Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	8MHz
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)	3V ~ 5.5V
Data Converters	A/D 12x10b
Oscillator Type	External
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	68-LCC (J-Lead)
Supplier Device Package	68-PLCC (24.23x24.23)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic17lc756a-08-l

1.0 OVERVIEW

This data sheet covers the PIC17C7XX group of the PIC17CXXX family of microcontrollers. The following devices are discussed in this data sheet:

- PIC17C752
- PIC17C756A
- PIC17C762
- PIC17C766

The PIC17C7XX devices are 68/84-pin, EPROM based members of the versatile PIC17CXXX family of low cost, high performance, CMOS, fully static, 8-bit microcontrollers.

All PIC® microcontrollers employ an advanced RISC architecture. The PIC17CXXX has enhanced core features, 16-level deep stack, and multiple internal and external interrupt sources. The separate instruction and data buses of the Harvard architecture allow a 16-bit wide instruction word with a separate 8-bit wide data path. The two stage instruction pipeline allows all instructions to execute in a single cycle, except for program branches (which require two cycles). A total of 58 instructions (reduced instruction set) are available. Additionally, a large register set gives some of the architectural innovations used to achieve a very high performance. For mathematical intensive applications, all devices have a single cycle 8 x 8 Hardware Multiplier.

PIC17CXXX microcontrollers typically achieve a 2:1 code compression and a 4:1 speed improvement over other 8-bit microcontrollers in their class.

PIC17C7XX devices have up to 902 bytes of RAM and 66 I/O pins. In addition, the PIC17C7XX adds several peripheral features, useful in many high performance applications, including:

- Four timer/counters
- Four capture inputs
- Three PWM outputs
- Two independent Universal Synchronous Asynchronous Receiver Transmitters (USARTs)
- An A/D converter (multi-channel, 10-bit resolution)
- A Synchronous Serial Port (SPI and I^2C w/ Master mode)

These special features reduce external components, thus reducing cost, enhancing system reliability and reducing power consumption.

There are four oscillator options, of which the single pin RC oscillator provides a low cost solution, the LF oscillator is for low frequency crystals and minimizes power consumption, XT is a standard crystal and the EC is for external clock input.

The SLEEP (power-down) mode offers additional power saving. Wake-up from SLEEP can occur through several external and internal interrupts and device RESETS.

A highly reliable Watchdog Timer with its own on-chip RC oscillator provides protection against software malfunction.

There are four configuration options for the device operational mode:

- Microprocessor
- Microcontroller
- Extended microcontroller
- Protected microcontroller

The microprocessor and extended microcontroller modes allow up to 64K-words of external program memory.

The device also has Brown-out Reset circuitry. This allows a device RESET to occur if the device V_{DD} falls below the Brown-out voltage trip point (BV_{DD}). The chip will remain in Brown-out Reset until V_{DD} rises above BV_{DD}.

A UV erasable, CERQUAD packaged version (compatible with PLCC), is ideal for code development, while the cost-effective One-Time-Programmable (OTP) version is suitable for production in any volume.

The PIC17C7XX fits perfectly in applications that require extremely fast execution of complex software programs. These include applications ranging from precise motor control and industrial process control to automotive, instrumentation, and telecom applications.

The EPROM technology makes customization of application programs (with unique security codes, combinations, model numbers, parameter storage, etc.) fast and convenient. Small footprint package options (including die sales) make the PIC17C7XX ideal for applications with space limitations that require high performance.

High speed execution, powerful peripheral features, flexible I/O, and low power consumption all at low cost make the PIC17C7XX ideal for a wide range of embedded control applications.

1.1 Family and Upward Compatibility

The PIC17CXXX family of microcontrollers have architectural enhancements over the PIC16C5X and PIC16CXX families. These enhancements allow the device to be more efficient in software and hardware requirements. Refer to Appendix A for a detailed list of enhancements and modifications. Code written for PIC16C5X or PIC16CXX can be easily ported to PIC17CXXX devices (Appendix B).

1.2 Development Support

The PIC17CXXX family is supported by a full featured macro assembler, a software simulator, an in-circuit emulator, a universal programmer, a “C” compiler and fuzzy logic support tools. For additional information, see Section 19.0.

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

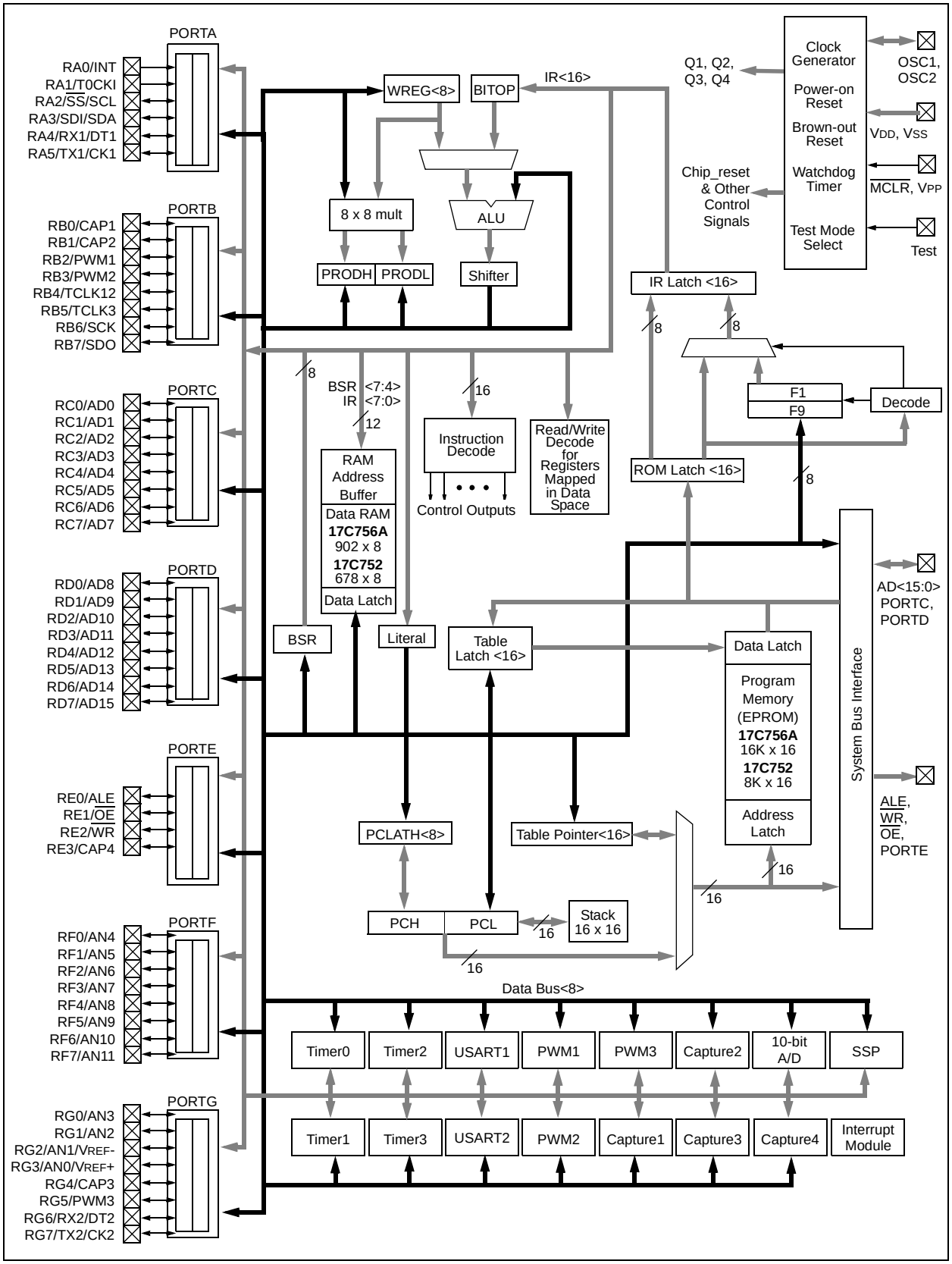
Microchip offers masked ROM versions of several of the highest volume parts, thus giving customers a low cost option for high volume, mature products.

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.

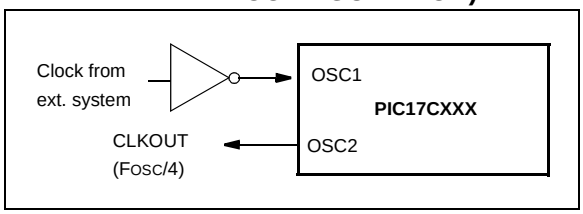
FIGURE 3-1: PIC17C752/756A BLOCK DIAGRAM



4.1.4 EXTERNAL CLOCK OSCILLATOR

In the EC oscillator mode, the OSC1 input can be driven by CMOS drivers. In this mode, the OSC1/CLKIN pin is hi-impedance and the OSC2/CLKOUT pin is the CLKOUT output (4 TOSC).

FIGURE 4-4: EXTERNAL CLOCK INPUT OPERATION (EC OSC CONFIGURATION)



4.1.5 EXTERNAL CRYSTAL OSCILLATOR CIRCUIT

Either a prepackaged oscillator can be used, or a simple oscillator circuit with TTL gates can be built. Prepackaged oscillators provide a wide operating range and better stability. A well designed crystal oscillator will provide good performance with TTL gates. Two types of crystal oscillator circuits can be used: one with series resonance, or one with parallel resonance.

Figure 4-5 shows implementation of a parallel resonant oscillator circuit. The circuit is designed to use the fundamental frequency of the crystal. The 74AS04 inverter performs the 180-degree phase shift that a parallel oscillator requires. The 4.7 kΩ resistor provides the negative feedback for stability. The 10 kΩ potentiometer biases the 74AS04 in the linear region. This could be used for external oscillator designs.

FIGURE 4-5: EXTERNAL PARALLEL RESONANT CRYSTAL OSCILLATOR CIRCUIT

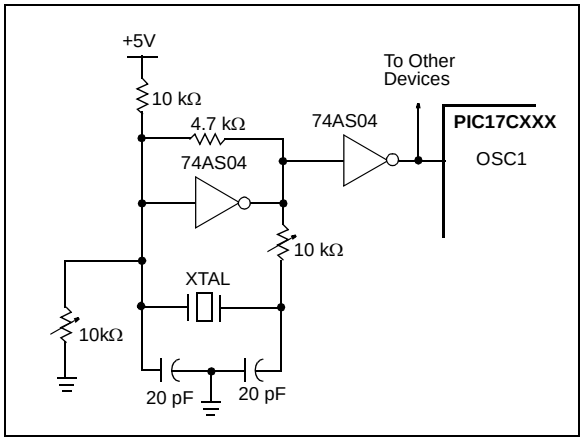
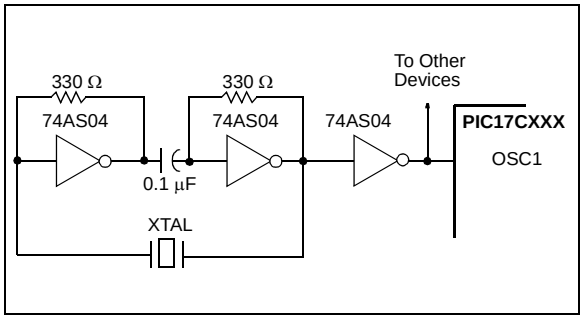


Figure 4-6 shows a series resonant oscillator circuit. This circuit is also designed to use the fundamental frequency of the crystal. The inverter performs a 180-degree phase shift in a series resonant oscillator circuit. The 330 Ω resistors provide the negative feedback to bias the inverters in their linear region.

FIGURE 4-6: EXTERNAL SERIES RESONANT CRYSTAL OSCILLATOR CIRCUIT



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

12.0 TIMER0

The Timer0 module consists of a 16-bit timer/counter, TMR0. The high byte is register TMR0H and the low byte is register TMR0L. A software programmable 8-bit prescaler makes Timer0 an effective 24-bit overflow timer. The clock source is software programmable as either the internal instruction clock, or an external clock on the RA1/T0CKI pin. The control bits for this module are in register T0STA (Figure 12-1).

REGISTER 12-1: T0STA REGISTER (ADDRESS: 05h, UNBANKED)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
INTEDG	T0SE	T0CS	T0PS3	T0PS2	T0PS1	T0PS0	—
bit 7							bit 0

bit 7 **INTEDG**: RA0/INT Pin Interrupt Edge Select bit
This bit selects the edge upon which the interrupt is detected.
1 = Rising edge of RA0/INT pin generates interrupt
0 = Falling edge of RA0/INT pin generates interrupt

bit 6 **T0SE**: Timer0 Clock Input Edge Select bit
This bit selects the edge upon which TMR0 will increment.

When T0CS = 0 (External Clock):
1 = Rising edge of RA1/T0CKI pin increments TMR0 and/or sets the T0CKIF bit
0 = Falling edge of RA1/T0CKI pin increments TMR0 and/or sets the T0CKIF bit

When T0CS = 1 (Internal Clock):
Don't care

bit 5 **T0CS**: Timer0 Clock Source Select bit
This bit selects the clock source for TMR0.
1 = Internal instruction clock cycle (T_{CY})
0 = External clock input on the T0CKI pin

bit 4-1 **T0PS3:T0PS0**: Timer0 Prescale Selection bits
These bits select the prescale value for TMR0.

T0PS3:T0PS0	Prescale Value
0000	1:1
0001	1:2
0010	1:4
0011	1:8
0100	1:16
0101	1:32
0110	1:64
0111	1:128
1xxx	1:256

bit 0 **Unimplemented**: Read as '0'

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

12.1 Timer0 Operation

When the T0CS (T0STA<5>) bit is set, TMR0 increments on the internal clock. When T0CS is clear, TMR0 increments on the external clock (RA1/T0CKI pin). The external clock edge can be selected in software. When the T0SE (T0STA<6>) bit is set, the timer will increment on the rising edge of the RA1/T0CKI pin. When T0SE is clear, the timer will increment on the falling edge of the RA1/T0CKI pin. The prescaler can be programmed to introduce a prescale of 1:1 to 1:256. The timer increments from 0000h to FFFFh and rolls over to 0000h. On overflow, the TMR0 Interrupt Flag bit (T0IF) is set. The TMR0 interrupt can be masked by clearing the corresponding TMR0 Interrupt Enable bit (T0IE). The TMR0 Interrupt Flag bit (T0IF) is automatically cleared when vectoring to the TMR0 interrupt vector.

12.2 Using Timer0 with External Clock

When an external clock input is used for Timer0, it is synchronized with the internal phase clocks. Figure 12-2 shows the synchronization of the external clock. This synchronization is done after the prescaler. The output of the prescaler (PSOUT) is sampled twice in every instruction cycle to detect a rising or a falling edge. The timing requirements for the external clock are detailed in the electrical specification section.

12.2.1 DELAY FROM EXTERNAL CLOCK EDGE

Since the prescaler output is synchronized with the internal clocks, there is a small delay from the time the external clock edge occurs to the time TMR0 is actually incremented. Figure 12-2 shows that this delay is between 3TOSC and 7TOSC. Thus, for example, measuring the interval between two edges (e.g. period) will be accurate within ±4TOSC (±121 ns @ 33 MHz).

FIGURE 12-1: TIMER0 MODULE BLOCK DIAGRAM

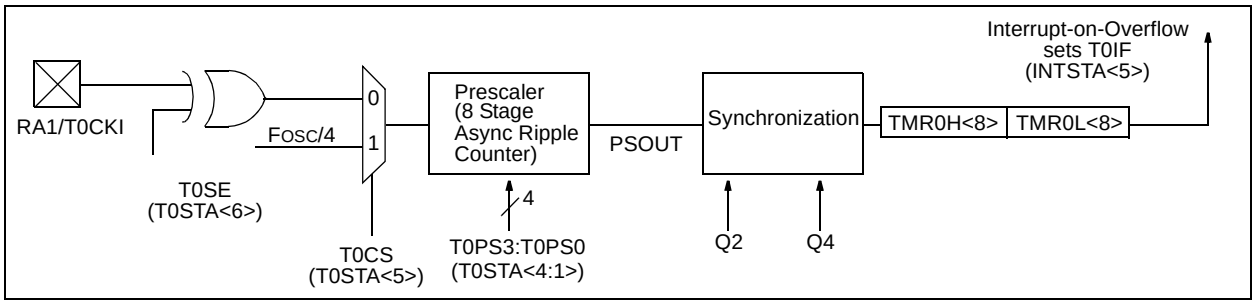
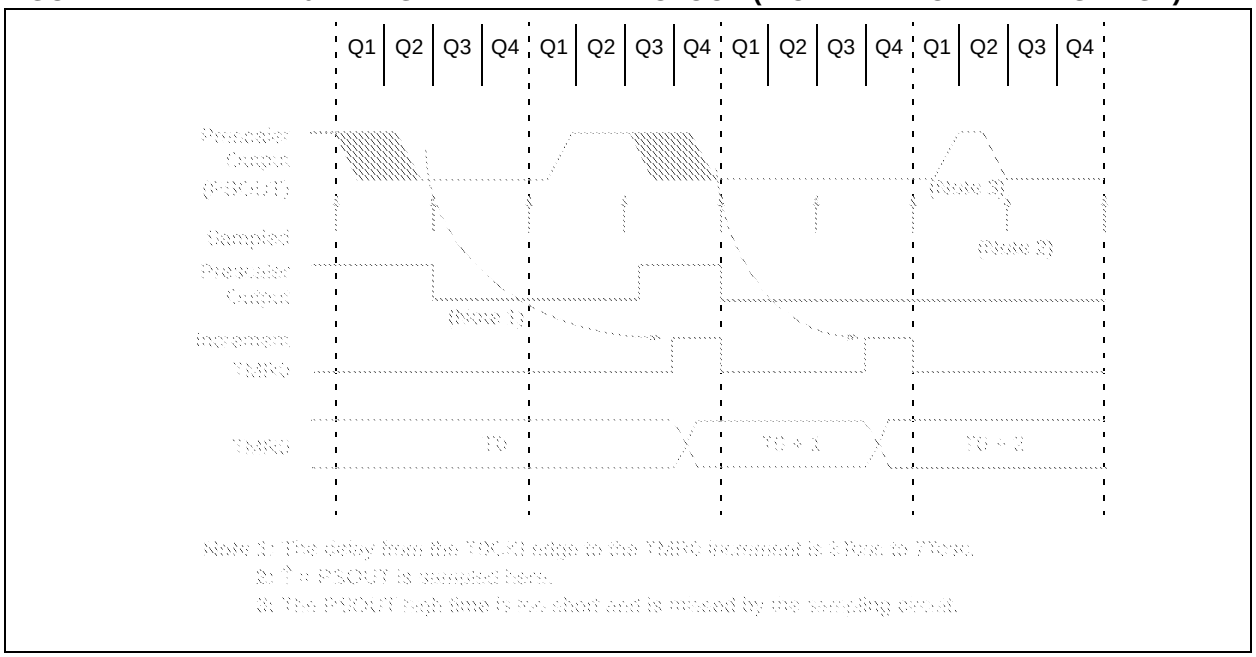


FIGURE 12-2: TMR0 TIMING WITH EXTERNAL CLOCK (INCREMENT ON FALLING EDGE)



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 14-5: BAUD RATES FOR ASYNCHRONOUS MODE

BAUD RATE (K)	FOSC = 33 MHz			FOSC = 25 MHz			FOSC = 20 MHz			FOSC = 16 MHz		
	KBAUD	%ERROR	SPBRG VALUE (DECIMAL)	KBAUD	%ERROR	SPBRG VALUE (DECIMAL)	KBAUD	%ERROR	SPBRG VALUE (DECIMAL)	KBAUD	%ERROR	SPBRG VALUE (DECIMAL)
0.3	NA	—	—	NA	—	—	NA	—	—	NA	—	—
1.2	NA	—	—	NA	—	—	1.221	+1.73	255	1.202	+0.16	207
2.4	2.398	-0.07	214	2.396	0.14	162	2.404	+0.16	129	2.404	+0.16	103
9.6	9.548	-0.54	53	9.53	-0.76	40	9.469	-1.36	32	9.615	+0.16	25
19.2	19.09	-0.54	26	19.53	+1.73	19	19.53	+1.73	15	19.23	+0.16	12
76.8	73.66	-4.09	6	78.13	+1.73	4	78.13	+1.73	3	83.33	+8.51	2
96	103.12	+7.42	4	97.65	+1.73	3	104.2	+8.51	2	NA	—	—
300	257.81	-14.06	1	390.63	+30.21	0	312.5	+4.17	0	NA	—	—
500	515.62	+3.13	0	NA	—	—	NA	—	—	NA	—	—
HIGH	515.62	—	0	—	—	0	312.5	—	0	250	—	0
LOW	2.014	—	255	1.53	—	255	1.221	—	255	0.977	—	255

BAUD RATE (K)	FOSC = 10 MHz			FOSC = 7.159 MHz			FOSC = 5.068 MHz		
	KBAUD	%ERROR	SPBRG VALUE (DECIMAL)	KBAUD	%ERROR	SPBRG VALUE (DECIMAL)	KBAUD	%ERROR	SPBRG VALUE (DECIMAL)
0.3	NA	—	—	NA	—	—	0.31	+3.13	255
1.2	1.202	+0.16	129	1.203	_0.23	92	1.2	0	65
2.4	2.404	+0.16	64	2.380	-0.83	46	2.4	0	32
9.6	9.766	+1.73	15	9.322	-2.90	11	9.9	-3.13	7
19.2	19.53	+1.73	7	18.64	-2.90	5	19.8	+3.13	3
76.8	78.13	+1.73	1	NA	—	—	79.2	+3.13	0
96	NA	—	—	NA	—	—	NA	—	—
300	NA	—	—	NA	—	—	NA	—	—
500	NA	—	—	NA	—	—	NA	—	—
HIGH	156.3	—	0	111.9	—	0	79.2	—	0
LOW	0.610	—	255	0.437	—	255	0.309	—	255

BAUD RATE (K)	FOSC = 3.579 MHz			FOSC = 1 MHz			FOSC = 32.768 kHz		
	KBAUD	%ERROR	SPBRG VALUE (DECIMAL)	KBAUD	%ERROR	SPBRG VALUE (DECIMAL)	KBAUD	%ERROR	SPBRG VALUE (DECIMAL)
0.3	0.301	+0.23	185	0.300	+0.16	51	0.256	-14.67	1
1.2	1.190	-0.83	46	1.202	+0.16	12	NA	—	—
2.4	2.432	+1.32	22	2.232	-6.99	6	NA	—	—
9.6	9.322	-2.90	5	NA	—	—	NA	—	—
19.2	18.64	-2.90	2	NA	—	—	NA	—	—
76.8	NA	—	—	NA	—	—	NA	—	—
96	NA	—	—	NA	—	—	NA	—	—
300	NA	—	—	NA	—	—	NA	—	—
500	NA	—	—	NA	—	—	NA	—	—
HIGH	55.93	—	0	15.63	—	0	0.512	—	0
LOW	0.218	—	255	0.061	—	255	0.002	—	255

14.3.2 USART SYNCHRONOUS MASTER RECEPTION

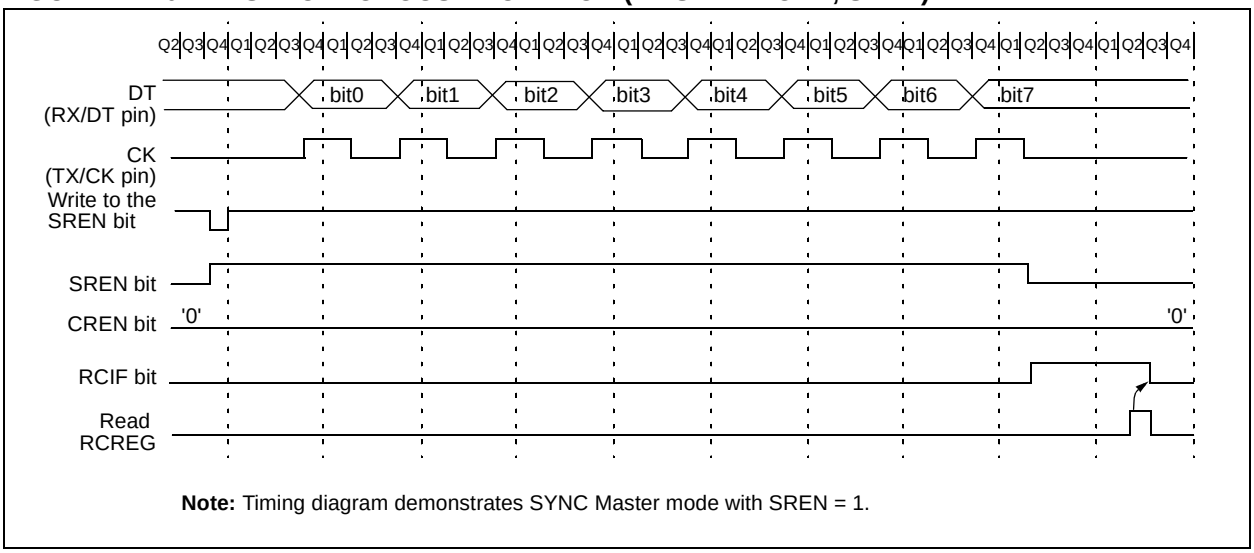
Once Synchronous mode is selected, reception is enabled by setting either the SREN (RCSTA<5>) bit or the CREN (RCSTA<4>) bit. Data is sampled on the RX/DT pin on the falling edge of the clock. If SREN is set, then only a single word is received. If CREN is set, the reception is continuous until CREN 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.

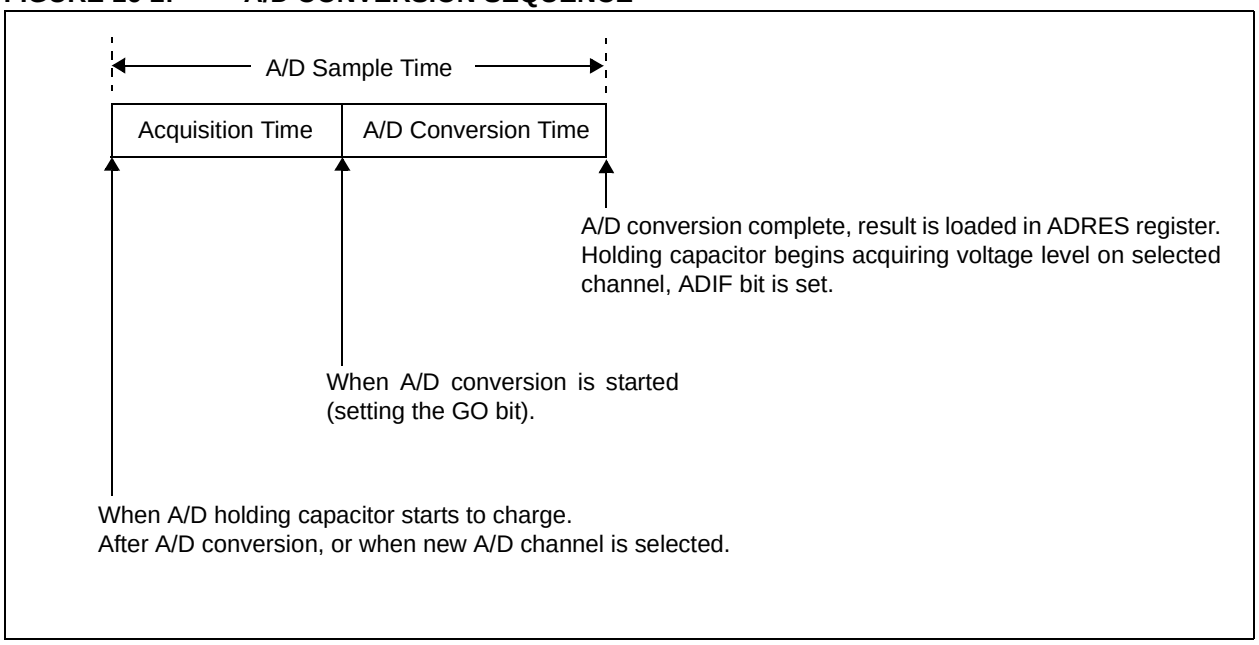
FIGURE 14-10: SYNCHRONOUS RECEPTION (MASTER MODE, SREN)



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

Figure 16-2 shows the conversion sequence and the terms that are used. Acquisition time is the time that the A/D module's holding capacitor is connected to the external voltage level. Then, there is the conversion time of 12 TAD, which is started when the GO bit is set. The sum of these two times is the sampling time. There is a minimum acquisition time to ensure that the holding capacitor is charged to a level that will give the desired accuracy for the A/D conversion.

FIGURE 16-2: A/D CONVERSION SEQUENCE



16.10 References

A good reference for understanding A/D converter is the "Analog-Digital Conversion Handbook" third edition, published by Prentice Hall (ISBN 0-13-03-2848-0).

TABLE 16-3: REGISTERS/BITS ASSOCIATED WITH A/D

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR, BOR	$\overline{\text{MCLR}}$, WDT
06h, unbanked	CPUSTA	—	—	STAKAV	GLINTD	$\overline{\text{TO}}$	$\overline{\text{PD}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$	--11 1100	--11 qq11
07h, unbanked	INTSTA	PEIF	T0CKIF	T0IF	INTF	PEIE	T0CKIE	T0IE	INTE	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
10h, Bank 5	DDRF	Data Direction Register for PORTF								1111 1111	1111 1111
11h, Bank 5	PORTF	RF7/ AN11	RF6/ AN10	RF5/ AN9	RF4/ AN8	RF3/ AN7	RF2/ AN6	RF1/ AN5	RF0/ AN4	0000 0000	0000 0000
12h, Bank 5	DDRG	Data Direction register for PORTG								1111 1111	1111 1111
13h, Bank 5	PORTG	RG7/ TX2/CK2	RG6/ RX2/DT2	RG5/ PWM3	RG4/ CAP3	RG3/ AN0/V_{REF+}	RG2/ AN1/V_{REF-}	RG1/ AN2	RG0/ AN3	xxxx 0000	uuuu 0000
14h, Bank 5	ADCON0	CHS3	CHS2	CHS1	CHS0	—	GO/$\overline{\text{DONE}}$	—	ADON	0000 -0-0	0000 -0-0
15h, Bank 5	ADCON1	ADCS1	ADCS0	ADFM	—	PCFG3	PCFG2	PCFG1	PCFG0	000- 0000	000- 0000
16h, Bank 5	ADRESL	A/D Result Low Register								xxxx xxxx	uuuu uuuu
17h, Bank 5	ADRESH	A/D Result High Register								xxxx xxxx	uuuu uuuu

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used for A/D conversion.

Note: Other (non power-up) RESETS include: external RESET through $\overline{\text{MCLR}}$ and Watchdog Timer Reset.

17.4 Power-down Mode (SLEEP)

The Power-down mode is entered by executing a SLEEP instruction. This clears the Watchdog Timer and postscaler (if enabled). The $\overline{PD}$ bit is cleared and the $\overline{TO}$ bit is set (in the CPUSTA register). In SLEEP mode, the oscillator driver is turned off. The I/O ports maintain their status (driving high,low, or hi-impedance input).

The $\overline{MCLR}/V_{PP}$ pin must be at a logic high level (V_{IHMC}). A WDT time-out RESET does not drive the $\overline{MCLR}/V_{PP}$ pin low.

17.4.1 WAKE-UP FROM SLEEP

The device can wake-up from SLEEP through one of the following events:

- Power-on Reset
- Brown-out Reset
- External RESET input on $\overline{MCLR}/V_{PP}$ pin
- WDT Reset (if WDT was enabled)
- Interrupt from RA0/INT pin, RB port change, T0CKI interrupt, or some peripheral interrupts

The following peripheral interrupts can wake the device from SLEEP:

- Capture interrupts
- USART synchronous slave transmit interrupts
- USART synchronous slave receive interrupts
- A/D conversion complete
- SPI slave transmit/receive complete
- I^2C slave receive

Other peripherals cannot generate interrupts since during SLEEP, no on-chip Q clocks are present.

Any RESET event will cause a device RESET. Any interrupt event is considered a continuation of program execution. The $\overline{TO}$ and $\overline{PD}$ bits in the CPUSTA register can be used to determine the cause of a device RESET. The $\overline{PD}$ bit, which is set on power-up, is cleared when SLEEP is invoked. The $\overline{TO}$ bit is cleared if WDT time-out occurred (and caused a RESET).

When the SLEEP instruction is being executed, the next instruction (PC + 1) is pre-fetched. For the device to wake-up through an interrupt event, the corresponding interrupt enable bit must be set (enabled). Wake-up is regardless of the state of the GLINTD bit. If the GLINTD bit is set (disabled), the device continues execution at the instruction after the SLEEP instruction. If the GLINTD bit is clear (enabled), the device executes the instruction after the SLEEP instruction and then branches to the interrupt vector address. In cases where the execution of the instruction following SLEEP is not desirable, the user should have a NOP after the SLEEP instruction.

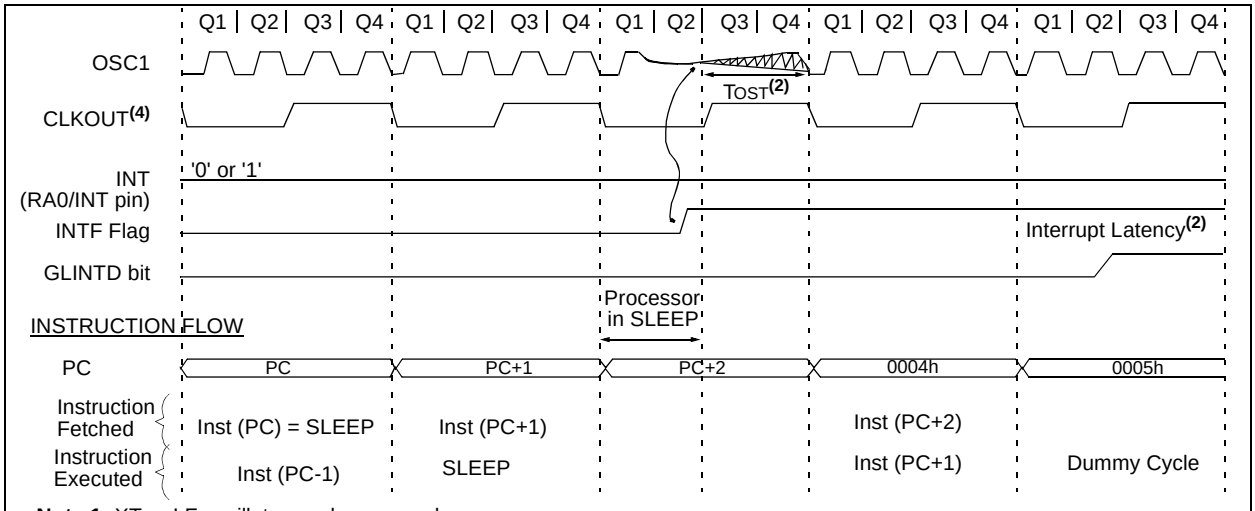
Note:	If the global interrupt is disabled (GLINTD is set), but any interrupt source has both its interrupt enable bit and the corresponding interrupt flag bit set, the device will immediately wake-up from SLEEP. The $\overline{TO}$ bit is set and the $\overline{PD}$ bit is cleared.

The WDT is cleared when the device wakes from SLEEP, regardless of the source of wake-up.

17.4.1.1 Wake-up Delay

When the oscillator type is configured in XT or LF mode, the Oscillator Start-up Timer (OST) is activated on wake-up. The OST will keep the device in RESET for 1024TOSC. This needs to be taken into account when considering the interrupt response time when coming out of SLEEP.

FIGURE 17-2: WAKE-UP FROM SLEEP THROUGH INTERRUPT



Note 1: XT or LF oscillator mode assumed.
2: TOST = 1024TOSC (drawing not to scale). This delay will not be there for RC osc mode.
3: When GLINTD = 0, processor jumps to interrupt routine after wake-up. If GLINTD = 1, execution will continue in line.
4: CLKOUT is not available in these osc modes, but shown here for timing reference.

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

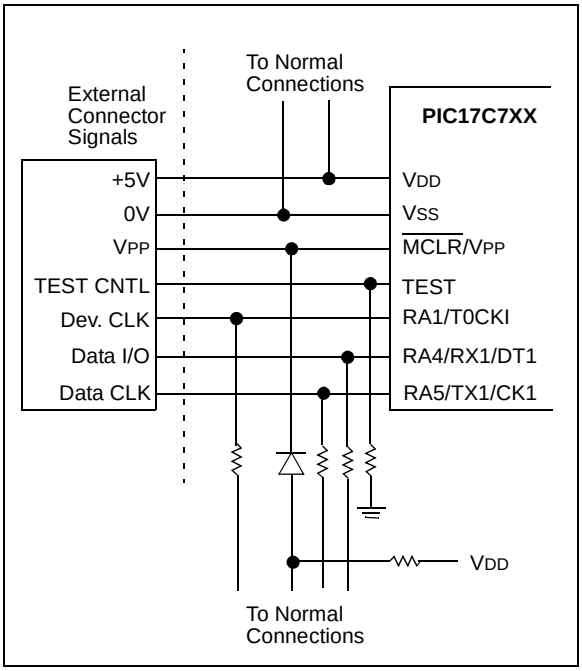


TABLE 17-3: ICSP INTERFACE PINS

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

MOVFP **Move f to p**

Syntax: [*label*] MOVFP f,p

Operands: $0 \le f \le 255$
$0 \le p \le 31$

Operation: (f) → (p)

Status Affected: None

Encoding:

011p	pppp	ffff	ffff

Description: Move data from data memory location 'f' to data memory location 'p'. Location 'f' can be anywhere in the 256 byte data space (00h to FFh), while 'p' can be 00h to 1Fh.

Either 'p' or 'f' can be WREG (a useful, special situation).

MOVFP is particularly useful for transferring a data memory location to a peripheral register (such as the transmit buffer or an I/O port). Both 'f' and 'p' can be indirectly addressed.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example: MOVFP REG1, REG2

Before Instruction
REG1 = 0x33,
REG2 = 0x11

After Instruction
REG1 = 0x33,
REG2 = 0x33

MOVLB **Move Literal to low nibble in BSR**

Syntax: [*label*] MOVLB k

Operands: $0 \le k \le 15$

Operation: k → (BSR<3:0>)

Status Affected: None

Encoding:

1011	1000	uuuu	kkkk

Description: The four-bit literal 'k' is loaded in the Bank Select Register (BSR). Only the low 4-bits of the Bank Select Register are affected. The upper 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<3:0>

Example: MOVLB 5

Before Instruction
BSR register = 0x22

After Instruction
BSR register = 0x25 (Bank 5)

MOVPF **Move p to f**

Syntax: [*label*] MOVPF p,f

Operands: $0 \le f \le 255$
$0 \le p \le 31$

Operation: (p) → (f)

Status Affected: Z

Encoding:

010p	pppp	ffff	ffff

Description: Move data from data memory location 'p' to data memory location 'f'. Location 'f' can be anywhere in the 256 byte data space (00h to FFh), while 'p' can be 00h to 1Fh.

Either 'p' or 'f' can be WREG (a useful, special situation).

MOVPF is particularly useful for transferring a peripheral register (e.g. the timer or an I/O port) to a data memory location. Both 'f' and 'p' can be indirectly addressed.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example:
```
MOVPF   REG1, REG2
```
Before Instruction
REG1 = 0x11
REG2 = 0x33

After Instruction
REG1 = 0x11
REG2 = 0x11

MOVWF **Move WREG to f**

Syntax: [*label*] MOVWF f

Operands: $0 \le f \le 255$

Operation: (WREG) → (f)

Status Affected: None

Encoding:

0000	0001	ffff	ffff

Description: Move data from WREG to register 'f'. Location 'f' can be anywhere in the 256 byte data space.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example:
```
MOVWF   REG
```
Before Instruction
WREG = 0x4F
REG = 0xFF

After Instruction
WREG = 0x4F
REG = 0x4F

Param No.	Sym	Characteristic		Min	Max	Units	Conditions
110	Tbuf	Bus free time	100 kHz mode	4.7	—	ms	Time the bus must be free before a new transmission can start
			400 kHz mode	1.3	—	ms	
			1 MHz mode(1)	0.5	—	ms	
D102	Cb	Bus capacitive loading		—	400	pF	

Note 1: Maximum pin capacitance = 10 pF for all I^2C pins.

2: A fast mode (400 KHz) I^2C bus device can be used in a standard mode I^2C 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 (V_{DD}=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 20-19: USART SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING

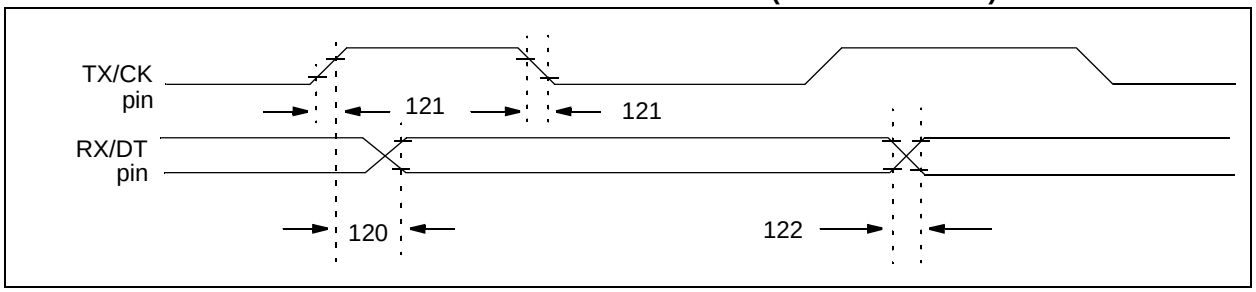


TABLE 20-14: USART SYNCHRONOUS TRANSMISSION REQUIREMENTS

Param No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
120	TckH2dtV	SYNC XMIT (MASTER & SLAVE) Clock high to data out valid	PIC17**C**XXX	—	—	50	ns	
			PIC17**LC**XXX	—	—	75	ns	
121	TckRF	Clock out rise time and fall time (Master mode)	PIC17**C**XXX	—	—	25	ns	
			PIC17**LC**XXX	—	—	40	ns	
122	TdtRF	Data out rise time and fall time	PIC17**C**XXX	—	—	25	ns	
			PIC17**LC**XXX	—	—	40	ns	

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

FIGURE 21-19: TYPICAL, MINIMUM AND MAXIMUM VOH vs. IOH (VDD = 3V, -40°C TO +125°C)

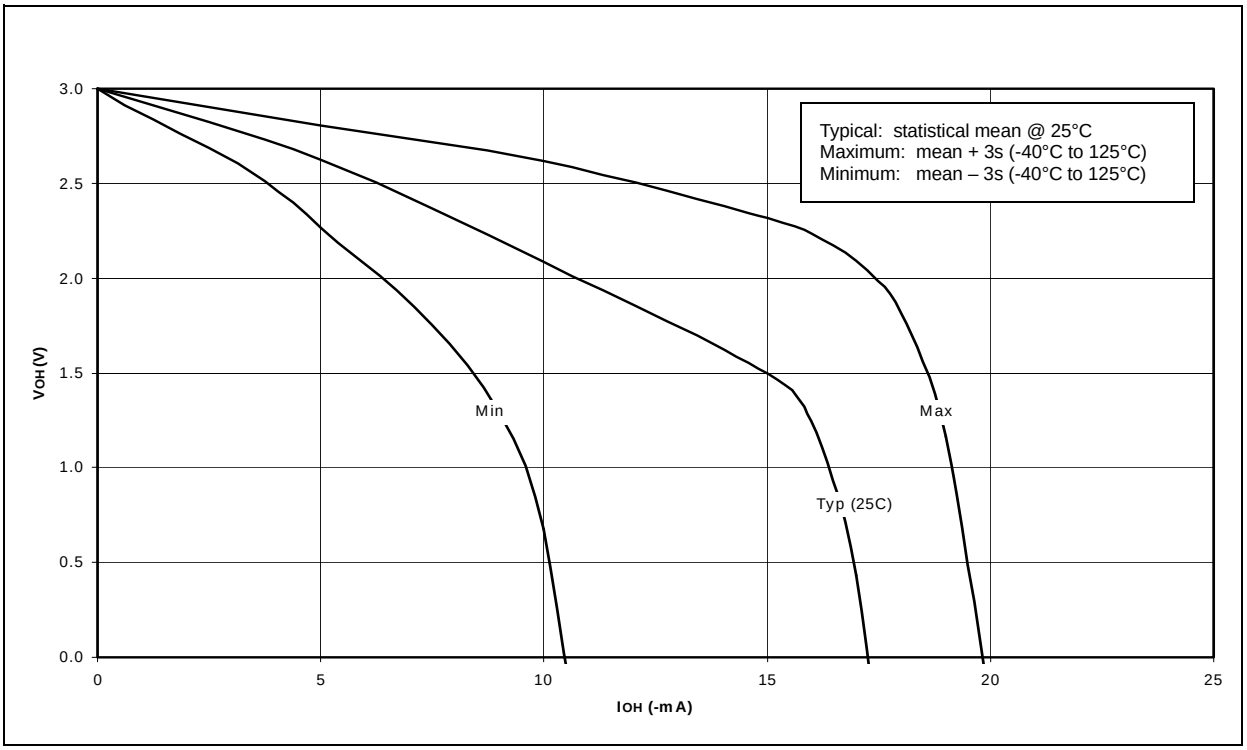
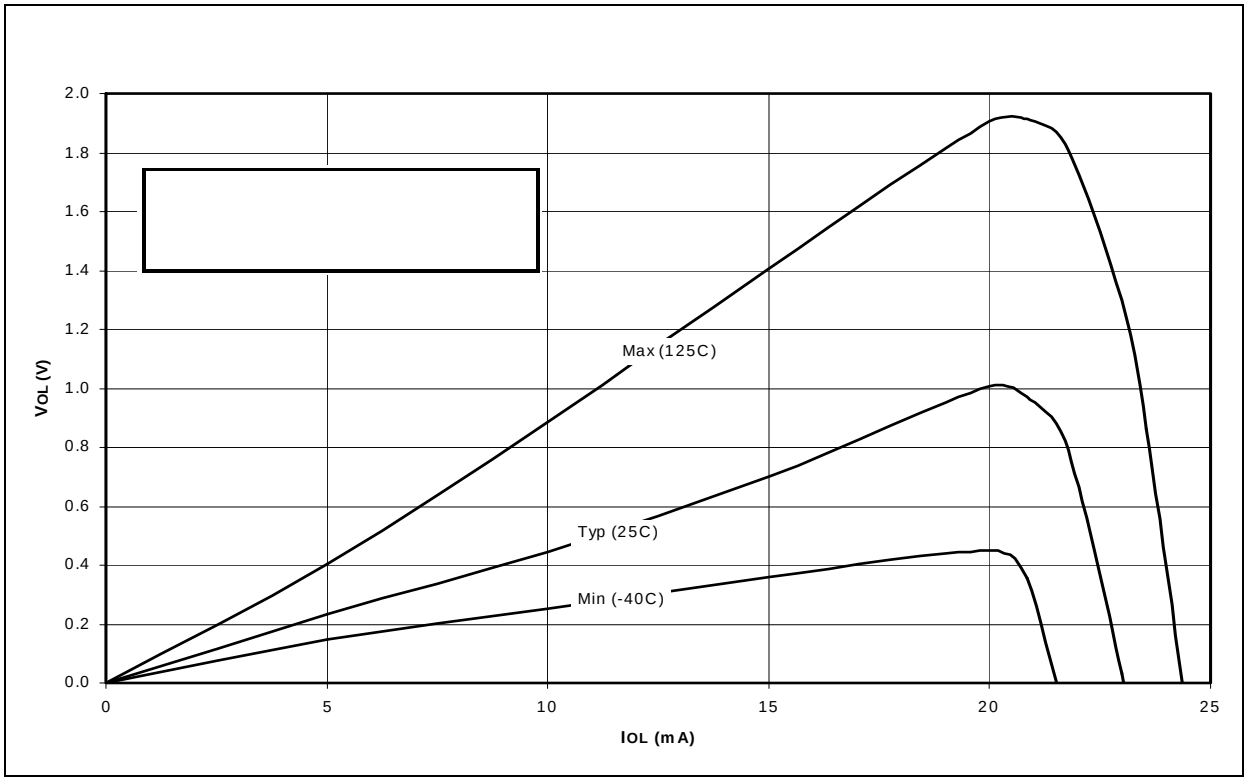
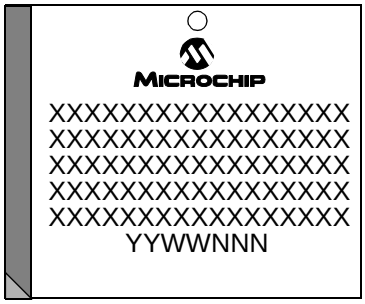


FIGURE 21-20: TYPICAL, MINIMUM AND MAXIMUM VOL vs. IOL (VDD = 3V, -40°C TO +125°C)

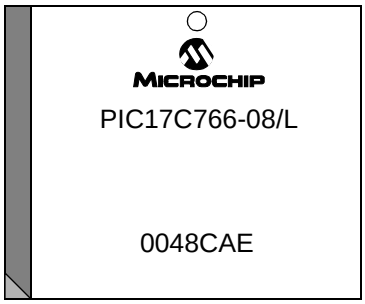


Package Marking Information (Cont.)

84-Lead PLCC



Example



NOTES: