

Welcome to [E-XFL.COM](https://www.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	I ² C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	50
Program Memory Size	16KB (8K x 16)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	454 x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 12x10b
Oscillator Type	External
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic17c752t-33-pt

4.2 Clocking Scheme/Instruction Cycle

The clock input (from OSC1) is internally divided by four to generate four non-overlapping quadrature clocks, namely Q1, Q2, Q3 and Q4. Internally, the program counter (PC) is incremented every Q1 and the instruction is fetched from the program memory and latched into the instruction register in Q4. The instruction is decoded and executed during the following Q1 through Q4. The clocks and instruction execution flow are shown in Figure 4-8.

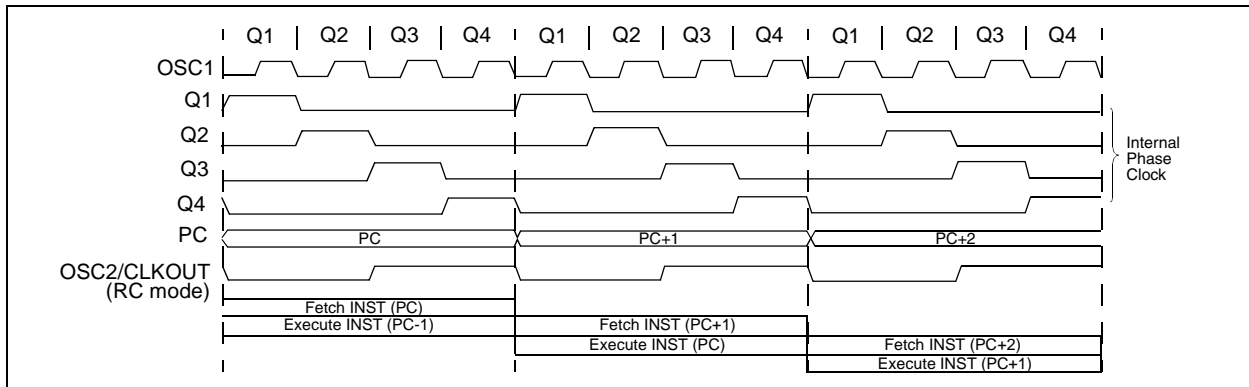
4.3 Instruction Flow/Pipelining

An "Instruction Cycle" consists of four Q cycles (Q1, Q2, Q3 and Q4). The instruction fetch and execute are pipelined such that fetch takes one instruction cycle, while decode and execute takes another instruction cycle. However, due to the pipelining, each instruction effectively executes in one cycle. If an instruction causes the program counter to change (e.g. GOTO), then two cycles are required to complete the instruction (Example 4-1).

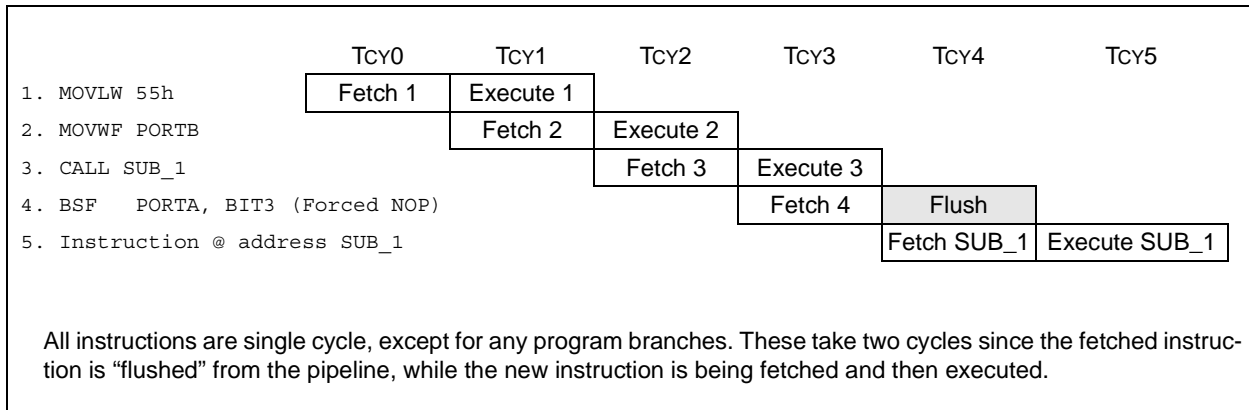
A fetch cycle begins with the program counter incrementing in Q1.

In the execution cycle, the fetched instruction is latched into the "Instruction Register (IR)" in cycle Q1. This instruction is then decoded and executed during the Q2, Q3 and Q4 cycles. Data memory is read during Q2 (operand read) and written during Q4 (destination write).

FIGURE 4-8: CLOCK/INSTRUCTION CYCLE



EXAMPLE 4-1: INSTRUCTION PIPELINE FLOW



PIC17C7XX

NOTES:

PIC17C7XX

6.1 Interrupt Status Register (INTSTA)

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

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

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

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

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

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

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

- bit 7 **PEIF:** Peripheral Interrupt Flag bit
This bit is the OR of all peripheral interrupt flag bits AND'ed with their corresponding enable bits. The interrupt logic forces program execution to address (20h) when a peripheral interrupt is pending.
1 = A peripheral interrupt is pending
0 = No peripheral interrupt is pending
- bit 6 **T0CKIF:** External Interrupt on T0CKI Pin Flag bit
This bit is cleared by hardware, when the interrupt logic forces program execution to address (18h).
1 = The software specified edge occurred on the RA1/T0CKI pin
0 = The software specified edge did not occur on the RA1/T0CKI pin
- bit 5 **T0IF:** TMR0 Overflow Interrupt Flag bit
This bit is cleared by hardware, when the interrupt logic forces program execution to address (10h).
1 = TMR0 overflowed
0 = TMR0 did not overflow
- bit 4 **INTF:** External Interrupt on INT Pin Flag bit
This bit is cleared by hardware, when the interrupt logic forces program execution to address (08h).
1 = The software specified edge occurred on the RA0/INT pin
0 = The software specified edge did not occur on the RA0/INT pin
- bit 3 **PEIE:** Peripheral Interrupt Enable bit
This bit acts as a global enable bit for the peripheral interrupts that have their corresponding enable bits set.
1 = Enable peripheral interrupts
0 = Disable peripheral interrupts
- bit 2 **T0CKIE:** External Interrupt on T0CKI Pin Enable bit
1 = Enable software specified edge interrupt on the RA1/T0CKI pin
0 = Disable interrupt on the RA1/T0CKI pin
- bit 1 **T0IE:** TMR0 Overflow Interrupt Enable bit
1 = Enable TMR0 overflow interrupt
0 = Disable TMR0 overflow interrupt
- bit 0 **INTE:** External Interrupt on RA0/INT Pin Enable bit
1 = Enable software specified edge interrupt on the RA0/INT pin
0 = Disable software specified edge interrupt on the RA0/INT pin

Legend:

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

PIC17C7XX

REGISTER 6-3: PIE2 REGISTER (ADDRESS: 11h, BANK 4)

R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
SSPIE	BCLIE	ADIE	—	CA4IE	CA3IE	TX2IE	RC2IE
bit 7				bit 0			

- bit 7 **SSPIE:** Synchronous Serial Port Interrupt Enable bit
1 = Enable SSP interrupt
0 = Disable SSP interrupt
- bit 6 **BCLIE:** Bus Collision Interrupt Enable bit
1 = Enable bus collision interrupt
0 = Disable bus collision interrupt
- bit 5 **ADIE:** A/D Module Interrupt Enable bit
1 = Enable A/D module interrupt
0 = Disable A/D module interrupt
- bit 4 **Unimplemented:** Read as '0'
- bit 3 **CA4IE:** Capture4 Interrupt Enable bit
1 = Enable Capture4 interrupt
0 = Disable Capture4 interrupt
- bit 2 **CA3IE:** Capture3 Interrupt Enable bit
1 = Enable Capture3 interrupt
0 = Disable Capture3 interrupt
- bit 1 **TX2IE:** USART2 Transmit Interrupt Enable bit
1 = Enable USART2 Transmit buffer empty interrupt
0 = Disable USART2 Transmit buffer empty interrupt
- bit 0 **RC2IE:** USART2 Receive Interrupt Enable bit
1 = Enable USART2 Receive buffer full interrupt
0 = Disable USART2 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

7.4.2 INDIRECT ADDRESSING OPERATION

The indirect addressing capability has been enhanced over that of the PIC16CXX family. There are two control bits associated with each FSR register. These two bits configure the FSR register to:

- Auto-decrement the value (address) in the FSR after an indirect access
- Auto-increment the value (address) in the FSR after an indirect access
- No change to the value (address) in the FSR after an indirect access

These control bits are located in the ALUSTA register. The FSR1 register is controlled by the FS3:FS2 bits and FSR0 is controlled by the FS1:FS0 bits.

When using the auto-increment or auto-decrement features, the effect on the FSR is not reflected in the ALUSTA register. For example, if the indirect address causes the FSR to equal '0', the Z bit will not be set.

If the FSR register contains a value of 0h, an indirect read will read 0h (Zero bit is set) while an indirect write will be equivalent to a NOP (status bits are not affected).

Indirect addressing allows single cycle data transfers within the entire data space. This is possible with the use of the MOVFP and MOVFP instructions, where either 'p' or 'f' is specified as INDF0 (or INDF1).

If the source or destination of the indirect address is in banked memory, the location accessed will be determined by the value in the BSR.

A simple program to clear RAM from 20h - FFh is shown in Example 7-1.

EXAMPLE 7-1: INDIRECT ADDRESSING

```

MOV LW 0x20      ;
MOVWF FSR0      ; FSR0 = 20h
BCF ALUSTA, FS1 ; Increment FSR
BSF ALUSTA, FS0 ; after access
BCF ALUSTA, C   ; C = 0
MOV LW END_RAM + 1 ;
LP CLRF INDF0, F ; Addr(FSR) = 0
CPFSEQ FSR0     ; FSR0 = END_RAM+1?
GOTO LP        ; NO, clear next
:              ; YES, All RAM is
:              ; cleared
    
```

7.5 Table Pointer (TBLPTRL and TBLPTRH)

File registers TBLPTRL and TBLPTRH form a 16-bit pointer to address the 64K program memory space. The table pointer is used by instructions TABLWT and TABLRD.

The TABLRD and the TABLWT instructions allow transfer of data between program and data space. The table pointer serves as the 16-bit address of the data word within the program memory. For a more complete description of these registers and the operation of Table Reads and Table Writes, see Section 8.0.

7.6 Table Latch (TBLATH, TBLATL)

The table latch (TBLAT) is a 16-bit register, with TBLATH and TBLATL referring to the high and low bytes of the register. It is not mapped into data or program memory. The table latch is used as a temporary holding latch during data transfer between program and data memory (see TABLRD, TABLWT, TLRD and TLWT instruction descriptions). For a more complete description of these registers and the operation of Table Reads and Table Writes, see Section 8.0.

10.0 I/O PORTS

PIC17C75X devices have seven I/O ports, PORTA through PORTG. PIC17C76X devices have nine I/O ports, PORTA through PORTJ. PORTB through PORTJ have a corresponding Data Direction Register (DDR), which is used to configure the port pins as inputs or outputs. Some of these ports pins are multiplexed with alternate functions.

PORTC, PORTD, and PORTE are multiplexed with the system bus. These pins are configured as the system bus when the device's configuration bits are selected to Microprocessor or Extended Microcontroller modes. In the two other microcontroller modes, these pins are general purpose I/O.

PORTA, PORTB, PORTE<3>, PORTF, PORTG and the upper four bits of PORTH are multiplexed with the peripheral features of the device. These peripheral features are:

- Timer Modules
- Capture Modules
- PWM Modules
- USART/SCI Modules
- SSP Module
- A/D Module
- External Interrupt pin

When some of these peripheral modules are turned on, the port pin will automatically configure to the alternate function. The modules that do this are:

- PWM Module
- SSP Module
- USART/SCI Module

When a pin is automatically configured as an output by a peripheral module, the pins data direction (DDR) bit is unknown. After disabling the peripheral module, the user should re-initialize the DDR bit to the desired configuration.

The other peripheral modules (which require an input) must have their data direction bits configured appropriately.

<p>Note: A pin that is a peripheral input, can be configured as an output (DDRx<y> is cleared). The peripheral events will be determined by the action output on the port pin.</p>

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

PIC17C7XX

TABLE 14-10: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE TRANSMISSION

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

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

TABLE 14-11: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE RECEPTION

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

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

15.2.12 I²C MASTER MODE RECEPTION

Master mode reception is enabled by programming the receive enable bit, RCEN (SSPCON2<3>).

Note: The SSP Module must be in an IDLE STATE before the RCEN bit is set, or the RCEN bit will be disregarded.

The baud rate generator begins counting and on each rollover, the state of the SCL pin changes (high to low/low to high) and data is shifted into the SSPSR. After the falling edge of the eighth clock, the receive enable flag is automatically cleared, the contents of the SSPSR are loaded into the SSPBUF, the BF flag is set, the SSPIF is set and the baud rate generator is suspended from counting, holding SCL low. The SSP is now in IDLE state, awaiting the next command. When the buffer is read by the CPU, the BF flag is automatically cleared. The user can then send an acknowledge bit at the end of reception, by setting the acknowledge sequence enable bit, ACKEN (SSPCON2<4>).

15.2.12.1 BF Status Flag

In receive operation, BF is set when an address or data byte is loaded into SSPBUF from SSPSR. It is cleared when SSPBUF is read.

15.2.12.2 SSPOV Status Flag

In receive operation, SSPOV is set when 8 bits are received into the SSPSR, and the BF flag is already set from a previous reception.

15.2.12.3 WCOL Status Flag

If the user writes the SSPBUF when a receive is already in progress (i.e., SSPSR is still shifting in a data byte), then WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).

PIC17C7XX

15.2.18 MULTI-MASTER COMMUNICATION, BUS COLLISION AND BUS ARBITRATION

Multi-Master mode support is achieved by bus arbitration. When the master outputs address/data bits onto the SDA pin, arbitration takes place when the master outputs a '1' on SDA, by letting SDA float high and another master asserts a '0'. When the SCL pin floats high, data should be stable. If the expected data on SDA is a '1' and the data sampled on the SDA pin = '0', then a bus collision has taken place. The master will set the Bus Collision Interrupt Flag, BCLIF and reset the I²C port to its IDLE state (Figure 15-34).

If a transmit was in progress when the bus collision occurred, the transmission is halted, the BF flag is cleared, the SDA and SCL lines are de-asserted and the SSPBUF can be written to. When the user services the bus collision Interrupt Service Routine and if the I²C bus is free, the user can resume communication by asserting a START condition.

If a START, Repeated Start, STOP, or Acknowledge condition was in progress when the bus collision occurred, the condition is aborted, the SDA and SCL lines are de-asserted and the respective control bits in the SSPCON2 register are cleared. When the user services the bus collision Interrupt Service Routine, and if the I²C bus is free, the user can resume communication by asserting a START condition.

The master will continue to monitor the SDA and SCL pins and if a STOP condition occurs, the SSPIF bit will be set.

A write to the SSPBUF will start the transmission of data at the first data bit, regardless of where the transmitter left off when bus collision occurred.

In Multi-Master mode, the interrupt generation on the detection of START and STOP conditions allows the determination of when the bus is free. Control of the I²C bus can be taken when the P bit is set in the SSPSTAT register, or the bus is idle and the S and P bits are cleared.

FIGURE 15-34: BUS COLLISION TIMING FOR TRANSMIT AND ACKNOWLEDGE

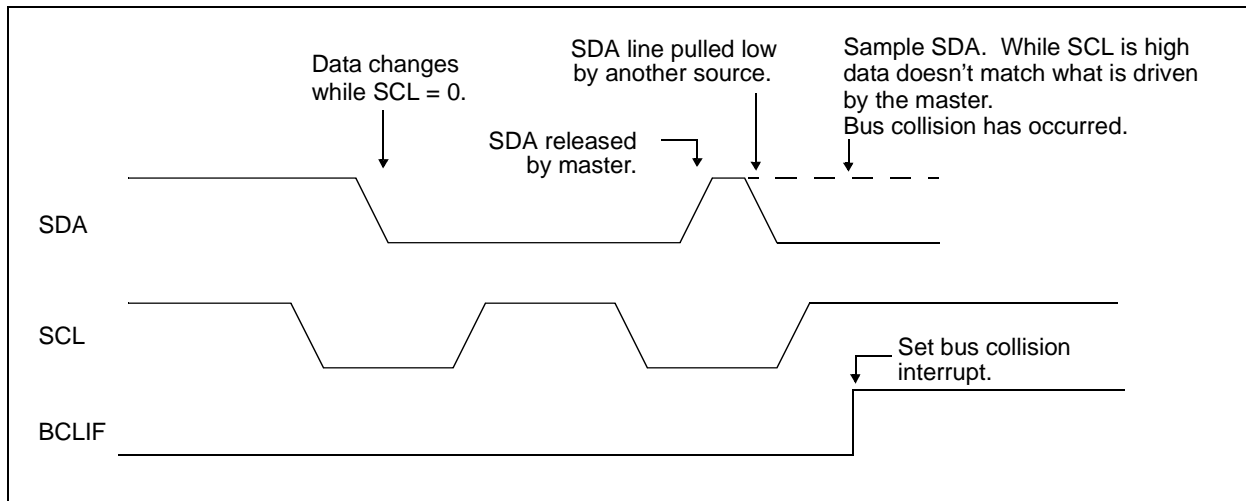
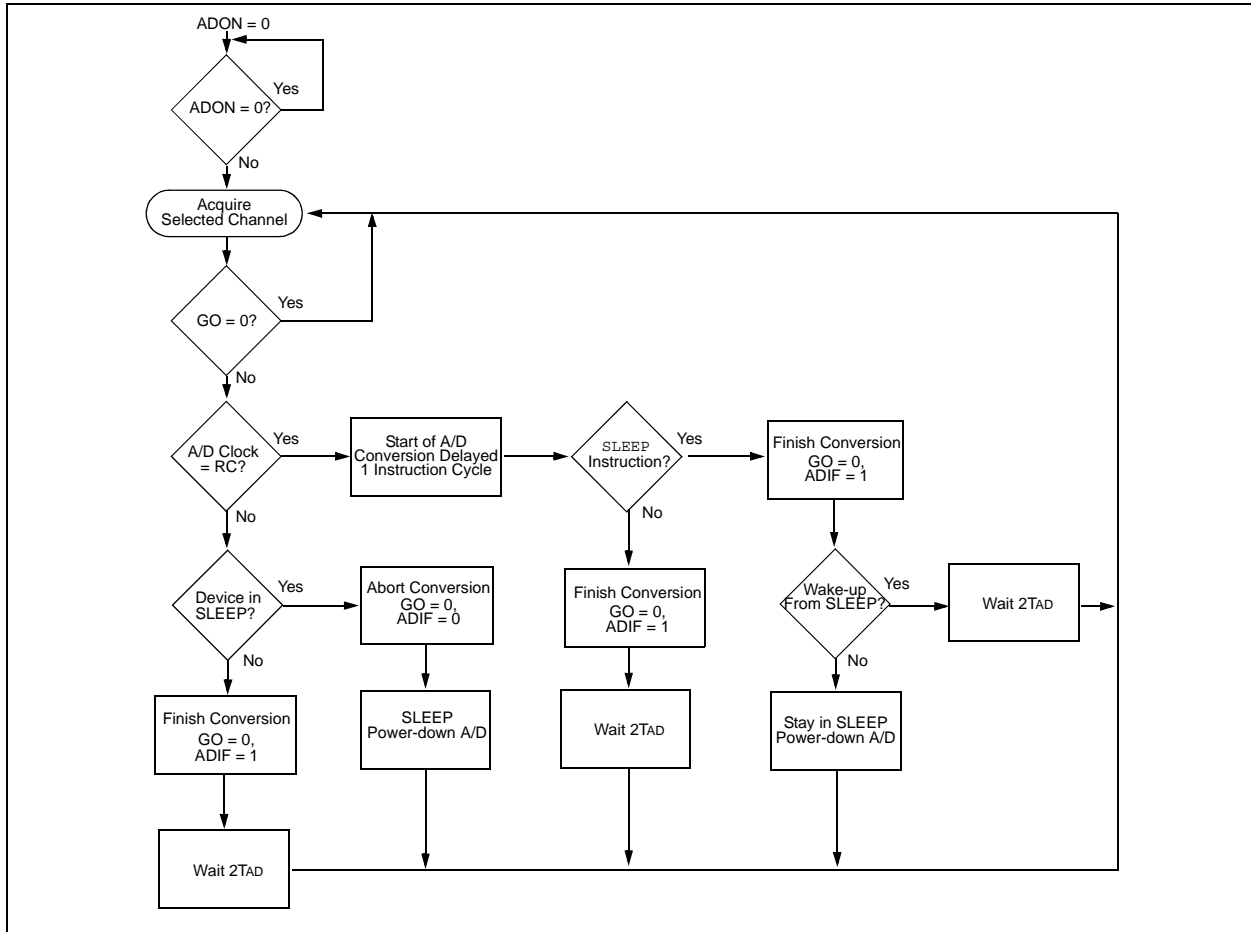


FIGURE 16-5: FLOW CHART OF A/D OPERATION



17.4 Power-down Mode (SLEEP)

The Power-down mode is entered by executing a `SLEEP` instruction. This clears the Watchdog Timer and postscale (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}/VPP pin must be at a logic high level (V_{IHMC}). A WDT time-out RESET does not drive the \overline{MCLR}/VPP 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}/VPP 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
- I²C 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 $1024T_{osc}$. 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

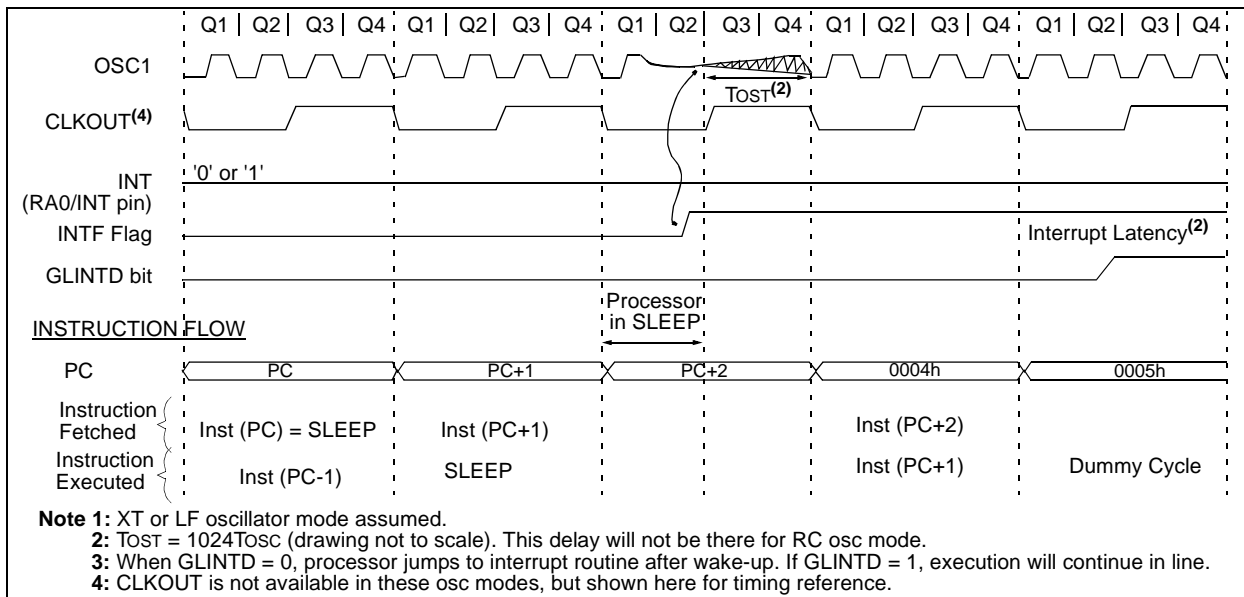


FIGURE 20-14: SPI MASTER MODE TIMING (CKE = 1)

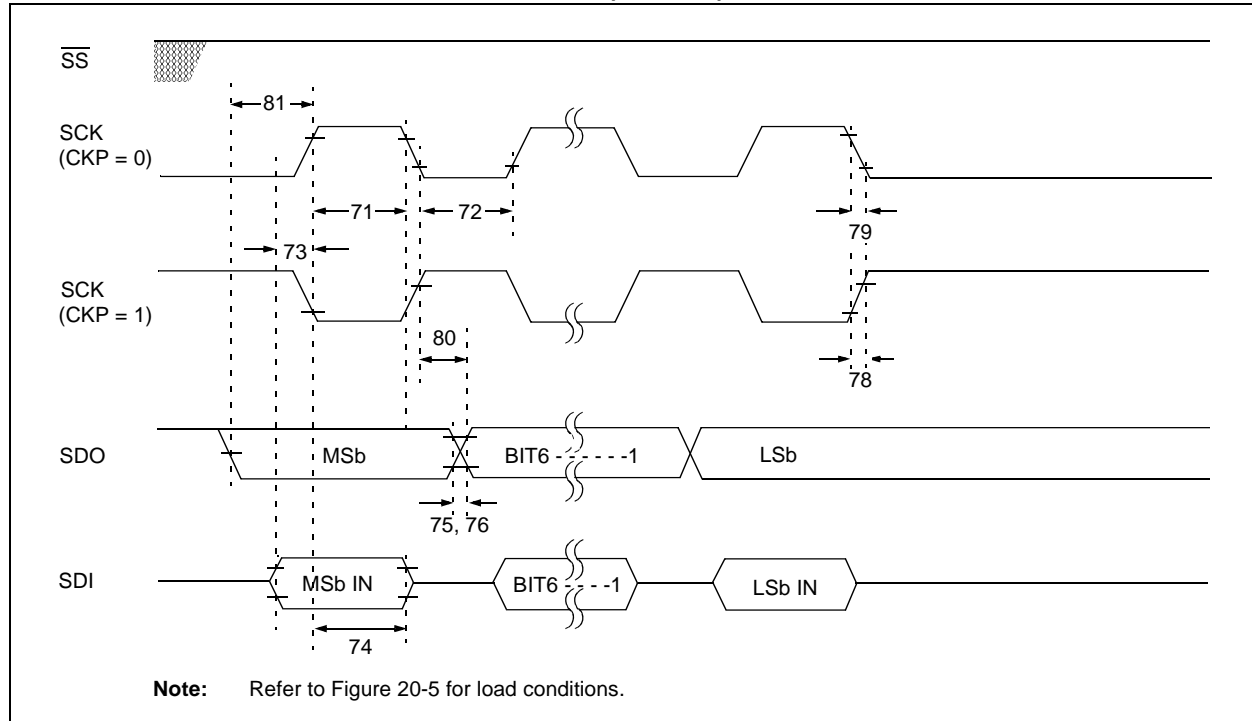


TABLE 20-9: SPI MODE REQUIREMENTS (MASTER MODE, CKE = 1)

Param. No.	Symbol	Characteristic	Min	Typ†	Max	Units	Conditions
71	TscH	SCK input high time	1.25Tcy + 30	—	—	ns	
71A		(Slave mode)					
		Continuous	1.25Tcy + 30	—	—	ns	(Note 1)
		Single Byte	40	—	—	ns	(Note 1)
72	TscL	SCK input low time	1.25 Tcy + 30	—	—	ns	
72A		(Slave mode)					
		Continuous	1.25 Tcy + 30	—	—	ns	(Note 1)
		Single Byte	40	—	—	ns	(Note 1)
73	TdiV2scH, TdiV2scL	Setup time of SDI data input to SCK edge	100	—	—	ns	
73A	Tb2B	Last clock edge of Byte1 to the 1st clock edge of Byte2	1.5Tcy + 40	—	—	ns	(Note 1)
74	Tsch2diL, TscL2diL	Hold time of SDI data input to SCK edge	100	—	—	ns	
75	TdoR	SDO data output rise time	—	10	25	ns	
76	TdoF	SDO data output fall time	—	10	25	ns	
78	TscR	SCK output rise time (Master mode)	—	10	25	ns	
79	TscF	SCK output fall time (Master mode)	—	10	25	ns	
80	Tsch2doV, TscL2doV	SDO data output valid after SCK edge	—	—	50	ns	
81	TdoV2scH, TdoV2scL	SDO data output setup to SCK edge	Tcy	—	—	ns	

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

Note 1: Specification 73A is only required if specifications 71A and 72A are used.

FIGURE 20-16: SPI SLAVE MODE TIMING (CKE = 1)

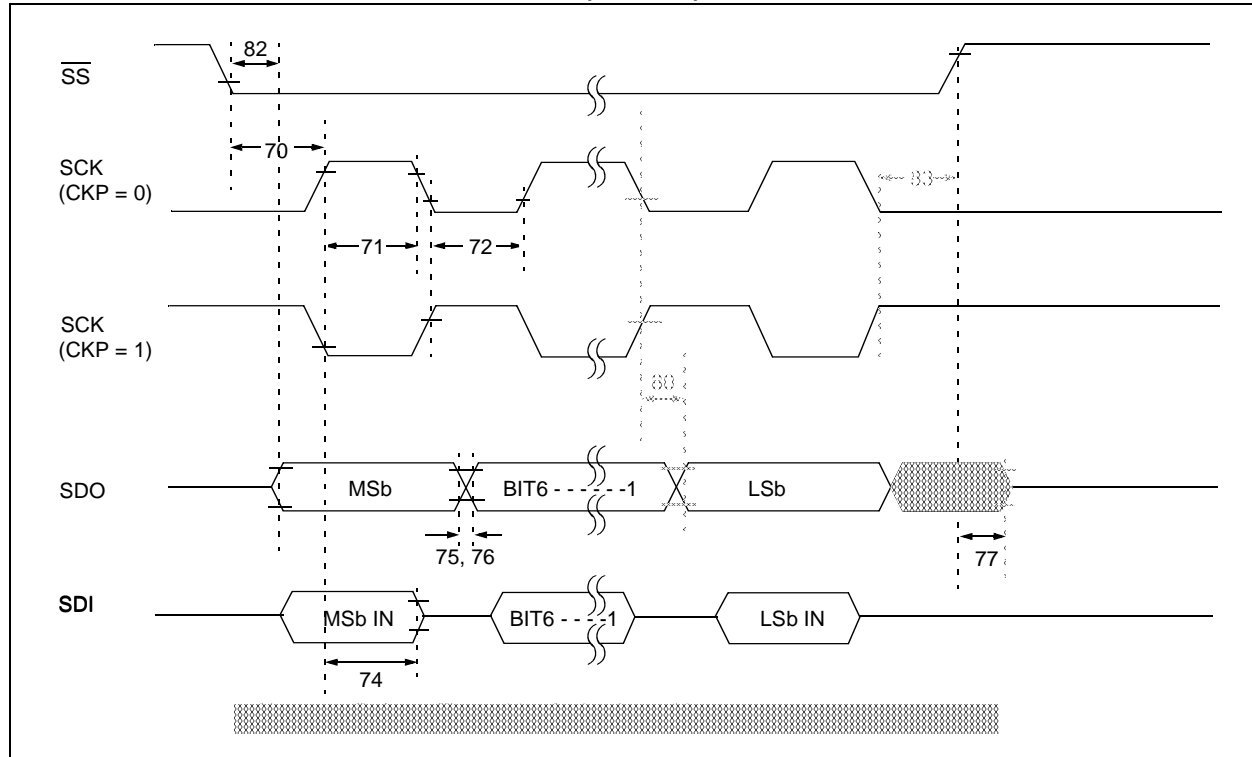


TABLE 20-11: SPI MODE REQUIREMENTS (SLAVE MODE, CKE = 1)

Param. No.	Symbol	Characteristic	Min	Typ†	Max	Units	Conditions
70	TssL2scH, TssL2scL	$\overline{SS}\downarrow$ to SCK \downarrow or SCK \uparrow input	Tcy	—	—	ns	
71	Tsch	SCK input high time (Slave mode)	Continuous	1.25Tcy + 30	—	ns	
71A		Single Byte	40	—	—	ns	(Note 1)
72	TscL	SCK input low time (Slave mode)	Continuous	1.25Tcy + 30	—	ns	
72A		Single Byte	40	—	—	ns	(Note 1)
73A	Tb2B	Last clock edge of Byte1 to the 1st clock edge of Byte2	1.5Tcy + 40	—	—	ns	(Note 1)
74	Tsch2diL, TscL2diL	Hold time of SDI data input to SCK edge	100	—	—	ns	
75	TdoR	SDO data output rise time	—	10	25	ns	
76	TdoF	SDO data output fall time	—	10	25	ns	
77	TssH2doZ	$\overline{SS}\uparrow$ to SDO output hi-impedance	10	—	50	ns	
80	Tsch2doV, TscL2doV	SDO data output valid after SCK edge	—	—	50	ns	
82	TssL2doV	SDO data output valid after $\overline{SS}\downarrow$ edge	—	—	50	ns	
83	Tsch2ssH, TscL2ssH	$\overline{SS}\uparrow$ after SCK edge	1.5Tcy + 40	—	—	ns	

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

Note 1: Specification 73A is only required if specifications 71A and 72A are used.

PIC17C7XX

FIGURE 20-21: USART ASYNCHRONOUS MODE START BIT DETECT

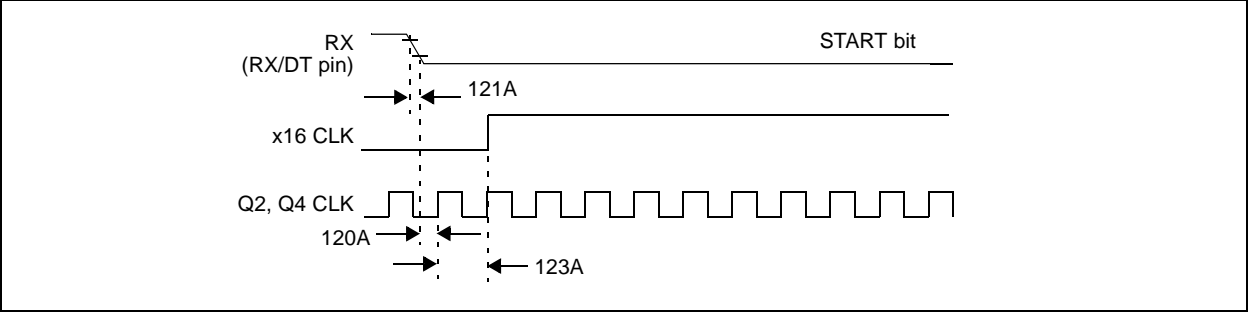


TABLE 20-16: USART ASYNCHRONOUS MODE START BIT DETECT REQUIREMENTS

Param No.	Sym	Characteristic	Min	Typ	Max	Unit s	Conditions
120A	TdtL2ckH	Time to ensure that the RX pin is sampled low	—	—	TcY	ns	
121A	TdtRF	Data rise time and fall time	—	—	(Note 1)	ns	
		Receive	—	—	40	ns	
123A	TckH2bckL	Time from RX pin sampled low to first rising edge of x16 clock	—	—	TcY	ns	

Note 1: Schmitt trigger will determine logic level.

FIGURE 20-22: USART ASYNCHRONOUS RECEIVE SAMPLING WAVEFORM

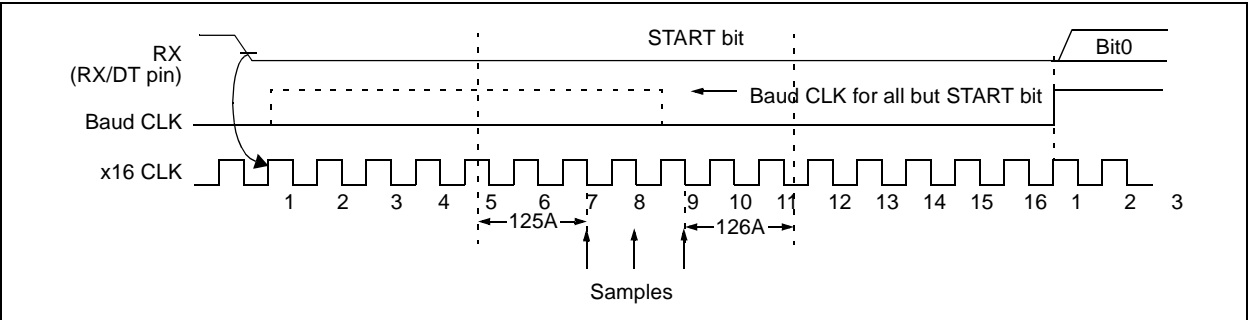


TABLE 20-17: USART ASYNCHRONOUS RECEIVE SAMPLING REQUIREMENTS

Param No.	Sym	Characteristic	Min	Typ	Max	Unit s	Conditions
125A	TdtL2ckH	Setup time of RX pin to first data sampled	TcY	—	—	ns	
126A	TdtL2ckH	Hold time of RX pin from last data sampled	TcY	—	—	ns	

FIGURE 21-19: TYPICAL, MINIMUM AND MAXIMUM V_{OH} vs. I_{OH} ($V_{DD} = 3V$, $-40^{\circ}C$ TO $+125^{\circ}C$)

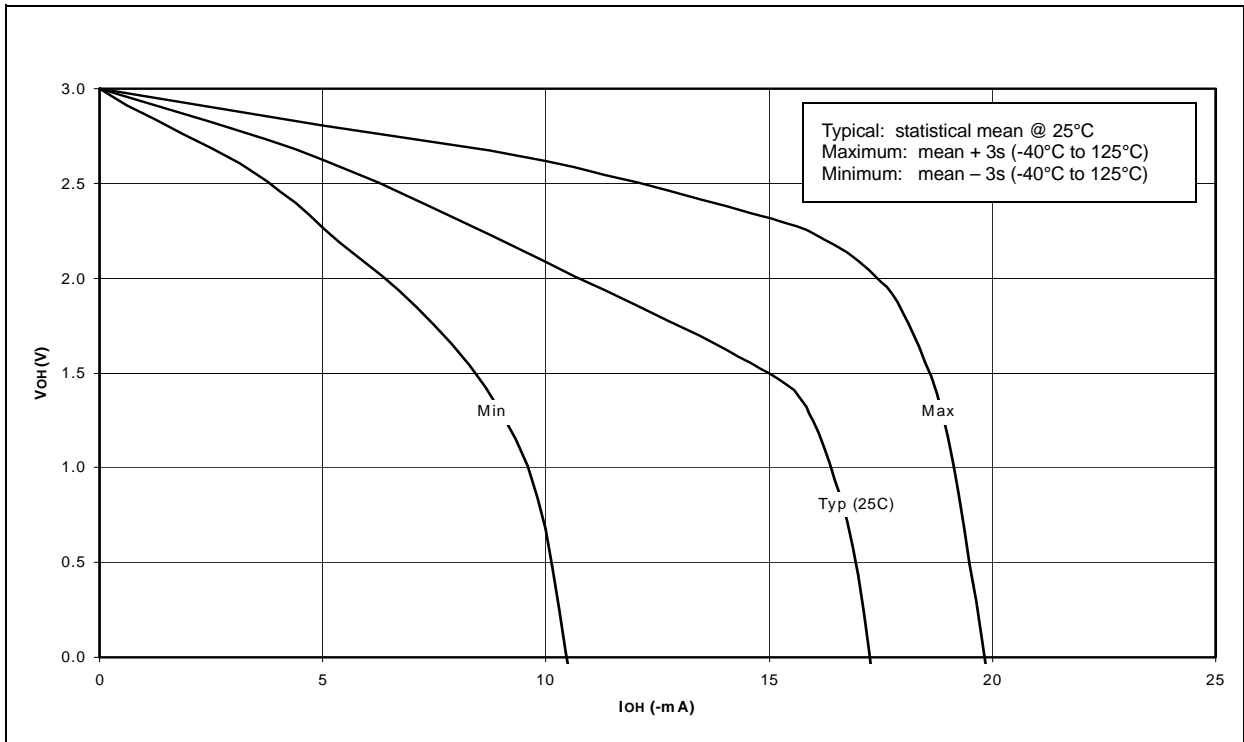
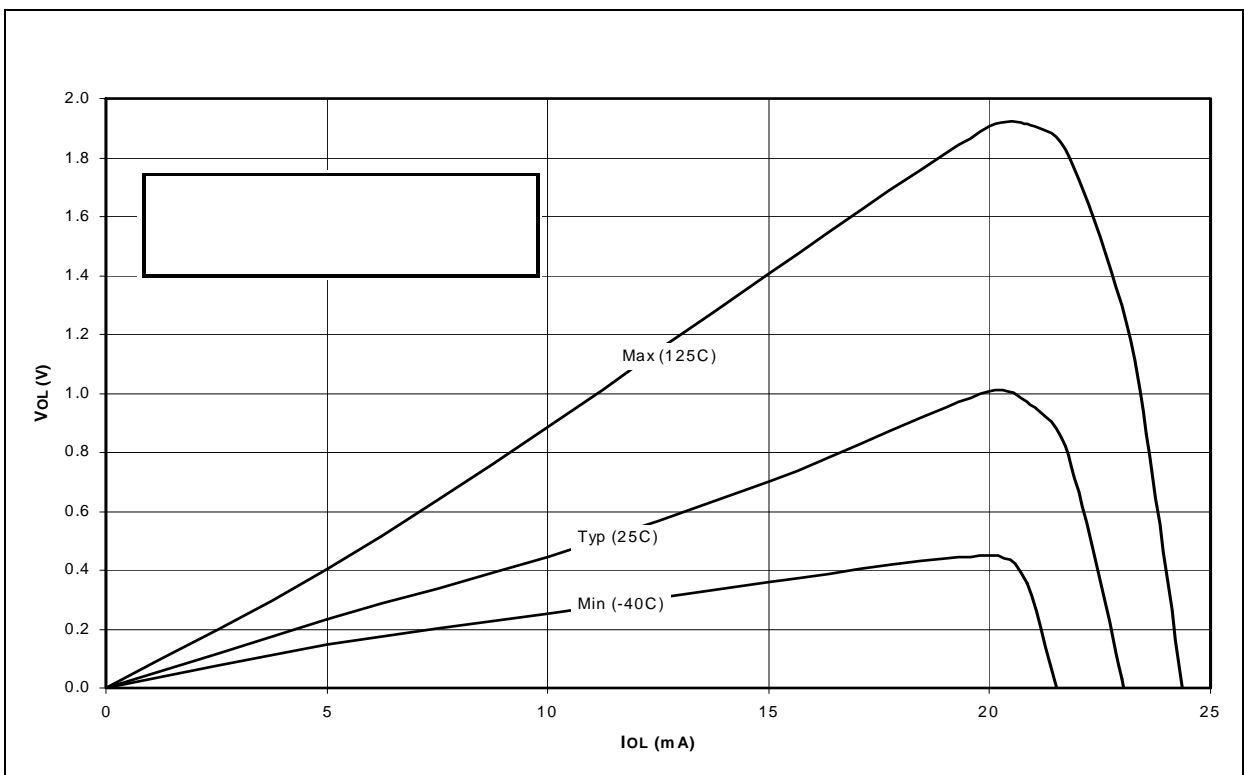


FIGURE 21-20: TYPICAL, MINIMUM AND MAXIMUM V_{OL} vs. I_{OL} ($V_{DD} = 3V$, $-40^{\circ}C$ TO $+125^{\circ}C$)



PIC17C7XX

BTG.....	206	Capture4 Interrupt	38
CALL.....	207	Context Saving	39
CLRF.....	207	Flag bits	
CLRWDT.....	208	TMR1IE	33
COMF.....	208	TMR1IF.....	33
CPFSEQ.....	209	TMR2IE	33
CPFSGT.....	209	TMR2IF.....	33
CPFSLT.....	210	TMR3IE	33
DAW.....	210	TMR3IF.....	33
DECF.....	211	Global Interrupt Disable.....	39
DECFSNZ.....	212	Interrupts	33
DECFSZ.....	211	Logic	33
GOTO.....	212	Operation	39
INCF.....	213	Peripheral Interrupt Enable.....	35
INCFSNZ.....	214	Peripheral Interrupt Request.....	37
INCFSZ.....	213	PIE2 Register	36
IORLW.....	214	PIR1 Register	37
IORWF.....	215	PIR2 Register	38
LCALL.....	215	PORTB Interrupt on Change	37
MOVFP.....	216	PWM.....	108
MOVLB.....	216	RA0/INT	39
MOVLR.....	217	Status Register	34
MOVLW.....	217	Synchronous Serial Port Interrupt.....	38
MOVFP.....	218	T0CKI Interrupt	39
MOVWF.....	218	Timing.....	40
MULLW.....	219	TMR1 Overflow Interrupt	37
MULWF.....	219	TMR2 Overflow Interrupt	37
NEGW.....	220	TMR3 Overflow Interrupt	37
NOP.....	220	USART1 Receive Interrupt	37
RETIE.....	221	USART1 Transmit Interrupt	37
RETLW.....	221	USART2 Receive Interrupt	38
RETURN.....	222	Vectors	
RLCF.....	222	Peripheral Interrupt.....	39
RLNCF.....	223	Program Memory Locations	43
RRCF.....	223	RA0/INT Interrupt	39
RRNCF.....	224	T0CKI Interrupt.....	39
SETF.....	224	Vectors/Priorities.....	39
SLEEP.....	225	Wake-up from SLEEP.....	194
SUBLW.....	225	INTF.....	34
SUBWF.....	226	INTSTA Register.....	34
SUBWFB.....	226	IORLW.....	214
SWAPF.....	227	IORWF.....	215
TABLRD.....	227, 228	IRBPU VS. VDD	274
TABLWT.....	228, 229	K	
TLRD.....	229	KeeLoq Evaluation and Programming Tools	236
TLWT.....	230	L	
TSTFSZ.....	230	LCALL.....	54, 215
XORLW.....	231	M	
XORWF.....	231	Maps	
Instruction Set Summary.....	197	Register File Map.....	47
Instructions		Memory	
TABLRD.....	64	External Interface	45
TLRD.....	64	External Memory Waveforms	45
INT Pin	40	Memory Map (Different Modes)	44
INTE.....	34	Mode Memory Access	44
INTEDG.....	53, 97	Organization	43
Inter-Integrated Circuit (I ² C).....	133	Program Memory	43
Internal Sampling Switch (R _{ss}) Impedance	183	Program Memory Map	43
Interrupt on Change Feature.....	74	Microcontroller	43
Interrupt Status Register (INTSTA)	34	Microprocessor	43
Interrupts		Minimizing Current Consumption.....	195
A/D Interrupt.....	38	MOVFP.....	46, 216
Bus Collision Interrupt.....	38	Moving Data Between Data and Program Memories	46
Capture1 Interrupt.....	37	MOVLB.....	46, 216
Capture2 Interrupt.....	37	MOVLR.....	217
Capture3 Interrupt.....	38	MOVLW.....	217

PIC17C7XX

Timer0	97	TMR0	98, 99
Timer1		TMR0 Read/Write in Timer Mode	100
16-bit Mode	105	TMR1, TMR2, and TMR3 in Timer Mode	115
Clock Source Select	101	Wake-Up from SLEEP	194
On bit	102, 103	TLRD	229
Section	101, 104	TLWT	230
Timer2		TMR0	
16-bit Mode	105	16-bit Read	99
Clock Source Select	101	16-bit Write	99
On bit	102, 103	Module	98
Section	101, 104	Operation	98
Timer3		Overview	95
Clock Source Select	101	Prescaler Assignments	99
On bit	102, 103	Read/Write Considerations	99
Section	101, 110	Read/Write in Timer Mode	100
Timers		Timing	98, 99
TCON3	103	TMR0 Status/Control Register (T0STA)	53
Timing Diagrams		TMR1	28, 49
A/D Conversion	264	8-bit Mode	104
Acknowledge Sequence Timing	165	External Clock Input	104
Asynchronous Master Transmission	123	Overview	95
Asynchronous Reception	126	Timer Mode	115
Back to Back Asynchronous Master Transmission	124	Two 8-bit Timer/Counter Mode	104
Baud Rate Generator with Clock Arbitration	153	Using with PWM	107
BRG Reset Due to SDA Collision	172	TMR1 Overflow Interrupt	37
Bus Collision		TMR1CS	101
START Condition Timing	171	TMR1IE	35
Bus Collision During a RESTART Condition		TMR1IF	37
(Case 1)	173	TMR1ON	102
Bus Collision During a RESTART Condition		TMR2	28, 49
(Case 2)	173	8-bit Mode	104
Bus Collision During a START Condition		External Clock Input	104
(SCL = 0)	172	In Timer Mode	115
Bus Collision During a		Two 8-bit Timer/Counter Mode	104
STOP Condition	174	Using with PWM	107
Bus Collision for Transmit and Acknowledge	170	TMR2 Overflow Interrupt	37
External Parallel Resonant Crystal Oscillator Circuit	19	TMR2CS	101
External Program Memory Access	45	TMR2IE	35
I ² C Bus Data	259	TMR2IF	37
I ² C Bus START/STOP bits	258	TMR2ON	102
I ² C Master Mode First START bit Timing	154	TMR3	
I ² C Master Mode Reception Timing	164	Example, Reading From	114
I ² C Master Mode Transmission Timing	161	Example, Writing To	114
Interrupt (INT, TMR0 Pins)	40	External Clock Input	114
Master Mode Transmit Clock Arbitration	169	In Timer Mode	115
Oscillator Start-up Time	24	One Capture and One Period Register Mode	110
PIC17C752/756 Capture Timing	253	Overview	95
PIC17C752/756 CLKOUT and I/O	250	Reading/Writing	114
PIC17C752/756 External Clock	249	TMR3 Interrupt Flag bit, TMR3IF	37
PIC17C752/756 Memory Interface Read	266	TMR3CS	101, 110
PIC17C752/756 Memory Interface Write	265	TMR3H	28, 49
PIC17C752/756 PWM Timing	253	TMR3IE	35
PIC17C752/756 Reset, Watchdog Timer, Oscillator		TMR3IF	37, 110
Start-up Timer and Power-up Timer	251	TMR3L	28, 49
PIC17C752/756 Timer0 Clock	252	TMR3ON	102, 110
PIC17C752/756 Timer1, Timer2 and Timer3 Clock	252	TO	52, 193, 194
PIC17C752/756 USART Module Synchronous		Transmit Status and Control Register	117
Receive	261	TSTFSZ	230
PIC17C752/756 USART Module		TTL INPUT	278
Synchronous Transmission	260	Turning on 16-bit Timer	105
Repeat START Condition	156	TX1IE	35
Slave Synchronization	140	TX1IF	37
STOP Condition Receive or Transmit	167	TX2IE	36
Synchronous Reception	129	TX2IF	38
Synchronous Transmission	128	TXREG	123, 127, 131, 132
Table Write	62	TXREG1	27, 48

PIC17C7XX

READER RESPONSE

It is our intention to provide you with the best documentation possible to ensure successful use of your Microchip product. If you wish to provide your comments on organization, clarity, subject matter, and ways in which our documentation can better serve you, please FAX your comments to the Technical Publications Manager at (480) 792-4150.

Please list the following information, and use this outline to provide us with your comments about this Data Sheet.

To: Technical Publications Manager
RE: Reader Response
From: Name _____
Company _____
Address _____
City / State / ZIP / Country _____
Telephone: (____) _____ - _____ FAX: (____) _____ - _____

Application (optional):

Would you like a reply? ____Y ____N

Device: **PIC17C7XX** Literature Number: **DS30289C**

Questions:

1. What are the best features of this document?

2. How does this document meet your hardware and software development needs?

3. Do you find the organization of this data sheet easy to follow? If not, why?

4. What additions to the data sheet do you think would enhance the structure and subject?

5. What deletions from the data sheet could be made without affecting the overall usefulness?

6. Is there any incorrect or misleading information (what and where)?

7. How would you improve this document?

8. How would you improve our software, systems, and silicon products?
