Welcome to E-XFL.COM

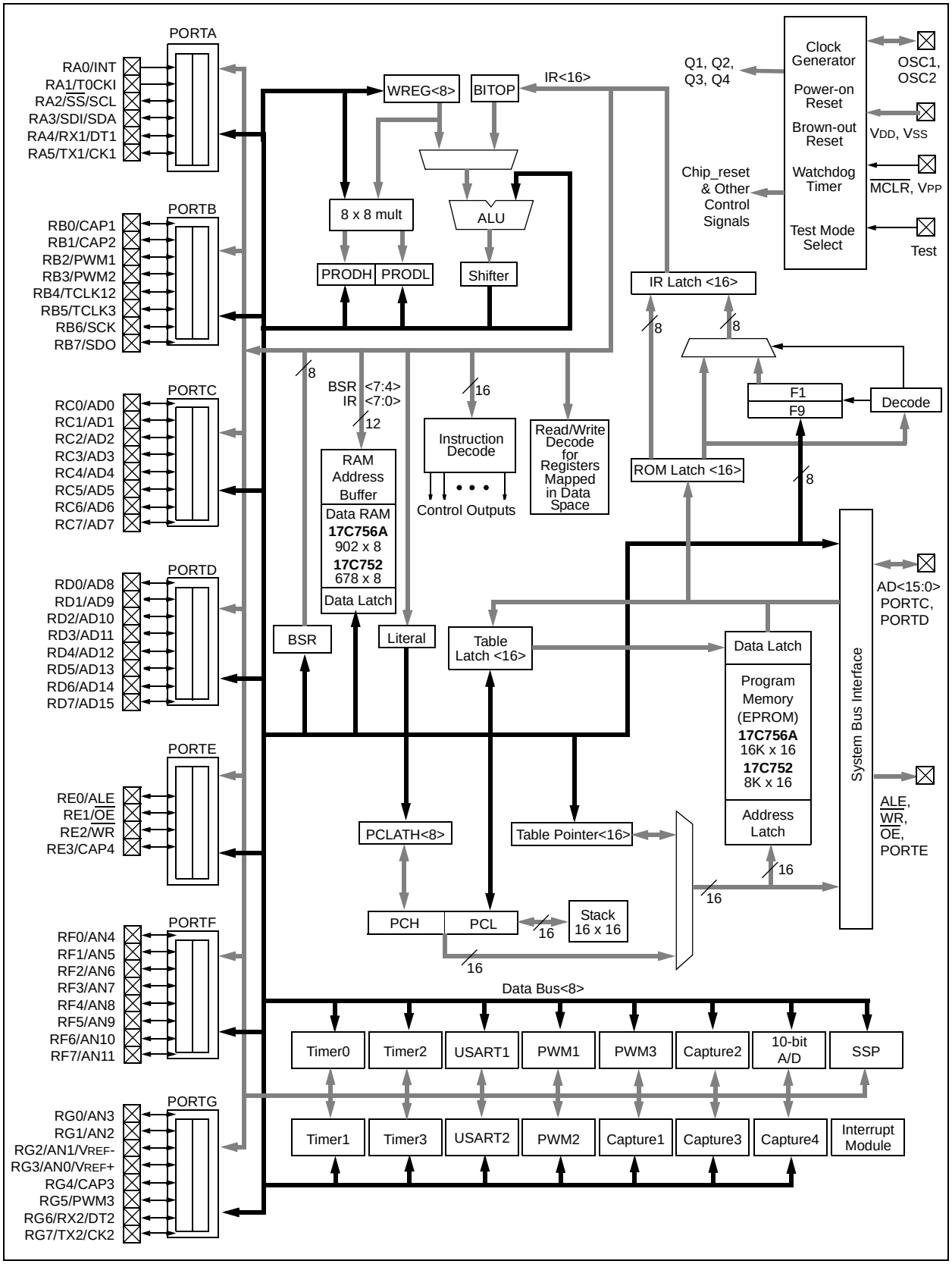
What is "Embedded - Microcontrollers"?

"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	33MHz
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 ~ 125°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/pic17c766-33e-pt

FIGURE 3-1: PIC17C752/756A BLOCK DIAGRAM



6.2 Peripheral Interrupt Enable Register1 (PIE1) and Register2 (PIE2)

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

REGISTER 6-2: PIE1 REGISTER (ADDRESS: 17h, BANK 1)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RBIE	TMR3IE	TMR2IE	TMR1IE	CA2IE	CA1IE	TX1IE	RC1IE
bit 7							bit 0

bit 7 **RBIE**: PORTB Interrupt-on-Change Enable bit
1 = Enable PORTB interrupt-on-change
0 = Disable PORTB interrupt-on-change

bit 6 **TMR3IE**: TMR3 Interrupt Enable bit
1 = Enable TMR3 interrupt
0 = Disable TMR3 interrupt

bit 5 **TMR2IE**: TMR2 Interrupt Enable bit
1 = Enable TMR2 interrupt
0 = Disable TMR2 interrupt

bit 4 **TMR1IE**: TMR1 Interrupt Enable bit
1 = Enable TMR1 interrupt
0 = Disable TMR1 interrupt

bit 3 **CA2IE**: Capture2 Interrupt Enable bit
1 = Enable Capture2 interrupt
0 = Disable Capture2 interrupt

bit 2 **CA1IE**: Capture1 Interrupt Enable bit
1 = Enable Capture1 interrupt
0 = Disable Capture1 interrupt

bit 1 **TX1IE**: USART1 Transmit Interrupt Enable bit
1 = Enable USART1 Transmit buffer empty interrupt
0 = Disable USART1 Transmit buffer empty interrupt

bit 0 **RC1IE**: USART1 Receive Interrupt Enable bit
1 = Enable USART1 Receive buffer full interrupt
0 = Disable USART1 Receive buffer full interrupt

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

EXAMPLE 6-1: SAVING STATUS AND WREG IN RAM (SIMPLE)

```
; The addresses that are used to store the CPUSTA and WREG values must be in the data memory
; address range of 1Ah - 1Fh. Up to 6 locations can be saved and restored using the MOVFP
; instruction. This instruction neither affects the status bits, nor corrupts the WREG register.
;
UNBANK1         EQU    0x01A       ; Address for 1st location to save
UNBANK2         EQU    0x01B       ; Address for 2nd location to save
UNBANK3         EQU    0x01C       ; Address for 3rd location to save
UNBANK4         EQU    0x01D       ; Address for 4th location to save
UNBANK5         EQU    0x01E       ; Address for 5th location to save
                                   ;    (Label Not used in program)
UNBANK6         EQU    0x01F       ; Address for 6th location to save
                                   ;    (Label Not used in program)
;
        :                          ; At Interrupt Vector Address
PUSH    MOVFP   ALUSTA, UNBANK1    ; Push ALUSTA value
        MOVFP   BSR, UNBANK2       ; Push BSR value
        MOVFP   WREG, UNBANK3      ; Push WREG value
        MOVFP   PCLATH, UNBANK4    ; Push PCLATH value
        ;
        :                          ; Interrupt Service Routine (ISR) code
        ;
POP     MOVFP   UNBANK4, PCLATH    ; Restore PCLATH value
        MOVFP   UNBANK3, WREG      ; Restore WREG value
        MOVFP   UNBANK2, BSR       ; Restore BSR value
        MOVFP   UNBANK1, ALUSTA    ; Restore ALUSTA value
;
        RETFIE                     ; Return from interrupt (enable interrupts)
```

13.1.3 USING PULSE WIDTH MODULATION (PWM) OUTPUTS WITH TIMER1 AND TIMER2

Three high speed pulse width modulation (PWM) outputs are provided. The PWM1 output uses Timer1 as its time base, while PWM2 and PWM3 may independently be software configured to use either Timer1 or Timer2 as the time base. The PWM outputs are on the RB2/PWM1, RB3/PWM2 and RG5/PWM3 pins.

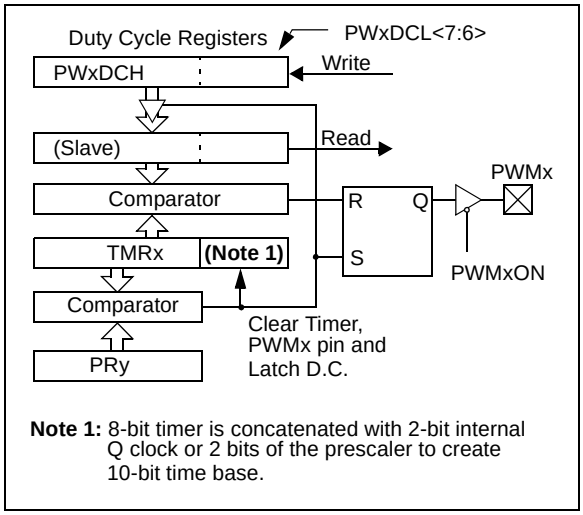
Each PWM output has a maximum resolution of 10-bits. At 10-bit resolution, the PWM output frequency is 32.2 kHz (@ 32 MHz clock) and at 8-bit resolution the PWM output frequency is 128.9 kHz. The duty cycle of the output can vary from 0% to 100%.

Figure 13-3 shows a simplified block diagram of a PWM module.

The duty cycle registers are double buffered for glitch free operation. Figure 13-4 shows how a glitch could occur if the duty cycle registers were not double buffered.

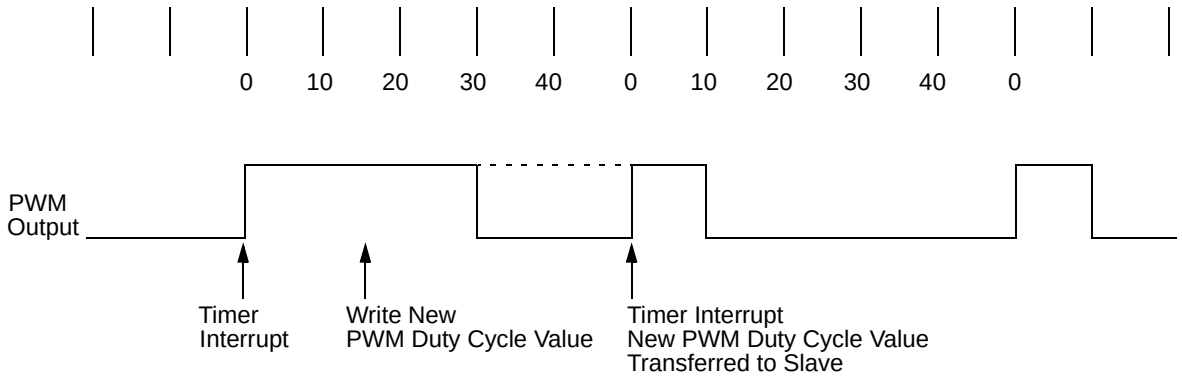
The user needs to set the PWM1ON bit (TCON2<4>) to enable the PWM1 output. When the PWM1ON bit is set, the RB2/PWM1 pin is configured as PWM1 output and forced as an output, irrespective of the data direction bit (DDRB<2>). When the PWM1ON bit is clear, the pin behaves as a port pin and its direction is controlled by its data direction bit (DDRB<2>). Similarly, the PWM2ON (TCON2<5>) bit controls the configuration of the RB3/PWM2 pin and the PWM3ON (TCON3<0>) bit controls the configuration of the RG5/PWM3 pin.

FIGURE 13-3: SIMPLIFIED PWM BLOCK DIAGRAM



Note 1: 8-bit timer is concatenated with 2-bit internal Q clock or 2 bits of the prescaler to create 10-bit time base.

FIGURE 13-4: PWM OUTPUT (NOT BUFFERED)



Note: The dotted line shows PWM output if duty cycle registers were not double buffered. If the new duty cycle is written after the timer has passed that value, then the PWM does not reset at all during the current cycle, causing a "glitch".

In this example, PWM period = 50. Old duty cycle is 30. New duty cycle value is 10.

14.3.2 USART SYNCHRONOUS MASTER RECEPTION

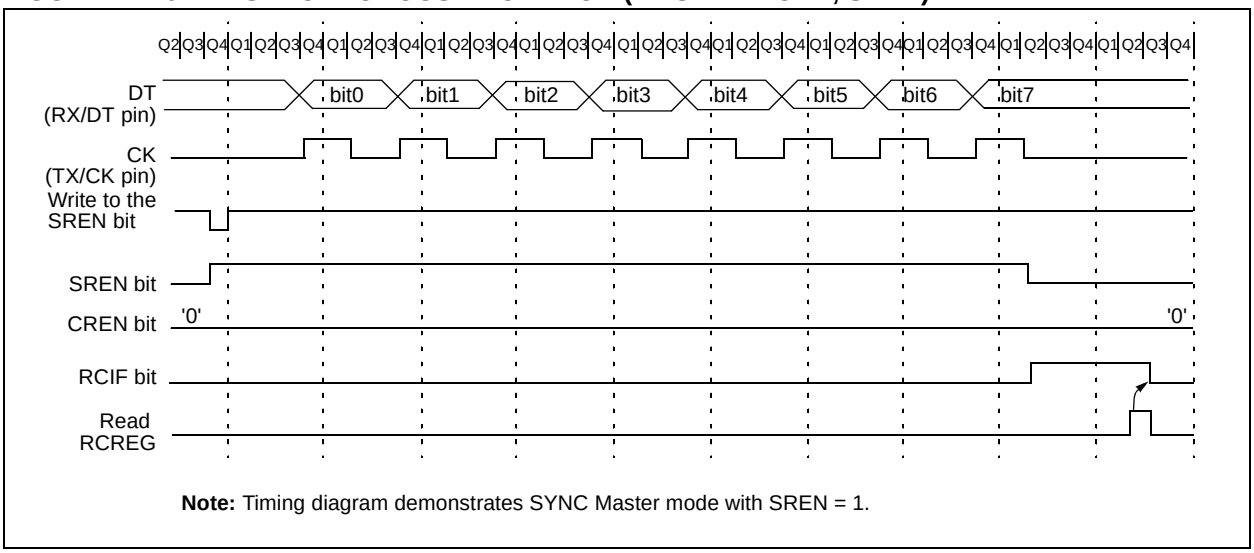
Once Synchronous mode is selected, reception is enabled by setting either the SREN (RCSTA<5>) bit or the CREN (RCSTA<4>) bit. Data is sampled on the RX/DT pin on the falling edge of the clock. If SREN is set, then only a single word is received. If CREN is set, the reception is continuous until CREN is reset. If both bits are set, then CREN takes precedence. After clocking the last bit, the received data in the Receive Shift Register (RSR) is transferred to 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 reset by the hardware. In this case, it is reset 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 to begin shifting into the RSR. On the clocking of the last bit of the third byte, if RCREG is still full, then the overrun error bit OERR (RCSTA<1>) is 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. This is done by clearing the CREN bit. If OERR is set, transfers from RSR to RCREG are inhibited, so it is essential to clear the OERR bit if it is set. The 9th receive bit is 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.

Steps to follow when setting up a Synchronous Master Reception:

1. Initialize the SPBRG register for the appropriate baud rate. See Section 14.1 for details.
2. Enable the synchronous master serial port by setting bits SYNC, SPEN, and CSRC.
3. If interrupts are desired, then set the RCIE bit.
4. If 9-bit reception is desired, then set the RX9 bit.
5. If a single reception is required, set bit SREN. For continuous reception set bit CREN.
6. The RCIF bit will be set when reception is complete and an interrupt will be generated if the RCIE bit was set.
7. Read RCSTA to get the ninth bit (if enabled) and determine if any error occurred during reception.
8. Read the 8-bit received data by reading RCREG.
9. If any error occurred, clear the error by clearing CREN.

Note:	To terminate a reception, either clear the SREN and CREN bits, or the SPEN bit. This will reset the receive logic so that it will be in the proper state when receive is re-enabled.

FIGURE 14-10: SYNCHRONOUS RECEPTION (MASTER MODE, SREN)



Note: Timing diagram demonstrates SYNC Master mode with SREN = 1.

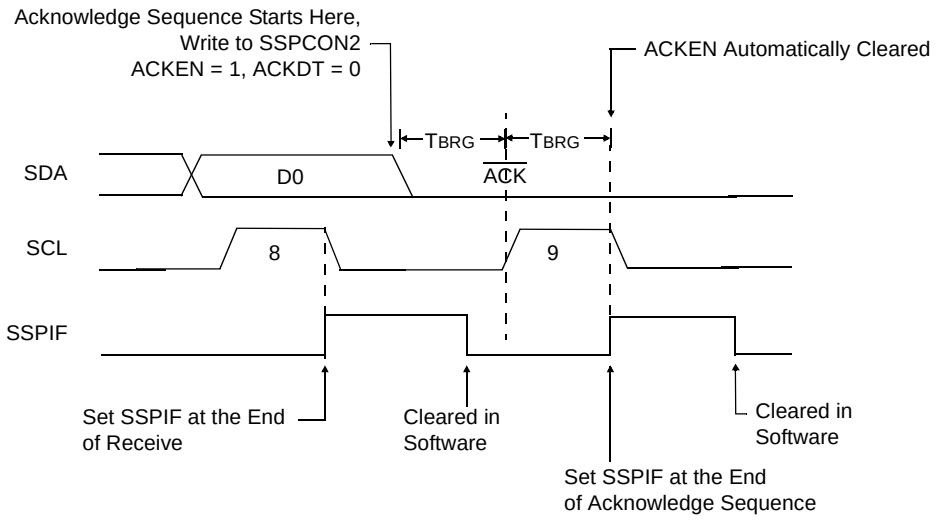
15.2.13 ACKNOWLEDGE SEQUENCE TIMING

An acknowledge sequence is enabled by setting the acknowledge sequence enable bit, ACKEN (SSPCON2<4>). When this bit is set, the SCL pin is pulled low and the contents of the acknowledge data bit is presented on the SDA pin. If the user wishes to generate an acknowledge, then the ACKDT bit should be cleared. If not, the user should set the ACKDT bit before starting an acknowledge sequence. The baud rate generator then counts for one rollover period (TBRG), and the SCL pin is de-asserted (pulled high). When the SCL pin is sampled high (clock arbitration), the baud rate generator counts for TBRG. The SCL pin is then pulled low. Following this, the ACKEN bit is automatically cleared, the baud rate generator is turned off and the SSP module then goes into IDLE mode (Figure 15-29).

15.2.13.1 WCOL Status Flag

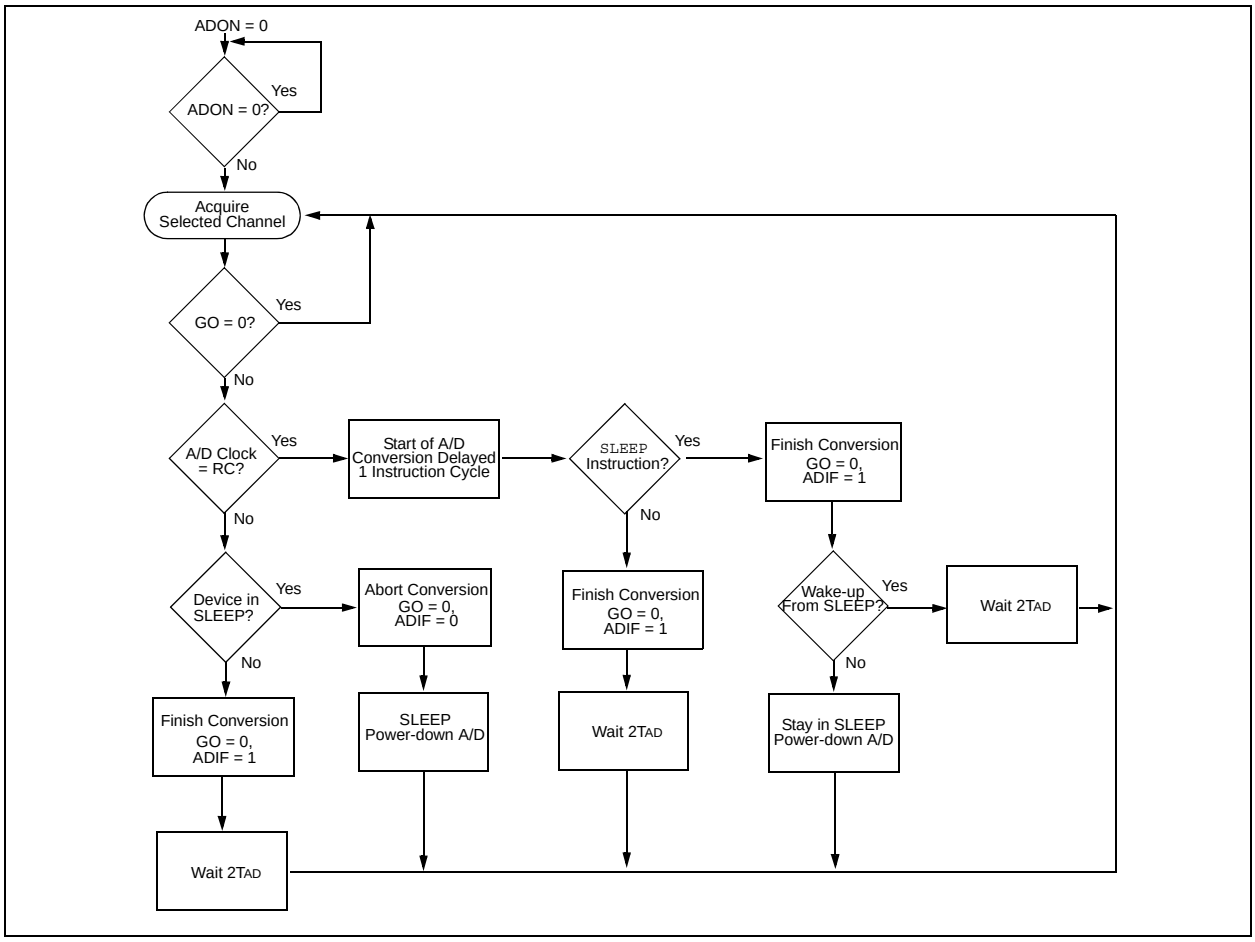
If the user writes the SSPBUF when an acknowledge sequence is in progress, then WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).

FIGURE 15-29: ACKNOWLEDGE SEQUENCE WAVEFORM



Note: TBRG = one baud rate generator period.

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



17.3 Watchdog Timer (WDT)

The Watchdog Timer's function is to recover from software malfunction, or to reset the device while in SLEEP mode. The WDT uses an internal free running on-chip RC oscillator for its clock source. This does not require any external components. This RC oscillator is separate from the RC oscillator of the OSC1/CLKIN pin. That means that the WDT will run even if the clock on the OSC1/CLKIN and OSC2/CLKOUT pins has been stopped, for example, by execution of a SLEEP instruction. During normal operation, a WDT time-out generates a device RESET. The WDT can be permanently disabled by programming the configuration bits WDTPS1:WDTPS0 as '00' (Section 17.1).

Under normal operation, the WDT must be cleared on a regular interval. This time must be less than the minimum WDT overflow time. Not clearing the WDT in this time frame will cause the WDT to overflow and reset the device.

17.3.1 WDT PERIOD

The WDT has a nominal time-out period of 12 ms (with postscaler = 1). The time-out periods vary with temperature, VDD and process variations from part to part (see DC specs). If longer time-out periods are desired, configuration bits should be used to enable the WDT with a greater prescale. Thus, typical time-out periods up to 3.0 seconds can be realized.

The CLRWDT and SLEEP instructions clear the WDT and its postscale setting and prevent it from timing out, thus generating a device RESET condition.

The $\overline{TO}$ bit in the CPUSTA register will be cleared upon a WDT time-out.

17.3.2 CLEARING THE WDT AND POSTSCALER

The WDT and postscaler are cleared when:

- The device is in the RESET state
- A SLEEP instruction is executed
- A CLRWDT instruction is executed
- Wake-up from SLEEP by an interrupt

The WDT counter/postscaler will start counting on the first edge after the device exits the RESET state.

17.3.3 WDT PROGRAMMING CONSIDERATIONS

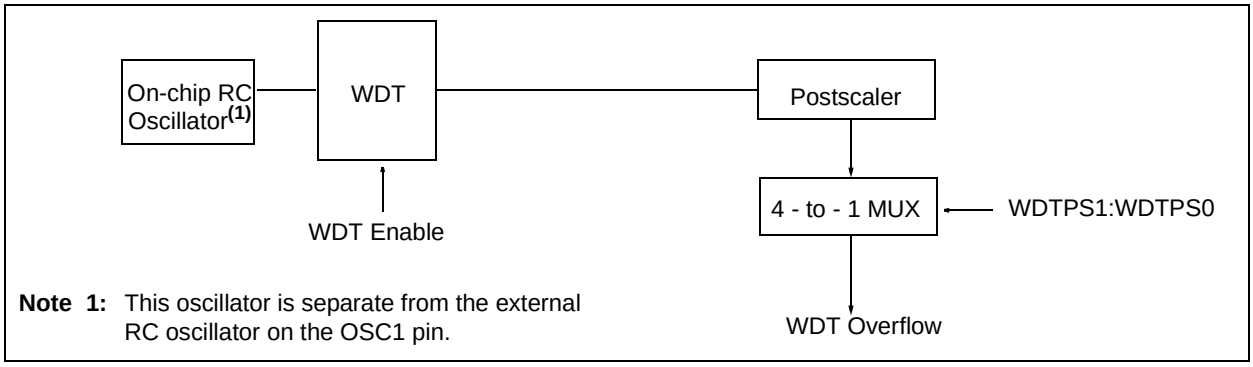
It should also be taken in account that under worst case conditions (VDD = Min., Temperature = Max., Max. WDT postscaler), it may take several seconds before a WDT time-out occurs.

The WDT and postscaler become the Power-up Timer whenever the PWRT is invoked.

17.3.4 WDT AS NORMAL TIMER

When the WDT is selected as a normal timer, the clock source is the device clock. Neither the WDT nor the postscaler are directly readable or writable. The overflow time is 65536 TOSC cycles. On overflow, the $\overline{TO}$ bit is cleared (device is not RESET). The CLRWDT instruction can be used to set the $\overline{TO}$ bit. This allows the WDT to be a simple overflow timer. The simple timer does not increment when in SLEEP.

FIGURE 17-1: WATCHDOG TIMER BLOCK DIAGRAM



Note 1: This oscillator is separate from the external RC oscillator on the OSC1 pin.

TABLE 17-2: REGISTERS/BITS ASSOCIATED WITH THE WATCHDOG TIMER

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	$\overline{MCLR}$, WDT
—	Config	See Figure 17-1 for location of WDTPSx bits in Configuration Word.								(Note 1)	(Note 1)
06h, Unbanked	CPUSTA	—	—	STKAV	GLINTD	$\overline{TO}$	$\overline{PD}$	$\overline{POR}$	$\overline{BOR}$	--11 11qq	--11 qquu

Legend: - = unimplemented, read as '0', q = value depends on condition. Shaded cells are not used by the WDT.
Note 1: This value will be as the device was programmed, or if unprogrammed, will read as all '1's.

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.

17.6 In-Circuit Serial Programming

The PIC17C7XX group of the high-end family (PIC17CXXX) has an added feature that allows serial programming while in the end application circuit. This is simply done with two lines for clock and data and three other lines for power, ground, and the programming voltage. This allows customers to manufacture boards with unprogrammed devices and then program the microcontroller just before shipping the product. This also allows the most recent firmware, or a custom firmware to be programmed.

Devices may be serialized to make the product unique; "special" variants of the product may be offered and code updates are possible. This allows for increased design flexibility.

To place the device into the Serial Programming Test mode, two pins will need to be placed at V_{IHH}. These are the TEST pin and the $\overline{MCLR}/V_{PP}$ pin. Also, a sequence of events must occur as follows:

1. The TEST pin is placed at V_{IHH}.
2. The $\overline{MCLR}/V_{PP}$ pin is placed at V_{IHH}.

There is a setup time between step 1 and step 2 that must be met.

After this sequence, the Program Counter is pointing to program memory address 0xFF60. This location is in the Boot ROM. The code initializes the USART/SCI so that it can receive commands. For this, the device must be clocked. The device clock source in this mode is the RA1/T0CKI pin. After delaying to allow the USART/SCI to initialize, commands can be received. The flow is shown in these 3 steps:

1. The device clock source starts.
2. Wait 80 device clocks for Boot ROM code to configure the USART/SCI.
3. Commands may now be sent.

For complete details of serial programming, please refer to the PIC17C7XX Programming Specification. (Contact your local Microchip Technology Sales Office for availability.)

FIGURE 17-3: TYPICAL IN-CIRCUIT SERIAL PROGRAMMING CONNECTION

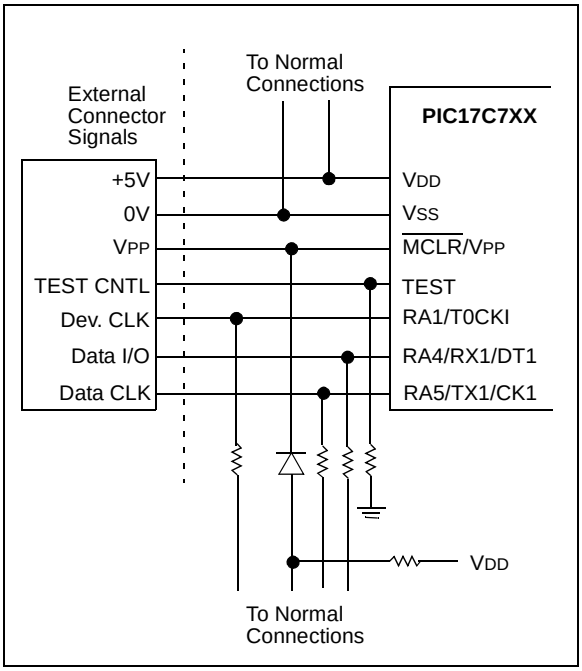


TABLE 17-3: ICSP INTERFACE PINS

Name	During Programming		
	Function	Type	Description
RA4/RX1/DT1	DT	I/O	Serial Data
RA5/TX1/CK1	CK	I	Serial Clock
RA1/T0CKI	OSCI	I	Device Clock Source
TEST	TEST	I	Test mode selection control input, force to V_{IHH}
$\overline{MCLR}/V_{PP}$	$\overline{MCLR}/V_{PP}$	P	Master Clear Reset and Device Programming Voltage
V_{DD}	V_{DD}	P	Positive supply for logic and I/O pins
V_{SS}	V_{SS}	P	Ground reference for logic and I/O pins

18.2 Q Cycle Activity

Each instruction cycle (TCY) is comprised of four Q cycles (Q1-Q4). The Q cycle is the same as the device oscillator cycle (TOSC). The Q cycles provide the timing/ designation for the Decode, Read, Process Data, Write, etc., of each instruction cycle. The following diagram shows the relationship of the Q cycles to the instruction cycle.

The four Q cycles that make up an instruction cycle (TCY) can be generalized as:

Q1: Instruction Decode Cycle or forced No operation

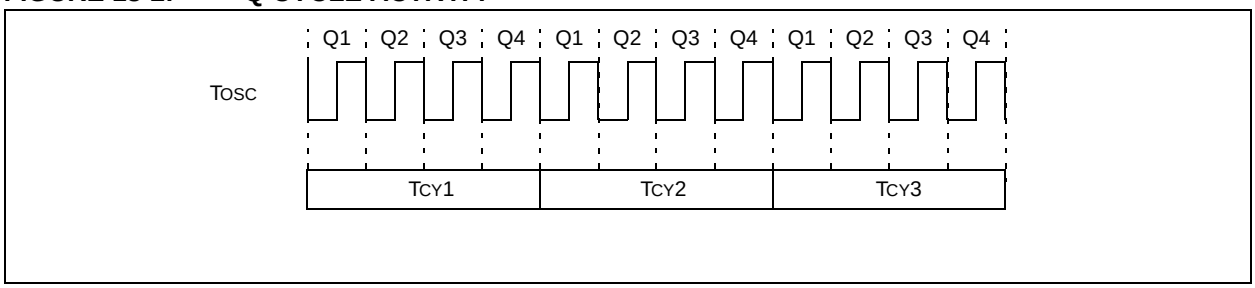
Q2: Instruction Read Cycle or No operation

Q3: Process the Data

Q4: Instruction Write Cycle or No operation

Each instruction will show the detailed Q cycle operation for the instruction.

FIGURE 18-2: Q CYCLE ACTIVITY



MOVLR — **Move Literal to high nibble in BSR**

Syntax: [*label*] MOVLR k

Operands: $0 \le k \le 15$

Operation: $k \rightarrow (BSR<7:4>)$

Status Affected: None

Encoding:

1011	101x	kkkk	uuuu

Description: The 4-bit literal 'k' is loaded into the most significant 4-bits of the Bank Select Register (BSR). Only the high 4-bits of the Bank Select Register are affected. The lower half of the BSR is unchanged. The assembler will encode the “u” fields as 0.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process Data	Write literal 'k' to BSR<7:4>

Example: `MOVLR 5`

Before Instruction
BSR register = 0x22

After Instruction
BSR register = 0x52

MOVLW — **Move Literal to WREG**

Syntax: [*label*] MOVLW k

Operands: $0 \le k \le 255$

Operation: $k \rightarrow (WREG)$

Status Affected: None

Encoding:

1011	0000	kkkk	kkkk

Description: The eight-bit literal 'k' is loaded into WREG.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process Data	Write to WREG

Example: `MOVLW 0x5A`

After Instruction
WREG = 0x5A

SLEEP **Enter SLEEP mode**

Syntax: [*label*] SLEEP

Operands: None

Operation: 00h → WDT;
0 → WDT postscaler;
1 → $\overline{TO}$;
0 → $\overline{PD}$

Status Affected: $\overline{TO}$, $\overline{PD}$

Encoding:

0000	0000	0000	0011

Description: The power-down status bit ($\overline{PD}$) is cleared. The time-out status bit ($\overline{TO}$) is set. Watchdog Timer and its post-scaler are cleared.
The processor is put into SLEEP mode with the oscillator stopped.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	Process Data	Go to sleep

Example: `SLEEP`

Before Instruction
$\overline{TO}$ = ?
$\overline{PD}$ = ?

After Instruction
$\overline{TO}$ = 1 †
$\overline{PD}$ = 0

† If WDT causes wake-up, this bit is cleared

SUBLW **Subtract WREG from Literal**

Syntax: [*label*] SUBLW k

Operands: $0 \le k \le 255$

Operation: k – (WREG) → (WREG)

Status Affected: OV, C, DC, Z

Encoding:

1011	0010	kkkk	kkkk

Description: WREG is subtracted from the eight-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 1: `SUBLW 0x02`

Before Instruction
WREG = 1
C = ?

After Instruction
WREG = 1
C = 1 ; result is positive
Z = 0

Example 2:

Before Instruction
WREG = 2
C = ?

After Instruction
WREG = 0
C = 1 ; result is zero
Z = 1

Example 3:

Before Instruction
WREG = 3
C = ?

After Instruction
WREG = FF ; (2's complement)
C = 0 ; result is negative
Z = 0

19.8 MPLAB ICD In-Circuit Debugger

Microchip's In-Circuit Debugger, MPLAB ICD, is a powerful, low cost, run-time development tool. This tool is based on the FLASH PIC16F87X and can be used to develop for this and other PIC microcontrollers from the PIC16CXXX family. The MPLAB ICD utilizes the in-circuit debugging capability built into the PIC16F87X. This feature, along with Microchip's In-Circuit Serial Programming™ protocol, offers cost-effective in-circuit FLASH debugging from the graphical user interface of the MPLAB Integrated Development Environment. This enables a designer to develop and debug source code by watching variables, single-stepping and setting break points. Running at full speed enables testing hardware in real-time.

19.9 PRO MATE II Universal Device Programmer

The PRO MATE II universal device programmer is a full-featured programmer, capable of operating in stand-alone mode, as well as PC-hosted mode. The PRO MATE II device programmer is CE compliant.

The PRO MATE II device programmer has programmable V_{DD} and V_{PP} supplies, which allow it to verify programmed memory at V_{DD} min and V_{DD} max for maximum reliability. It has an LCD display for instructions and error messages, keys to enter commands and a modular detachable socket assembly to support various package types. In stand-alone mode, the PRO MATE II device programmer can read, verify, or program PIC MCU devices. It can also set code protection in this mode.

19.10 PICSTART Plus Entry Level Development Programmer

The PICSTART Plus development programmer is an easy-to-use, low cost, prototype programmer. It connects to the PC via a COM (RS-232) port. MPLAB Integrated Development Environment software makes using the programmer simple and efficient.

The PICSTART Plus development programmer supports all PIC devices with up to 40 pins. Larger pin count devices, such as the PIC16C92X and PIC17C76X, may be supported with an adapter socket. The PICSTART Plus development programmer is CE compliant.

19.11 PICDEM 1 Low Cost PIC MCU Demonstration Board

The PICDEM 1 demonstration board is a simple board which demonstrates the capabilities of several of Microchip's microcontrollers. The microcontrollers supported are: PIC16C5X (PIC16C54 to PIC16C58A), PIC16C61, PIC16C62X, PIC16C71, PIC16C8X, PIC17C42, PIC17C43 and PIC17C44. All necessary hardware and software is included to run basic demo programs. The user can program the sample microcontrollers provided with the PICDEM 1 demonstration board on a PRO MATE II device programmer, or a PICSTART Plus development programmer, and easily test firmware. The user can also connect the PICDEM 1 demonstration board to the MPLAB ICE in-circuit emulator and download the firmware to the emulator for testing. A prototype area is available for the user to build some additional hardware and connect it to the microcontroller socket(s). Some of the features include an RS-232 interface, a potentiometer for simulated analog input, push button switches and eight LEDs connected to PORTB.

19.12 PICDEM 2 Low Cost PIC16CXX Demonstration Board

The PICDEM 2 demonstration board is a simple demonstration board that supports the PIC16C62, PIC16C64, PIC16C65, PIC16C73 and PIC16C74 microcontrollers. All the necessary hardware and software is included to run the basic demonstration programs. The user can program the sample microcontrollers provided with the PICDEM 2 demonstration board on a PRO MATE II device programmer, or a PICSTART Plus development programmer, and easily test firmware. The MPLAB ICE in-circuit emulator may also be used with the PICDEM 2 demonstration board to test firmware. A prototype area has been provided to the user for adding additional hardware and connecting it to the microcontroller socket(s). Some of the features include a RS-232 interface, push button switches, a potentiometer for simulated analog input, a serial EEPROM to demonstrate usage of the I^2C™ bus and separate headers for connection to an LCD module and a keypad.

FIGURE 20-7: CLKOUT AND I/O TIMING

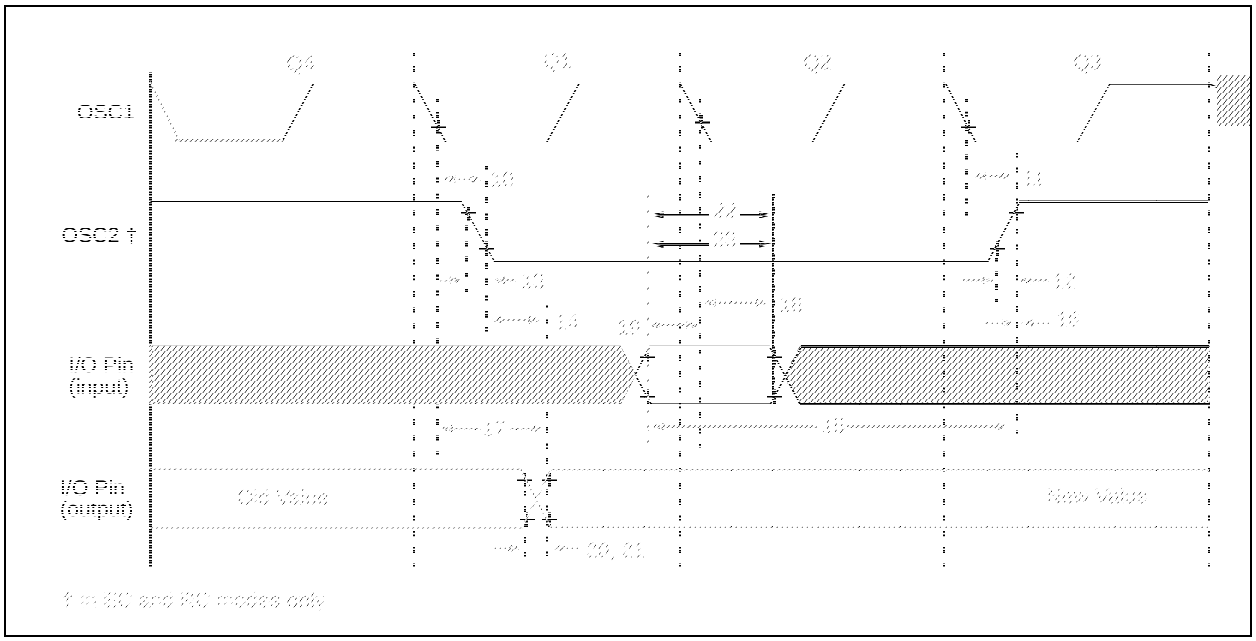


TABLE 20-2: CLKOUT AND I/O TIMING REQUIREMENTS

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
10	TosL2ckL	OSC1↓ to CLKOUT↓	—	15	30	ns	**(Note 1)**
11	TosL2ckH	OSC1↓ to CLKOUT↑	—	15	30	ns	**(Note 1)**
12	TckR	CLKOUT rise time	—	5	15	ns	**(Note 1)**
13	TckF	CLKOUT fall time	—	5	15	ns	**(Note 1)**
14	TckH2ioV	CLKOUT ↑ to Port out valid	—	—	0.5TCY + 20	ns	**(Note 1)**
15	TioV2ckH	Port in valid before CLKOUT↑	0.25TCY + 25	—	—	ns	**(Note 1)**
16	TckH2ioI	Port in hold after CLKOUT↑	0	—	—	ns	**(Note 1)**
17	TosL2ioV	OSC1↓ (Q1 cycle) to Port out valid	—	—	100	ns	
18	TosL2ioI	OSC1↓ (Q2 cycle) to Port input invalid (I/O in hold time)	0	—	—	ns	
19	TioV2osL	Port input valid to OSC1↓ (I/O in setup time)	30	—	—	ns	
20	TioR	Port output rise time	—	10	35	ns	
21	TioF	Port output fall time	—	10	35	ns	
22	TinHL	INT pin high or low time	25	—	—	ns	
23	TrbHL	RB7:RB0 change INT high or low time	25	—	—	ns	

† Data in “Typ” column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: Measurements are taken in EC mode, where CLKOUT output is 4 x TOSC.

TABLE 20-18: A/D CONVERTER CHARACTERISTICS

Param. No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
A01	NR	Resolution		—	—	10	bit	VREF+ = VDD = 5.12V, VSS ≤ VAIN ≤ VREF+
				—	—	10	bit	(VREF+ — VREF-) ≥ 3.0V, VREF- ≤ VAIN ≤ VREF+
A02	EABS	Absolute error		—	—	< ±1	LSb	VREF+ = VDD = 5.12V, VSS ≤ VAIN ≤ VREF+
				—	—	< ±1	LSb	(VREF+ — VREF-) ≥ 3.0V, VREF- ≤ VAIN ≤ VREF+
A03	EIL	Integral linearity error		—	—	< ±1	LSb	VREF+ = VDD = 5.12V, VSS ≤ VAIN ≤ VREF+
				—	—	< ±1	LSb	(VREF+ — VREF-) ≥ 3.0V, VREF- ≤ VAIN ≤ VREF+
A04	EDL	Differential linearity error		—	—	< ±1	LSb	VREF+ = VDD = 5.12V, VSS ≤ VAIN ≤ VREF+
				—	—	< ±1	LSb	(VREF+ — VREF-) ≥ 3.0V, VREF- ≤ VAIN ≤ VREF+
A05	EFS	Full scale error		—	—	< ±1	LSb	VREF+ = VDD = 5.12V, VSS ≤ VAIN ≤ VREF+
				—	—	< ±1	LSb	(VREF+ — VREF-) ≥ 3.0V, VREF- ≤ VAIN ≤ VREF+
A06	EOFF	Offset error		—	—	< ±1	LSb	VREF+ = VDD = 5.12V, VSS ≤ VAIN ≤ VREF+
				—	—	< ±1	LSb	(VREF+ — VREF-) ≥ 3.0V, VREF- ≤ VAIN ≤ VREF+
A10	—	Monotonicity		—	guaranteed(3)	—	—	VSS ≤ VAIN ≤ VREF
A20	VREF	Reference voltage (VREF+ — VREF-)		0V	—	—	V	VREF delta when changing voltage levels on VREF inputs
A20A				3V	—	—	V	Absolute minimum electrical spec. to ensure 10-bit accuracy
A21	VREF+	Reference voltage high		AVSS + 3.0V	—	AVDD + 0.3V	V	
A22	VREF-	Reference voltage low		Avss - 0.3V	—	AVDD - 3.0V	V	
A25	VAIN	Analog input voltage		AVSS - 0.3V	—	Vref + 0.3V	V	
A30	ZAIN	Recommended impedance of analog voltage source		—	—	10.0	kΩ	
A40	IAD	A/D conversion current (VDD)	PIC17**C**XXX	—	180	—	μA	Average current consumption when A/D is on **(Note 1)**
			PIC17**LC**XXX	—	90	—	μA	
A50	IREF	VREF input current **(Note 2)**		10	—	1000	μA	During VAIN acquisition. Based on differential of VHOLD to VAIN
				—	—	10	μA	During A/D conversion cycle

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

Note 1: When A/D is off, it will not consume any current other than minor leakage current. The power-down current spec includes any such leakage from the A/D module.

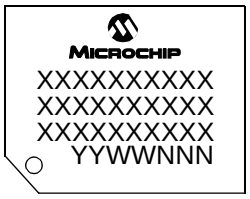
2: VREF current is from RG0 and RG1 pins or AVDD and AVSS pins, whichever is selected as reference input.

3: The A/D conversion result never decreases with an increase in the Input Voltage and has no missing codes.

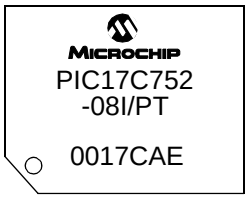
22.0 PACKAGING INFORMATION

22.1 Package Marking Information

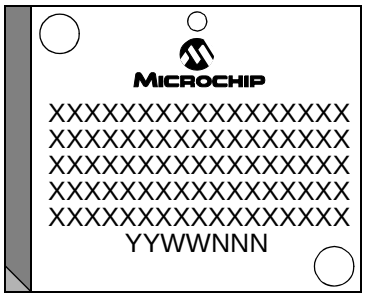
64-Lead TQFP



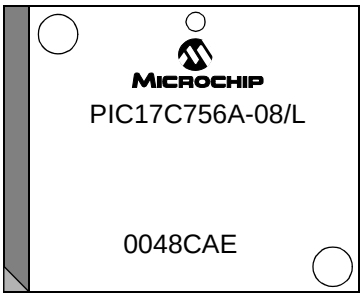
Example



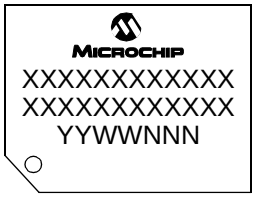
68-Lead PLCC



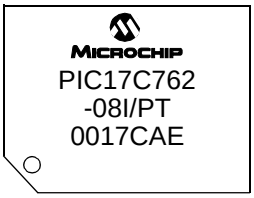
Example



80-Lead TQFP



Example



Legend:	XX...X	Customer-specific information
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week ‘01’)
	NNN	Alphanumeric traceability code
	(e3)	Pb-free JEDEC designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free JEDEC designator ((e3)) can be found on the outer packaging for this package.

Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

PART NO.	X	/XX	XXX
Device	Temperature Range	Package	Pattern

Device	PIC17C756: Standard VDD range PIC17C756T: (Tape and Reel) PIC17LC756: Extended VDD range
Temperature Range	- = 0°C to +70°C I = -40°C to +85°C
Package	CL = Windowed LCC PT = TQFP L = PLCC
Pattern	QTP, SQTP, ROM Code (factory specified) or Special Requirements . Blamk for OTP and Windowed devices.

Examples:

a) PIC17C756 – 16L Commercial Temp., PLCC package, 16 MHz, normal VDD limits

b) PIC17LC756–08/PT Commercial Temp., TQFP package, 8MHz, extended VDD limits

c) PIC17C756–33I/PT Industrial Temp., TQFP package, 33 MHz, normal VDD limits

* JW Devices are UV erasable and can be programmed to any device configuration. JW Devices meet the electrical requirement of each oscillator type.

Sales and Support

Data Sheets
Products supported by a preliminary Data Sheet may have an errata sheet describing minor operational differences and recommended workarounds. To determine if an errata sheet exists for a particular device, please contact one of the following:

1. Your local Microchip sales office
2. The Microchip Corporate Literature Center U.S. FAX: (480) 792-7277
3. The Microchip Worldwide Site (www.microchip.com)

Please specify which device, revision of silicon and Data Sheet (include Literature #) you are using.

New Customer Notification System
Register on our web site (www.microchip.com/cn) to receive the most current information on our products.

NOTES: