at the back of this data sheet to specify the correct part number. When discussing the functionality of the device, memory technology and voltage range does not matter.

There are two memory type options. These are specified in the middle characters of the part number.

1. **C**, as in PIC17**C**756A. These devices have EPROM type memory.
2. **CR**, as in PIC17**CR**756A. These devices have ROM type memory.

All these devices operate over the standard voltage range. Devices are also offered which operate over an extended voltage range (and reduced frequency range). Table 2-1 shows all possible memory types and voltage range designators for a particular device. These designators are in **bold** typeface.

TABLE 2-1: DEVICE MEMORY VARIETIES

Memory Type	Voltage Range	
	Standard	**Extended**
EPROM	PIC17**C**XXX	PIC17**LC**XXX
ROM	PIC17**CR**XXX	PIC17**LCR**XXX

Note: Not all memory technologies are available for a particular device.

2.1 UV Erasable Devices

The UV erasable version, offered in CERQUAD package, is optimal for prototype development and pilot programs.

The UV erasable version can be erased and reprogrammed to any of the configuration modes. Third party programmers also are available; refer to the *Third Party Guide* for a list of sources.

2.2 One-Time-Programmable (OTP) Devices

The availability of OTP devices is especially useful for customers expecting frequent code changes and updates.

The OTP devices, packaged in plastic packages, permit the user to program them once. In addition to the program memory, the configuration bits must be programmed.

2.3 Quick-Turnaround-Production (QTP) Devices

Microchip offers a QTP Programming Service for factory production orders. This service is made available for users who choose not to program a medium to high quantity of units and whose code patterns have stabilized. The devices are identical to the OTP devices but with all EPROM locations and configuration options already programmed by the factory. Certain code and prototype verification procedures apply before production shipments are available. Please contact your local Microchip Technology sales office for more details.

2.4 Serialized Quick-Turnaround Production (SQTP^{sm}) Devices

Microchip offers a unique programming service, where a few user defined locations in each device are programmed with different serial numbers. The serial numbers may be random, pseudo-random or sequential.

Serial programming allows each device to have a unique number which can serve as an entry code, password or ID number.

2.5 Read Only Memory (ROM) Devices

Microchip offers masked ROM versions of several of the highest volume parts, thus giving customers a low cost option for high volume, mature products.

ROM devices do not allow serialization information in the program memory space.

For information on submitting ROM code, please contact your regional sales office.

Note: Presently, NO ROM versions of the PIC17C7XX devices are available.

5.1.4 TIME-OUT SEQUENCE

On power-up, the time-out sequence is as follows: First, the internal POR signal goes high when the POR trip point is reached. If MCLR is high, then both the OST and PWRT timers start. In general, the PWRT time-out is longer, except with low frequency crystals/resonators. The total time-out also varies based on oscillator configuration. Table 5-1 shows the times that are associated with the oscillator configuration. Figure 5-5 and Figure 5-6 display these time-out sequences.

If the device voltage is not within electrical specification at the end of a time-out, the MCLR/VPP pin must be held low until the voltage is within the device specification. The use of an external RC delay is sufficient for many of these applications.

The time-out sequence begins from the first rising edge of MCLR.

Table 5-3 shows the RESET conditions for some special registers, while Table 5-4 shows the initialization conditions for all the registers.

TABLE 5-1: TIME-OUT IN VARIOUS SITUATIONS

Oscillator Configuration	POR, BOR	Wake-up from SLEEP	MCLR Reset
XT, LF	Greater of: 96 ms or 1024TOSC	1024TOSC	—
EC, RC	Greater of: 96 ms or 1024TOSC	—	—

TABLE 5-2: STATUS BITS AND THEIR SIGNIFICANCE

POR	BOR(1)	TO	PD	Event
0	0	1	1	Power-on Reset
1	1	1	0	MCLR Reset during SLEEP or interrupt wake-up from SLEEP
1	1	0	1	WDT Reset during normal operation
1	1	0	0	WDT Wake-up during SLEEP
1	1	1	1	MCLR Reset during normal operation
1	0	1	1	Brown-out Reset
0	0	0	x	Illegal, TO is set on POR
0	0	x	0	Illegal, PD is set on POR
x	x	1	1	CLRWDT instruction executed

Note 1: When BODEN is enabled, else the BOR status bit is unknown.

TABLE 5-3: RESET CONDITION FOR THE PROGRAM COUNTER AND THE CPUSTA REGISTER

Event		PCH:PCL	CPUSTA(4)	OST Active
Power-on Reset		0000h	--11 1100	Yes
Brown-out Reset		0000h	--11 1110	Yes
MCLR Reset during normal operation		0000h	--11 1111	No
MCLR Reset during SLEEP		0000h	--11 1011	Yes(2)
WDT Reset during normal operation		0000h	--11 0111	No
WDT Reset during SLEEP(3)		0000h	--11 0011	Yes(2)
Interrupt Wake-up from SLEEP	GLINTD is set	PC + 1	--11 1011	Yes(2)
	GLINTD is clear	PC + 1(1)	--10 1011	Yes(2)

Legend: u = unchanged, x = unknown, - = unimplemented, read as '0'

Note 1: On wake-up, this instruction is executed. The instruction at the appropriate interrupt vector is fetched and then executed.

2: The OST is only active (on wake-up) when the oscillator is configured for XT or LF modes.

3: The Program Counter = 0; that is, the device branches to the RESET vector and places SFRs in WDT Reset states. This is different from the mid-range devices.

4: When BODEN is enabled, else the BOR status bit is unknown.

6.3 Peripheral Interrupt Request Register1 (PIR1) and Register2 (PIR2)

These registers contains the individual flag bits for the peripheral interrupts.

Note: These bits will be set by the specified condition, even if the corresponding interrupt enable bit is cleared (interrupt disabled), or the GLINTD bit is set (all interrupts disabled). Before enabling an interrupt, the user may wish to clear the interrupt flag to ensure that the program does not immediately branch to the peripheral Interrupt Service Routine.

REGISTER 6-4: PIR1 REGISTER (ADDRESS: 16h, BANK 1)

R/W-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-1	R-0
RBIF	TMR3IF	TMR2IF	TMR1IF	CA2IF	CA1IF	TX1IF	RC1IF
bit 7							bit 0

bit 7 **RBIF**: PORTB Interrupt-on-Change Flag bit
1 = One of the PORTB inputs changed (software must end the mismatch condition)
0 = None of the PORTB inputs have changed

bit 6 **TMR3IF**: TMR3 Interrupt Flag bit

If Capture1 is enabled (CA1/$\overline{PR3}$ = 1):
1 = TMR3 overflowed
0 = TMR3 did not overflow

If Capture1 is disabled (CA1/$\overline{PR3}$ = 0):
1 = TMR3 value has rolled over to 0000h from equalling the period register (PR3H:PR3L) value
0 = TMR3 value has not rolled over to 0000h from equalling the period register (PR3H:PR3L) value

bit 5 **TMR2IF**: TMR2 Interrupt Flag bit
1 = TMR2 value has rolled over to 0000h from equalling the period register (PR2) value
0 = TMR2 value has not rolled over to 0000h from equalling the period register (PR2) value

bit 4 **TMR1IF**: TMR1 Interrupt Flag bit

If TMR1 is in 8-bit mode (T16 = 0):
1 = TMR1 value has rolled over to 0000h from equalling the period register (PR1) value
0 = TMR1 value has not rolled over to 0000h from equalling the period register (PR1) value

If Timer1 is in 16-bit mode (T16 = 1):
1 = TMR2:TMR1 value has rolled over to 0000h from equalling the period register (PR2:PR1) value
0 = TMR2:TMR1 value has not rolled over to 0000h from equalling the period register (PR2:PR1) value

bit 3 **CA2IF**: Capture2 Interrupt Flag bit
1 = Capture event occurred on RB1/CAP2 pin
0 = Capture event did not occur on RB1/CAP2 pin

bit 2 **CA1IF**: Capture1 Interrupt Flag bit
1 = Capture event occurred on RB0/CAP1 pin
0 = Capture event did not occur on RB0/CAP1 pin

bit 1 **TX1IF**: USART1 Transmit Interrupt Flag bit (state controlled by hardware)
1 = USART1 Transmit buffer is empty
0 = USART1 Transmit buffer is full

bit 0 **RC1IF**: USART1 Receive Interrupt Flag bit (state controlled by hardware)
1 = USART1 Receive buffer is full
0 = USART1 Receive buffer is empty

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

7.2.2.2 CPU Status Register (CPUSTA)

The CPUSTA register contains the status and control bits for the CPU. This register has a bit that is used to globally enable/disable interrupts. If only a specific interrupt is desired to be enabled/disabled, please refer to the Interrupt Status (INTSTA) register and the Peripheral Interrupt Enable (PIE) registers. The CPUSTA register also indicates if the stack is available and contains the Power-down ($\overline{PD}$) and Time-out ($\overline{TO}$) bits. The $\overline{TO}$, $\overline{PD}$, and STKAV bits are not writable. These bits are set and cleared according to device logic. Therefore, the result of an instruction with the CPUSTA register as destination may be different than intended.

The $\overline{POR}$ bit allows the differentiation between a Power-on Reset, external $\overline{MCLR}$ Reset, or a WDT Reset. The $\overline{BOR}$ bit indicates if a Brown-out Reset occurred.

Note 1:	The $\overline{BOR}$ status bit is a don't care and is not necessarily predictable if the Brown-out circuit is disabled (when the BODEN bit in the Configuration word is programmed).

REGISTER 7-2: CPUSTA REGISTER (ADDRESS: 06h, UNBANKED)

U-0	U-0	R-1	R/W-1	R-1	R-1	R/W-0	R/W-1
—	—	STKAV	GLINTD	$\overline{TO}$	$\overline{PD}$	$\overline{POR}$	$\overline{BOR}$
bit 7							bit 0

bit 7-6 **Unimplemented**: Read as '0'

bit 5 **STKAV**: Stack Available bit
This bit indicates that the 4-bit stack pointer value is Fh, or has rolled over from Fh → 0h (stack overflow).
1 = Stack is available
0 = Stack is full, or a stack overflow may have occurred (once this bit has been cleared by a stack overflow, only a device RESET will set this bit)

bit 4 **GLINTD**: Global Interrupt Disable bit
This bit disables all interrupts. When enabling interrupts, only the sources with their enable bits set can cause an interrupt.
1 = Disable all interrupts
0 = Enables all unmasked interrupts

bit 3 **$\overline{TO}$**: WDT Time-out Status bit
1 = After power-up, by a CLRWDT instruction, or by a SLEEP instruction
0 = A Watchdog Timer time-out occurred

bit 2 **$\overline{PD}$**: Power-down Status bit
1 = After power-up or by the CLRWDT instruction
0 = By execution of the SLEEP instruction

bit 1 **$\overline{POR}$**: Power-on Reset Status bit
1 = No Power-on Reset occurred
0 = A Power-on Reset occurred (must be set by software)

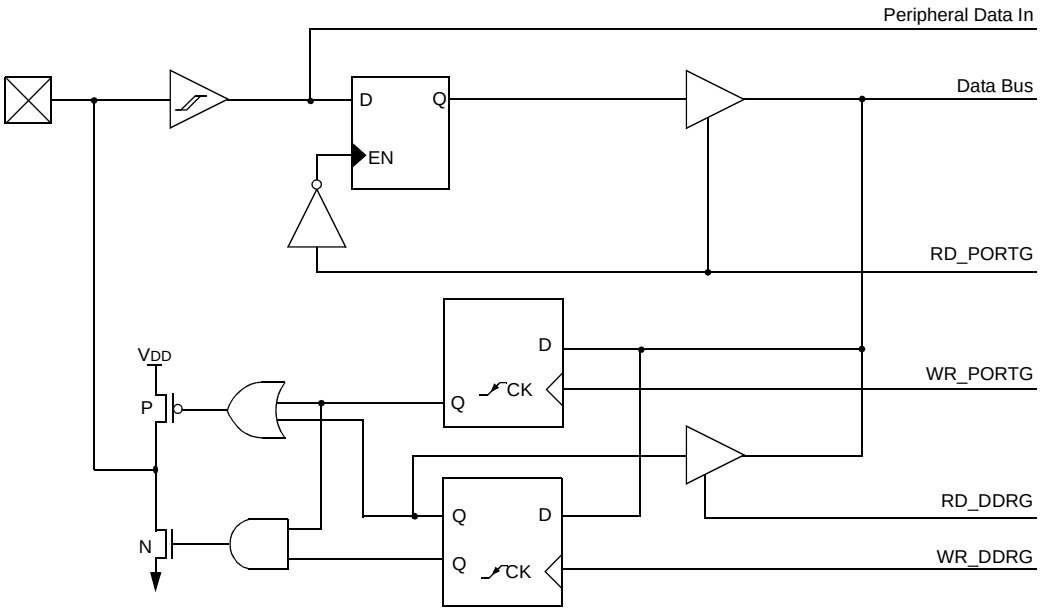
bit 0 **$\overline{BOR}$**: Brown-out Reset Status bit

When BODEN Configuration bit is set (enabled):
1 = No Brown-out Reset occurred
0 = A Brown-out Reset occurred (must be set by software)

When BODEN Configuration bit is clear (disabled):
Don't care

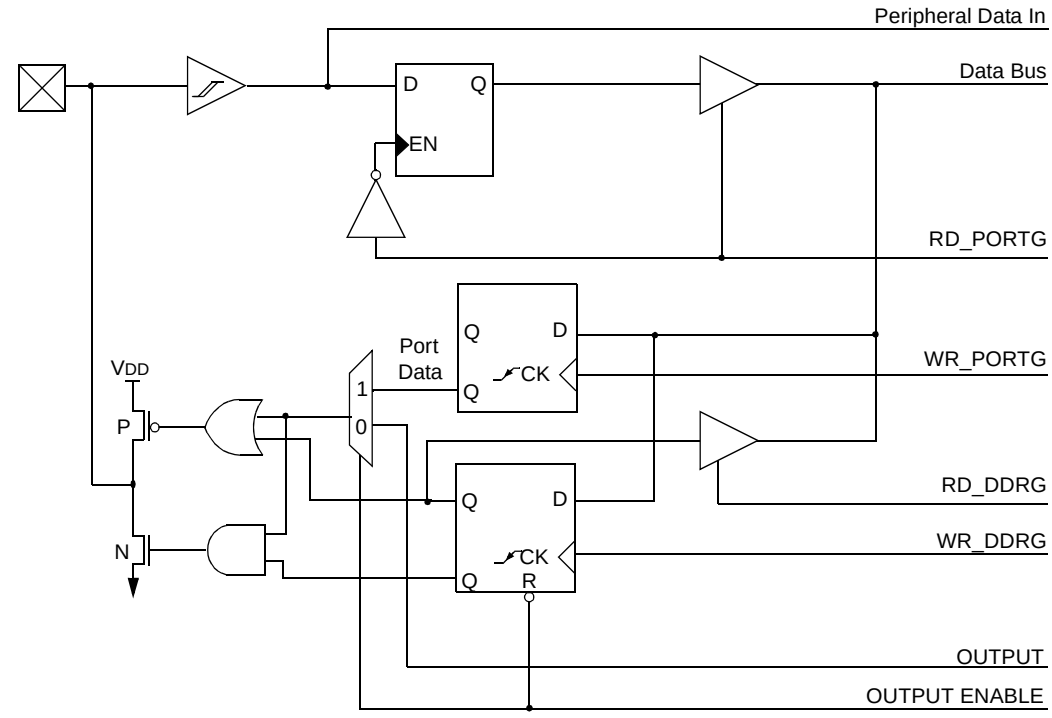
Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
- n = Value at POR Reset	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

FIGURE 10-15: RG4 BLOCK DIAGRAM



Note: I/O pins have protection diodes to VDD and VSS.

FIGURE 10-16: RG7:RG5 BLOCK DIAGRAM



Note: I/O pins have protection diodes to VDD and VSS.

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:

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

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

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

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 (F_{PWM}), 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 (F_{PWM}).

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

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

where: F_{PWM} = 1 / period of PWM

The PWMx duty cycle is as follows:

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

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.

14.0 UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (USART) MODULES

Each USART module is a serial I/O module. There are two USART modules that are available on the PIC17C7XX. They are specified as USART1 and USART2. The description of the operation of these modules is generic in regard to the register names and pin names used. Table 14-1 shows the generic names that are used in the description of operation and the actual names for both USART1 and USART2. Since the control bits in each register have the same function, their names are the same (there is no need to differentiate).

The Transmit Status and Control Register (TXSTA) is shown in Figure 14-1, while the Receive Status and Control Register (RCSTA) is shown in Figure 14-2.

TABLE 14-1: USART MODULE GENERIC NAMES

Generic Name	USART1 Name	USART2 Name
Registers		
RCSTA	RCSTA1	RCSTA2
TXSTA	TXSTA1	TXSTA2
SPBRG	SPBRG1	SPBRG2
RCREG	RCREG1	RCREG2
TXREG	TXREG1	TXREG2
Interrupt Control Bits		
RCIE	RC1IE	RC2IE
RCIF	RC1IF	RC2IF
TXIE	TX1IE	TX2IE
TXIF	TX1IF	TX2IF
Pins		
RX/DT	RA4/RX1/DT1	RG6/RX2/DT2
TX/CK	RA5/TX1/CK1	RG7/TX2/CK2

REGISTER 14-1: TXSTA1 REGISTER (ADDRESS: 15h, BANK 0)
TXSTA2 REGISTER (ADDRESS: 15h, BANK 4)

R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R-1	R/W-x
CSRC	TX9	TXEN	SYNC	—	—	TRMT	TX9D
bit 7							bit 0

bit 7 **CSRC**: Clock Source Select bit

Synchronous mode:
1 = Master mode (clock generated internally from BRG)
0 = Slave mode (clock from external source)

Asynchronous mode:
Don't care

bit 6 **TX9**: 9-bit Transmit Select bit
1 = Selects 9-bit transmission
0 = Selects 8-bit transmission

bit 5 **TXEN**: Transmit Enable bit
1 = Transmit enabled
0 = Transmit disabled
SREN/CREN overrides TXEN in SYNC mode

bit 4 **SYNC**: USART Mode Select bit
(Synchronous/Asynchronous)
1 = Synchronous mode
0 = Asynchronous mode

bit 3-2 **Unimplemented**: Read as '0'

bit 1 **TRMT**: Transmit Shift Register (TSR) Empty bit
1 = TSR empty
0 = TSR full

bit 0 **TX9D**: 9th bit of Transmit Data (can be used to calculate the parity in software)

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

TABLE 14-9: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER 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, 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
14h, Bank 0	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 master reception.

FIGURE 15-8: SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 0)

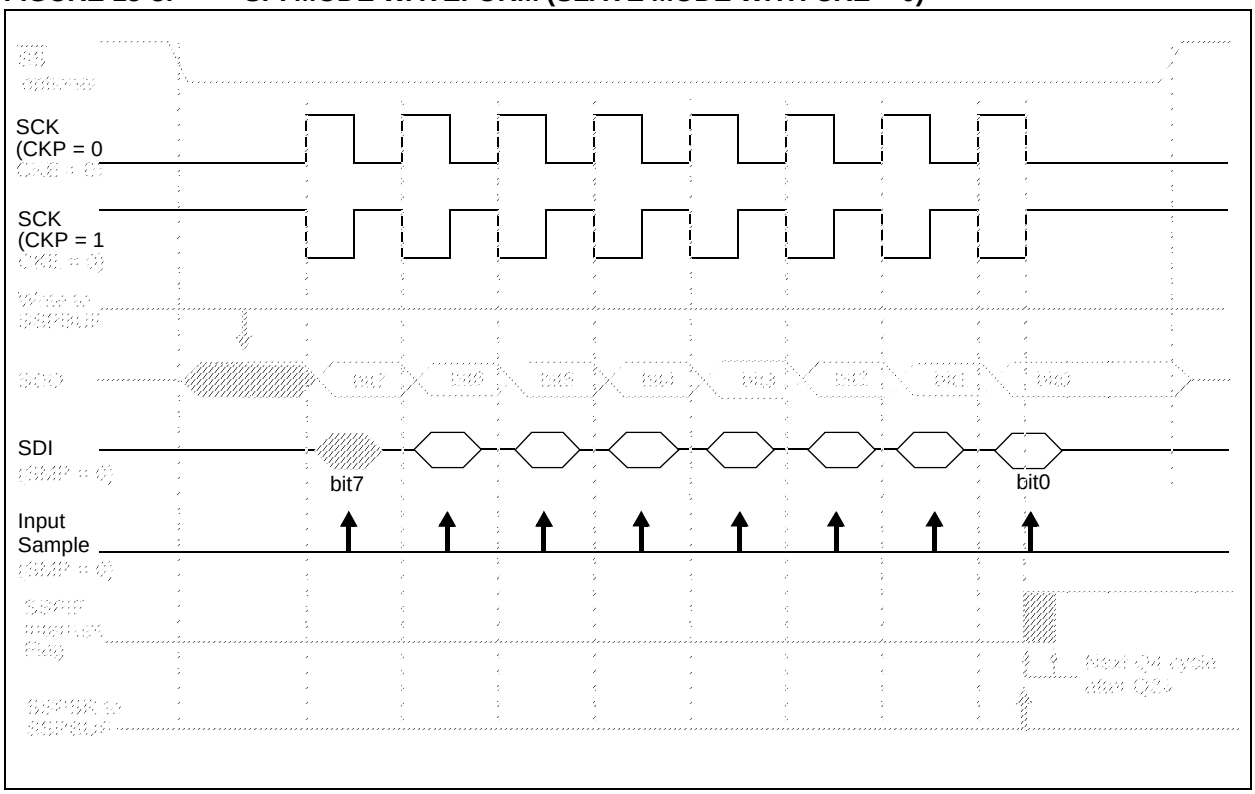
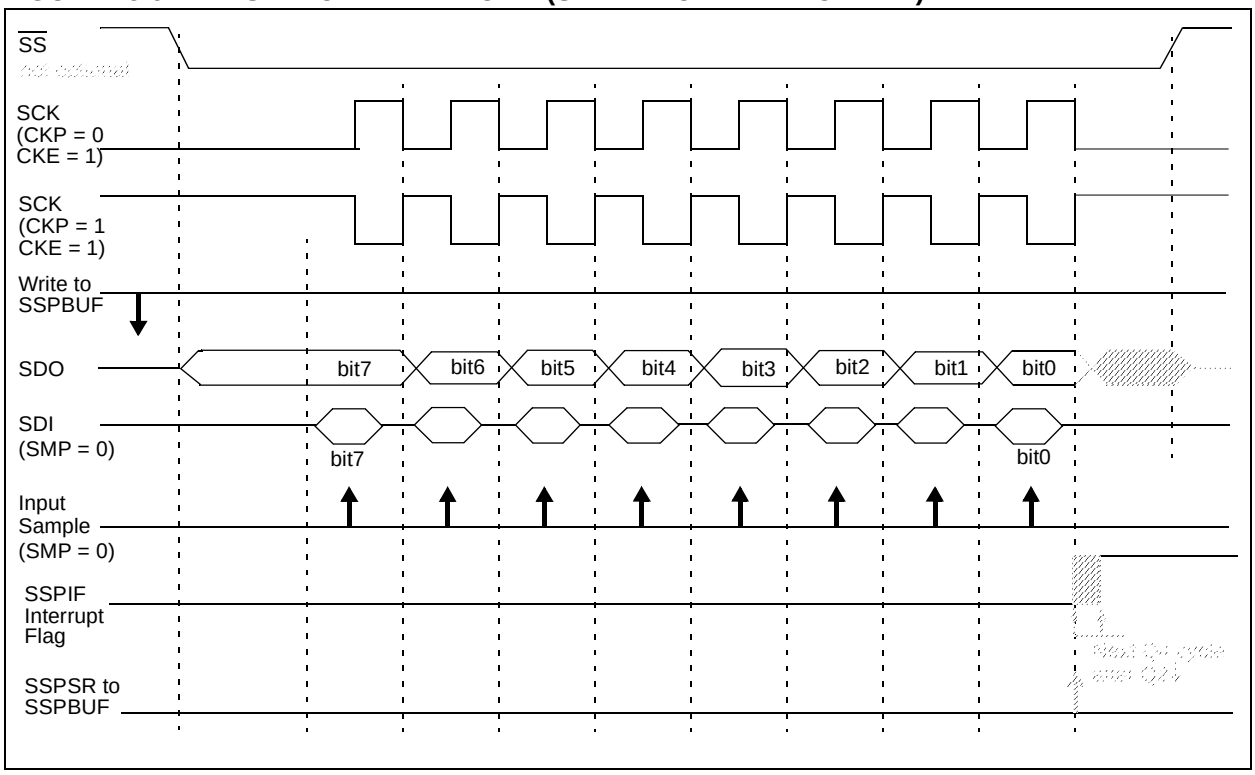


FIGURE 15-9: SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 1)



15.2.1.1 Addressing

Once the MSSP module has been enabled, it waits for a START condition to occur. Following the START condition, the 8-bits are shifted into the SSPSR register. All incoming bits are sampled with the rising edge of the clock (SCL) line. The value of register SSPSR<7:1> is compared to the value of the SSPADD register. The address is compared on the falling edge of the eighth clock (SCL) pulse. If the addresses match and the BF and SSPOV bits are clear, the following events occur:

a) The SSPSR register value is loaded into the SSPBUF register on the falling edge of the 8th SCL pulse.
b) The buffer full bit, BF, is set on the falling edge of the 8th SCL pulse.
c) An $\overline{ACK}$ pulse is generated.
d) SSP interrupt flag bit, SSPIF (PIR2<7>), is set (interrupt is generated if enabled) - on the falling edge of the 9th SCL pulse.

In 10-bit address mode, two address bytes need to be received by the slave. The five Most Significant bits (MSbs) of the first address byte specify if this is a 10-bit address. Bit R/$\overline{W}$ (SSPSTAT<2>) must specify a write so the slave device will receive the second address byte. For a 10-bit address, the first byte would equal '1111 0 A9 A8 0', where A9 and A8 are the two MSbs of the address. The sequence of events for a 10-bit address is as follows, with steps 7- 9 for slave-transmitter:

1. Receive first (high) byte of Address (bits SSPIF, BF and bit UA (SSPSTAT<1>) are set).
2. Update the SSPADD register with second (low) byte of Address (clears bit UA and releases the SCL line).
3. Read the SSPBUF register (clears bit BF) and clear flag bit SSPIF.
4. Receive second (low) byte of Address (bits SSPIF, BF and UA are set).
5. Update the SSPADD register with the first (high) byte of Address. This will clear bit UA and release the SCL line.
6. Read the SSPBUF register (clears bit BF) and clear flag bit SSPIF.
7. Receive Repeated Start condition.
8. Receive first (high) byte of Address (bits SSPIF and BF are set).
9. Read the SSPBUF register (clears bit BF) and clear flag bit SSPIF.

Note:	Following the Repeated Start condition (step 7) in 10-bit mode, the user only needs to match the first 7-bit address. The user does not update the SSPADD for the second half of the address.

15.2.1.2 Slave Reception

When the R/$\overline{W}$ bit of the address byte is clear and an address match occurs, the R/$\overline{W}$ bit of the SSPSTAT register is cleared. The received address is loaded into the SSPBUF register.

When the address byte overflow condition exists, then no acknowledge ($\overline{ACK}$) pulse is given. An overflow condition is defined as either bit BF (SSPSTAT<0>) is set, or bit SSPOV (SSPCON1<6>) is set.

An SSP interrupt is generated for each data transfer byte. Flag bit SSPIF (PIR2<7>) must be cleared in software. The SSPSTAT register is used to determine the status of the received byte.

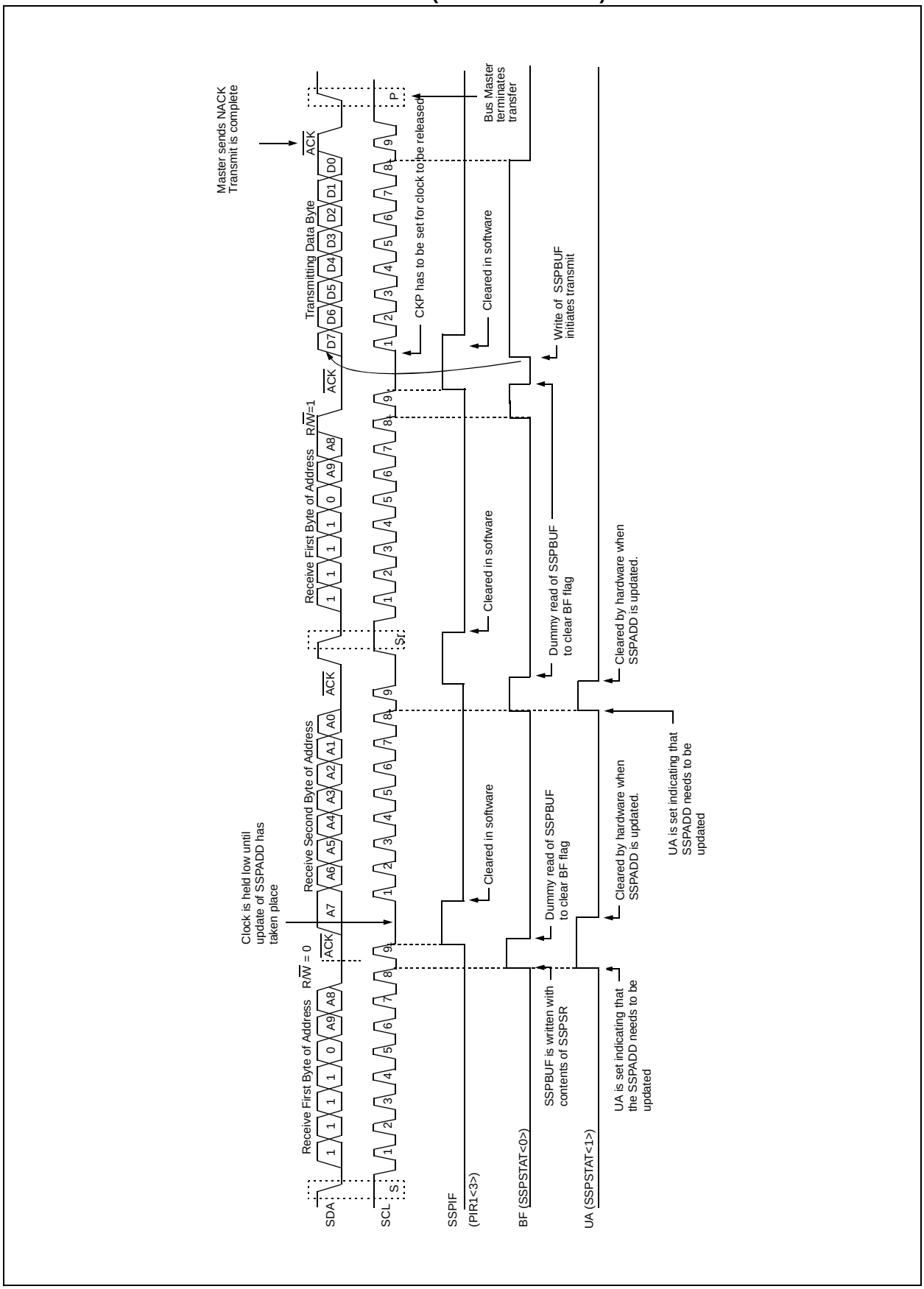
Note:	The SSPBUF will be loaded if the SSPOV bit is set and the BF flag is cleared. If a read of the SSPBUF was performed, but the user did not clear the state of the SSPOV bit before the next receive occurred, the $\overline{ACK}$ is not sent and the SSPBUF is updated.

TABLE 15-2: DATA TRANSFER RECEIVED BYTE ACTIONS

Status Bits as Data Transfer is Received		SSPSR → SSPBUF	Generate $\overline{ACK}$ Pulse	Set bit SSPIF (SSP Interrupt occurs if enabled)
BF	SSPOV			
0	0	Yes	Yes	Yes
1	0	No	No	Yes
1	1	No	No	Yes
0	1	Yes	No	Yes

Note 1: Shaded cells show the conditions where the user software did not properly clear the overflow condition.

FIGURE 15-14: I²C SLAVE-TRANSMITTER (10-BIT ADDRESS)



15.4 Example Program

Example 15-2 shows MPLAB® C17 'C' code for using the I^2C 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();
    }
}
```

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

```
void ACKPoll(void)
{
        StartI2C();                     // Send start bit
        IdleI2C();                      // Wait for idle condition
        WriteI2C(CONTROL);              // Send control byte
        IdleI2C();                      // Wait for idle condition
        // Poll the ACK bit coming from the 24LC01B
        // Loop as long as the 24LC01B NACKs
        while (SSPCON2bits.ACKSTAT)
        {
                RestartI2C();           // Send a restart bit
                IdleI2C();              // Wait for idle condition
                WriteI2C(CONTROL);      // Send control byte
                IdleI2C();              // Wait for idle condition
        }
        IdleI2C();                      // Wait for idle condition
        StopI2C();                      // Send stop bit
        IdleI2C();                      // Wait for idle condition
        return;
}
```

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.	U-x	R/P-1	R/P-1	U-x	U-x	U-x	U-x	U-x	U-x
FE0Fh - FE08h	—	PM2	BODEN	—	—	—	—	—	—
	bit 15 bit 8	bit 7							bit 0

Low (L) Table Read Addr.	U-x	U-x	R/P-1	U-x	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
FE07h - FE00h	—	—	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**

17.4.2 MINIMIZING CURRENT CONSUMPTION

To minimize current consumption, all I/O pins should be either at VDD, or VSS, with no external circuitry drawing current from the I/O pin. I/O pins that are hi-impedance inputs should be pulled high or low externally to avoid switching currents caused by floating inputs. The T0CKI input should be at VDD or VSS. The contributions from on-chip pull-ups on PORTB should also be considered and disabled, when possible.

17.5 Code Protection

The code in the program memory can be protected by selecting the microcontroller in Code Protected mode (PM2:PM0 = '000').

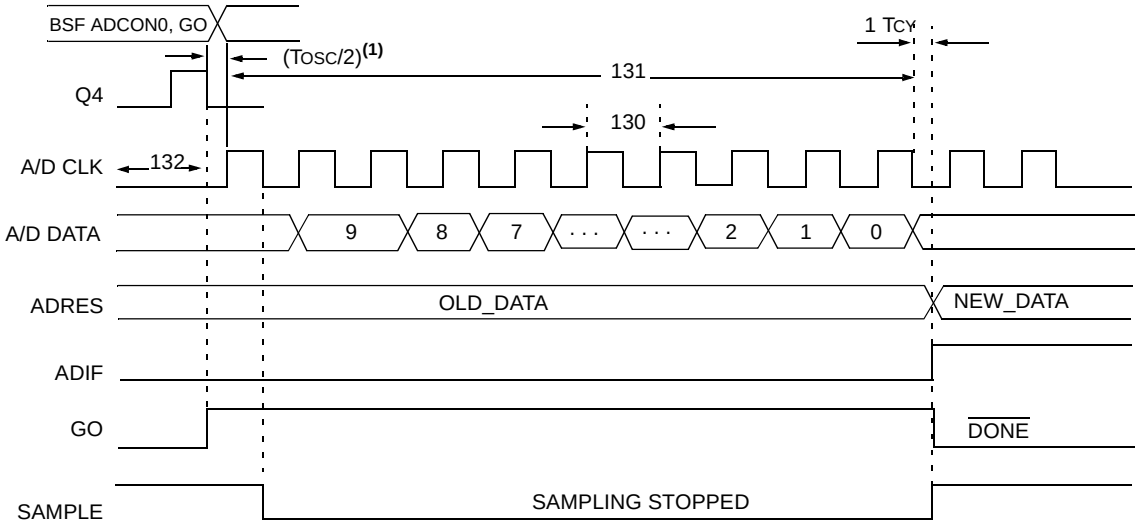
In this mode, instructions that are in the on-chip program memory space, can continue to read or write the program memory. An instruction that is executed outside of the internal program memory range will be inhibited from writing to, or reading from, program memory.

Note:	Microchip does not recommend code protecting windowed devices.

If the code protection bit(s) have not been programmed, the on-chip program memory can be read out for verification purposes.

NOTES:

FIGURE 20-23: A/D CONVERSION TIMING



Note 1: 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.

TABLE 20-19: A/D CONVERSION REQUIREMENTS

Param. No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
130	TAD	A/D clock period	PIC17**C**XXX	1.6	—	—	µs	TOSC based, VREF ≥ 3.0V
			PIC17**LC**XXX	3.0	—	—	µs	TOSC based, VREF full range
			PIC17**C**XXX	2.0	4.0	6.0	µs	A/D RC mode
			PIC17**LC**XXX	3.0	6.0	9.0	µs	A/D RC mode
131	TCNV	Conversion time (not including acquisition time) **(Note 1)**		11	—	12	Tad	
132	TACQ	Acquisition time		**(Note 2)**	20	—	µs	
				10	—	—	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).
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.

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

FIGURE 20-25: MEMORY INTERFACE READ TIMING

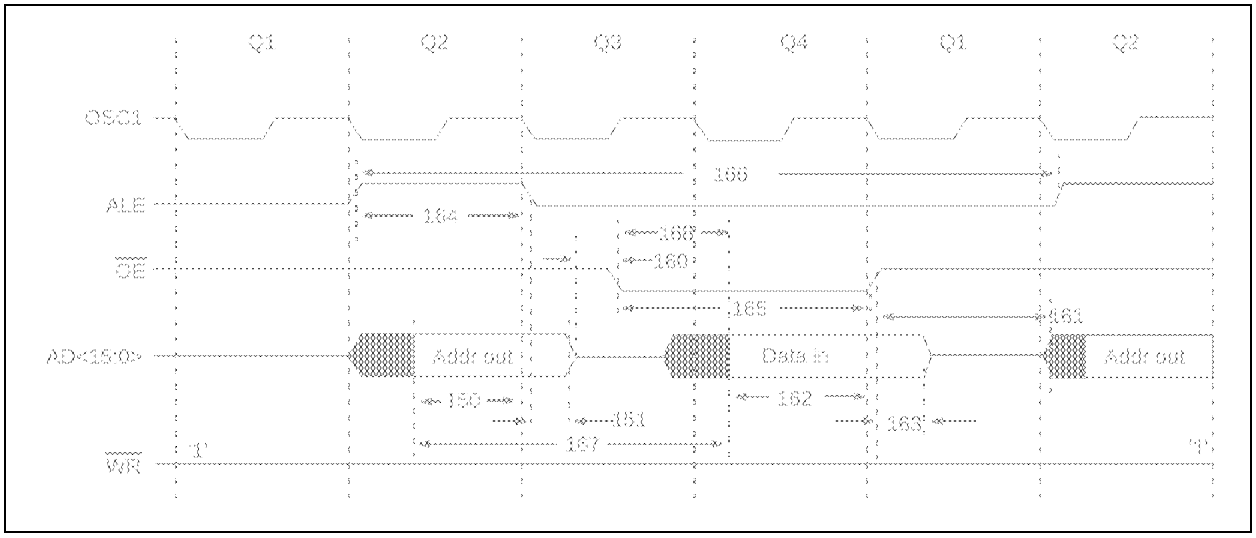


TABLE 20-21: MEMORY INTERFACE READ REQUIREMENTS

Param. No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
150	TadV2alL	AD15:AD0 (address) valid to ALE↓ (address setup time)	PIC17**C**XXX	0.25TCY - 10	—	—	ns	
			PIC17**LC**XXX	0.25TCY - 10	—	—		
151	TalL2adI	ALE↓ to address out invalid (address hold time)	PIC17**C**XXX	5	—	—	ns	
			PIC17**LC**XXX	5	—	—		
160	TadZ2oeL	AD15:AD0 hi-impedance to $\overline{OE}$↓	PIC17**C**XXX	0	—	—	ns	
			PIC17**LC**XXX	0	—	—		
161	ToeH2adD	$\overline{OE}$↑ to AD15:AD0 driven	PIC17**C**XXX	0.25TCY - 15	—	—	ns	
			PIC17**LC**XXX	0.25TCY - 15	—	—		
162	TadV2oeH	Data in valid before $\overline{OE}$↑ (data setup time)	PIC17**C**XXX	35	—	—	ns	
			PIC17**LC**XXX	45	—	—		
163	ToeH2adI	$\overline{OE}$↑to data in invalid (data hold time)	PIC17**C**XXX	0	—	—	ns	
			PIC17**LC**XXX	0	—	—		
164	TalH	ALE pulse width	PIC17**C**XXX	—	0.25TCY	—	ns	
			PIC17**LC**XXX	—	0.25TCY	—		
165	ToeL	$\overline{OE}$ pulse width	PIC17**C**XXX	0.5TCY - 35	—	—	ns	
			PIC17**LC**XXX	0.5TCY - 35	—	—		
166	TalH2alH	ALE↑ to ALE↑(cycle time)	PIC17**C**XXX	—	TCY	—	ns	
			PIC17**LC**XXX	—	TCY	—		
167	Tacc	Address access time	PIC17**C**XXX	—	—	0.75TCY - 30	ns	
			PIC17**LC**XXX	—	—	0.75TCY - 45		
168	Toe	Output enable access time ($\overline{OE}$ low to data valid)	PIC17**C**XXX	—	—	0.5TCY - 45	ns	
			PIC17**LC**XXX	—	—	0.5TCY - 75		

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

FIGURE 21-23: MAXIMUM AND MINIMUM VIN vs. VDD (I^2C Input, -40°C to +125°C)