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	EPROM, UV
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	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	84-CLCC Window (J-Lead)
Supplier Device Package	84-CERQUAD (29.21x29.21)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic17c766-cl

TABLE 3-1: PINOUT DESCRIPTIONS (CONTINUED)

Name	PIC17C75X			PIC17C76X		I/O/P Type	Buffer Type	Description
	DIP No.	PLCC No.	TQFP No.	PLCC No.	QFP No.			
								PORTC is a bi-directional I/O Port.
RC0/AD0	2	3	58	3	72	I/O	TTL	This is also the least significant byte (LSB) of the 16-bit wide system bus in Microprocessor mode or Extended Microcontroller mode. In multiplexed system bus configuration, these pins are address output as well as data input or output.
RC1/AD1	63	67	55	83	69	I/O	TTL	
RC2/AD2	62	66	54	82	68	I/O	TTL	
RC3/AD3	61	65	53	81	67	I/O	TTL	
RC4/AD4	60	64	52	80	66	I/O	TTL	
RC5/AD5	58	63	51	79	65	I/O	TTL	
RC6/AD6	58	62	50	78	64	I/O	TTL	
RC7/AD7	57	61	49	77	63	I/O	TTL	
								PORTD is a bi-directional I/O Port.
RD0/AD8	10	11	2	15	4	I/O	TTL	This is also the most significant byte (MSB) of the 16-bit system bus in Microprocessor mode or Extended Microcontroller mode. In multiplexed system bus configuration, these pins are address output as well as data input or output.
RD1/AD9	9	10	1	14	3	I/O	TTL	
RD2/AD10	8	9	64	9	78	I/O	TTL	
RD3/AD11	7	8	63	8	77	I/O	TTL	
RD4/AD12	6	7	62	7	76	I/O	TTL	
RD5/AD13	5	6	61	6	75	I/O	TTL	
RD6/AD14	4	5	60	5	74	I/O	TTL	
RD7/AD15	3	4	59	4	73	I/O	TTL	
								PORTE is a bi-directional I/O Port.
RE0/ALE	11	12	3	16	5	I/O	TTL	In Microprocessor mode or Extended Microcontroller mode, RE0 is the Address Latch Enable (ALE) output. Address should be latched on the falling edge of ALE output.
RE1/$\overline{OE}$	12	13	4	17	6	I/O	TTL	In Microprocessor or Extended Microcontroller mode, RE1 is the Output Enable ($\overline{OE}$) control output (active low).
RE2/$\overline{WR}$	13	14	5	18	7	I/O	TTL	In Microprocessor or Extended Microcontroller mode, RE2 is the Write Enable ($\overline{WR}$) control output (active low).
RE3/CAP4	14	15	6	19	8	I/O	ST	RE3 can also be the Capture4 input pin.
								PORTF is a bi-directional I/O Port.
RF0/AN4	26	28	18	36	24	I/O	ST	RF0 can also be analog input 4.
RF1/AN5	25	27	17	35	23	I/O	ST	RF1 can also be analog input 5.
RF2/AN6	24	26	16	30	18	I/O	ST	RF2 can also be analog input 6.
RF3/AN7	23	25	15	29	17	I/O	ST	RF3 can also be analog input 7.
RF4/AN8	22	24	14	28	16	I/O	ST	RF4 can also be analog input 8.
RF5/AN9	21	23	13	27	15	I/O	ST	RF5 can also be analog input 9.
RF6/AN10	20	22	12	26	14	I/O	ST	RF6 can also be analog input 10.
RF7/AN11	19	21	11	25	13	I/O	ST	RF7 can also be analog input 11.

Legend: I = Input only; O = Output only; I/O = Input/Output;
P = Power; — = Not Used; TTL = TTL input; ST = Schmitt Trigger input

Note 1: The output is only available by the peripheral operation.

2: Open drain input/output pin. Pin forced to input upon any device RESET.

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.

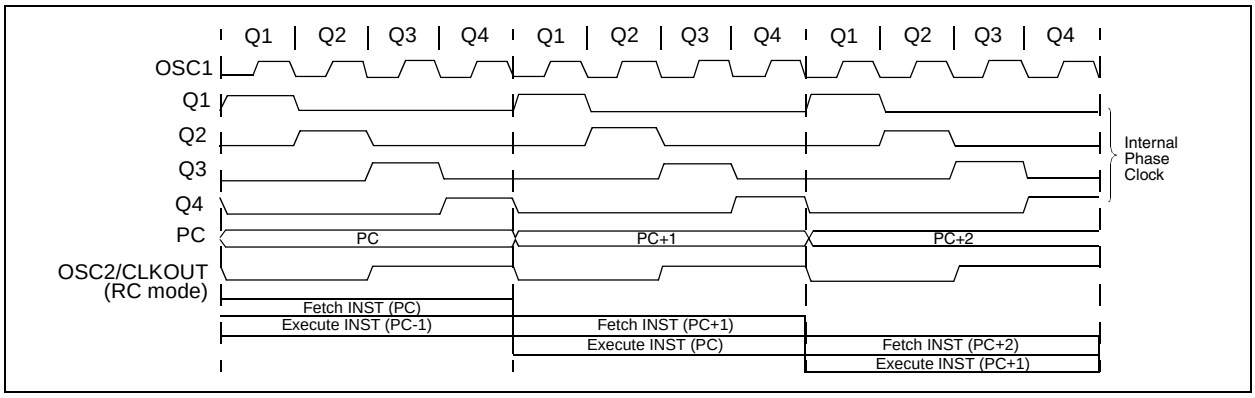
4.3 Instruction Flow/Pipelining

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).

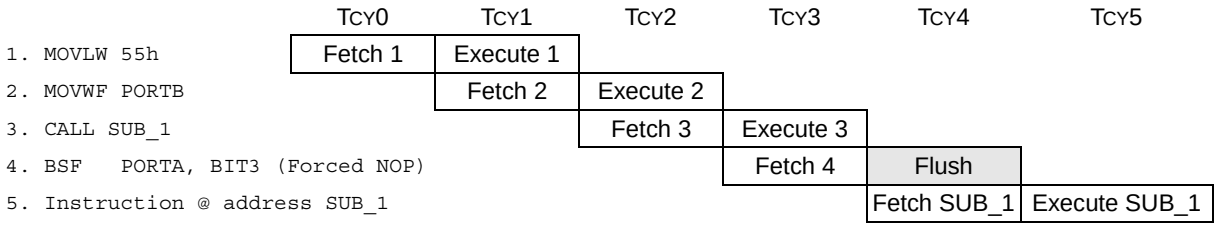
A fetch cycle begins with the program counter incrementing in Q1.

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



All instructions are single cycle, except for any program branches. These take two cycles since the fetched instruction is “flushed” from the pipeline, while the new instruction is being fetched and then executed.

NOTES:

10.3 PORTC and DDRC Registers

PORTC is an 8-bit bi-directional port. The corresponding data direction register is DDRC. A '1' in DDRC configures the corresponding port pin as an input. A '0' in the DDRC register configures the corresponding port pin as an output. Reading PORTC reads the status of the pins, whereas writing to PORTC will write to the port latch. PORTC is multiplexed with the system bus. When operating as the system bus, PORTC is the low order byte of the address/data bus (AD7:AD0). The timing for the system bus is shown in the Electrical Specifications section.

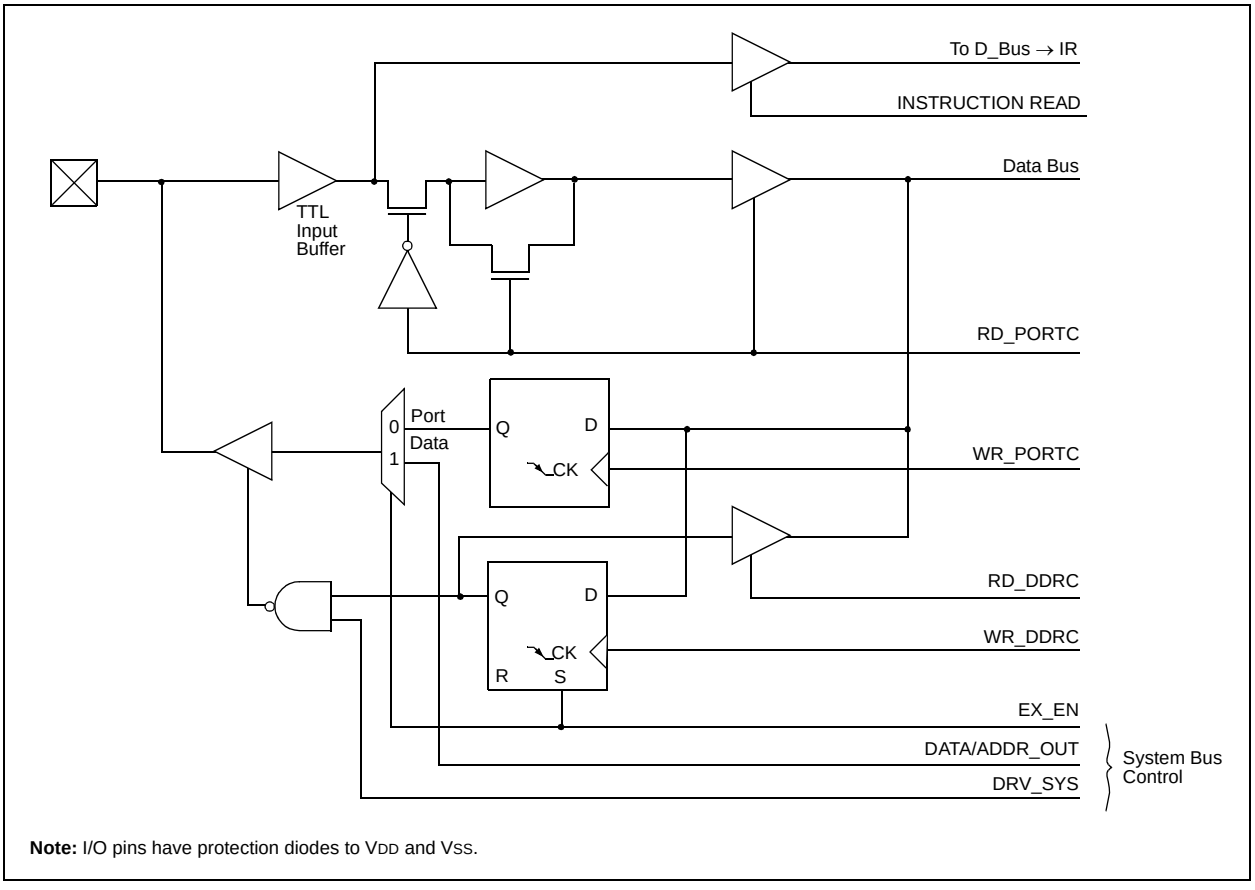
Note:	This port is configured as the system bus when the device's configuration bits are selected to Microprocessor or Extended Microcontroller modes. In the two other microcontroller modes, this port is a general purpose I/O.

Example 10-3 shows an instruction sequence to initialize PORTC. The Bank Select Register (BSR) must be selected to Bank 1 for the port to be initialized. The following example uses the `MOVLB` instruction to load the BSR register for bank selection.

EXAMPLE 10-3: INITIALIZING PORTC

```
MOVLB   1          ; Select Bank 1
CLRF    PORTC, F   ; Initialize PORTC data
                   ; latches before setting
                   ; the data direction reg
MOVLW    0xCF      ; Value used to initialize
                   ; data direction
MOVWF    DDRC      ; Set RC<3:0> as inputs
                   ; RC<5:4> as outputs
                   ; RC<7:6> as inputs
```

FIGURE 10-9: BLOCK DIAGRAM OF RC7:RC0 PORT PINS



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

TABLE 13-3: SUMMARY OF TIMER1, TIMER2 AND TIMER3 REGISTERS

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

Legend: x = unknown, u = unchanged, - = unimplemented, read as a '0', q = value depends on condition.
Shaded cells are not used by Timer1 or Timer2.

This mode (3 Capture, 1 Period) is selected if control bit CA1/PR3 is clear. In this mode, the Capture1 register, consisting of high byte (PR3H/CA1H) and low byte (PR3L/CA1L), is configured as the period control register for TMR3. Capture1 is disabled in this mode and the corresponding interrupt bit, CA1IF, is never set. TMR3 increments until it equals the value in the period register and then resets to 0000h on the next timer clock.

All other Captures are active in this mode.

13.2.1.1 Capture Operation

The CAxED1 and CAxED0 bits determine the event on which capture will occur. The possible events are:

- Capture on every falling edge
- Capture on every rising edge
- Capture every 4th rising edge
- Capture every 16th rising edge

When a capture takes place, an interrupt flag is latched into the CAxIF bit. This interrupt can be enabled by setting the corresponding mask bit CAxIE. The Peripheral Interrupt Enable bit (PEIE) must be set and the Global Interrupt Disable bit (GLINTD) must be cleared for the interrupt to be acknowledged. The CAxIF interrupt flag bit is cleared in software.

When the capture prescale select is changed, the prescaler is not reset and an event may be generated. Therefore, the first capture after such a change will be ambiguous. However, it sets the time-base for the next capture. The prescaler is reset upon chip RESET.

The capture pin, CAPx, is a multiplexed pin. When used as a port pin, the capture is not disabled. However, the user can simply disable the Capture interrupt by clearing CAxIE. If the CAPx pin is used as an output pin, the user can activate a capture by writing to the port pin. This may be useful during development phase to emulate a capture interrupt.

The input on the capture pin CAPx is synchronized internally to internal phase clocks. This imposes certain restrictions on the input waveform (see the Electrical Specification section for timing).

The capture overflow status flag bit is double buffered. The master bit is set if one captured word is already residing in the Capture register (CAxH:CAxL) and another “event” has occurred on the CAPx pin. The new event will not transfer the TMR3 value to the capture register, protecting the previous unread capture value. When the user reads both the high and the low bytes (in any order) of the Capture register, the master overflow bit is transferred to the slave overflow bit (CAxOVF) and then the master bit is reset. The user can then read TCONx to determine the value of CAxOVF.

The recommended sequence to read capture registers and capture overflow flag bits is shown in Example 13-1.

REGISTER 15-1: SSPSTAT: SYNC SERIAL PORT STATUS REGISTER (ADDRESS: 13h, BANK 6)

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF
bit 7							bit 0

bit 7 **SMP:** Sample bit

SPI Master mode:
1 = Input data sampled at end of data output time
0 = Input data sampled at middle of data output time

SPI Slave mode:
SMP must be cleared when SPI is used in Slave mode

In I^2C Master or Slave mode:
1 = Slew rate control disabled for Standard Speed mode (100 kHz and 1 MHz)
0 = Slew rate control enabled for High Speed mode (400 kHz)

bit 6 **CKE**: SPI Clock Edge Select (Figure 15-6, Figure 15-8 and Figure 15-9)

CKP = 0:
1 = Data transmitted on rising edge of SCK
0 = Data transmitted on falling edge of SCK

CKP = 1:
1 = Data transmitted on falling edge of SCK
0 = Data transmitted on rising edge of SCK

bit 5 **D/$\overline{A}$**: Data/$\overline{Address}$ bit (I^2C mode only)
1 = Indicates that the last byte received or transmitted was data
0 = Indicates that the last byte received or transmitted was address

bit 4 **P**: STOP bit
(I^2C mode only. This bit is cleared when the MSSP module is disabled, SSPEN is cleared.)
1 = Indicates that a STOP bit has been detected last (this bit is '0' on RESET)
0 = STOP bit was not detected last

bit 3 **S**: START bit
(I^2C mode only. This bit is cleared when the MSSP module is disabled, SSPEN is cleared.)
1 = Indicates that a START bit has been detected last (this bit is '0' on RESET)
0 = START bit was not detected last

bit 2 **R/$\overline{W}$**: Read/Write bit Information (I^2C mode only)
This bit holds the R/W bit information following the last address match. This bit is only valid from the address match to the next START bit, STOP bit, or not ACK bit.

In I^2C Slave mode:
1 = Read
0 = Write

In I^2C Master mode:
1 = Transmit is in progress
0 = Transmit is not in progress
Or'ing this bit with SEN, RSEN, PEN, RCEN, or ACKEN will indicate if the MSSP is in IDLE mode.

bit 1 **UA**: Update Address (10-bit I^2C mode only)
1 = Indicates that the user needs to update the address in the SSPADD register
0 = Address does not need to be updated

bit 0 **BF**: Buffer Full Status bit

Receive (SPI and I^2C modes)
1 = Receive complete, SSPBUF is full
0 = Receive not complete, SSPBUF is empty

Transmit (I^2C mode only)
1 = Data transmit in progress (does not include the $\overline{ACK}$ and STOP bits), SSPBUF is full
0 = Data transmit complete (does not include the $\overline{ACK}$ and STOP bits), SSPBUF is empty

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

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

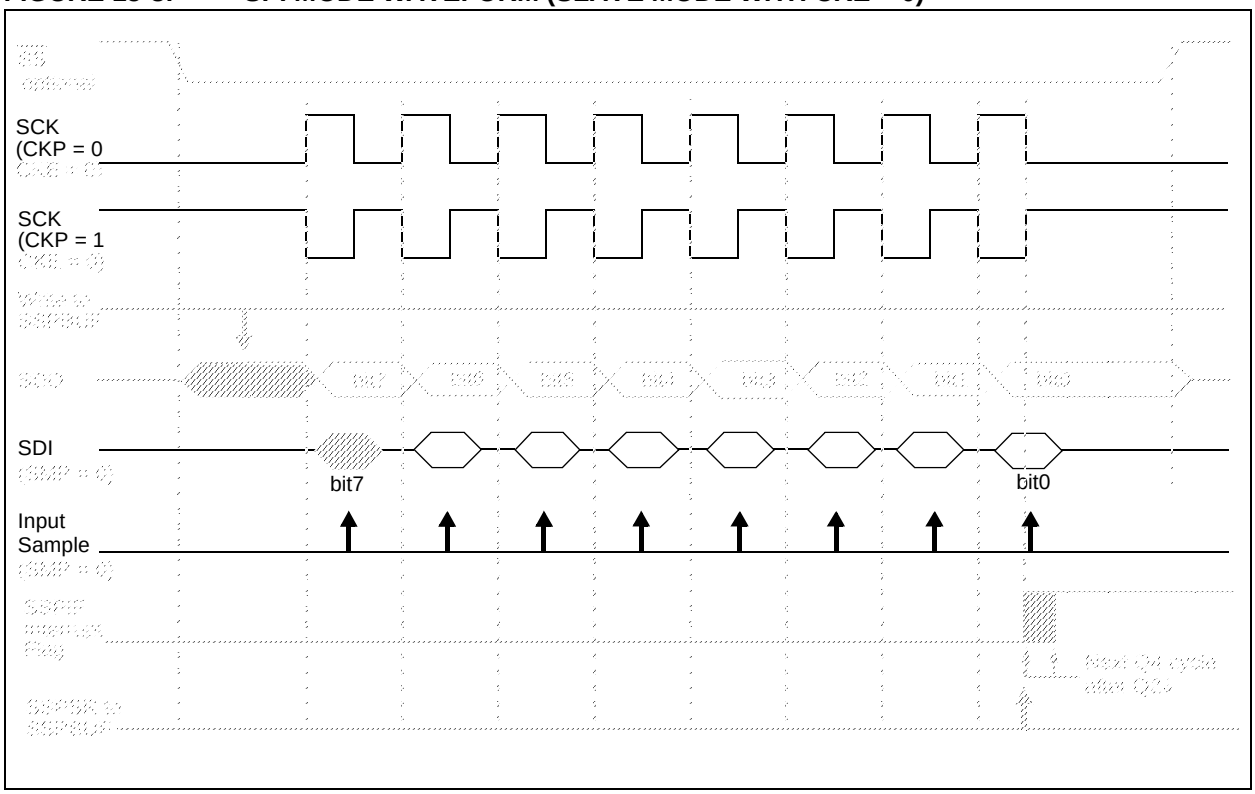
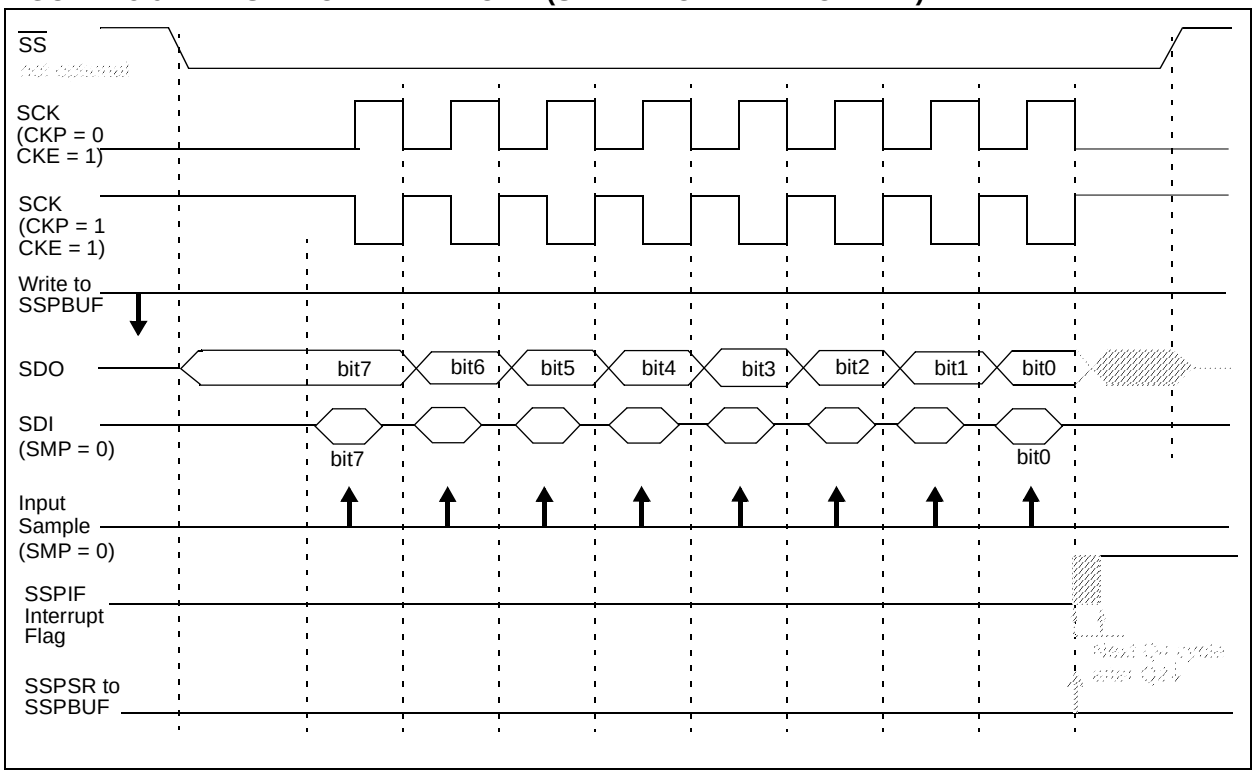


FIGURE 15-9: SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 1)



15.4 Example Program

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

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

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

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

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

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

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

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

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

16.0 ANALOG-TO-DIGITAL CONVERTER (A/D) MODULE

The analog-to-digital (A/D) converter module has twelve analog inputs for the PIC17C75X devices and sixteen for the PIC17C76X devices.

The analog input charges a sample and hold capacitor. The output of the sample and hold capacitor is the input into the converter. The converter then generates a digital result of this analog level via successive approximation. This A/D conversion of the analog input signal, results in a corresponding 10-bit digital number.

The analog reference voltages (positive and negative supply) are software selectable to either the device's supply voltages (AVDD, AVss), or the voltage level on the RG3/AN0/VREF+ and RG2/AN1/VREF- pins.

The A/D converter has a unique feature of being able to operate while the device is in SLEEP mode. To operate in SLEEP, the A/D clock must be derived from the A/D's internal RC oscillator.

The A/D module has four registers. These registers are:

- A/D Result High Register (ADRESH)
- A/D Result Low Register (ADRESL)
- A/D Control Register0 (ADCON0)
- A/D Control Register1 (ADCON1)

The ADCON0 register, shown in Register 16-1, controls the operation of the A/D module. The ADCON1 register, shown in Register 16-2, configures the functions of the port pins. The port pins can be configured as analog inputs (RG3 and RG2 can also be the voltage references), or as digital I/O.

REGISTER 16-1: ADCON0 REGISTER (ADDRESS: 14h, BANK 5)

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	U-0	R/W-0
CHS3	CHS2	CHS1	CHS0	—	GO/$\overline{\text{DONE}}$	—	$\overline{\text{ADON}}$
bit 7							bit 0

bit 7-4 **CHS3:CHS0**: Analog Channel Select bits
0000 = channel 0, (AN0)
0001 = channel 1, (AN1)
0010 = channel 2, (AN2)
0011 = channel 3, (AN3)
0100 = channel 4, (AN4)
0101 = channel 5, (AN5)
0110 = channel 6, (AN6)
0111 = channel 7, (AN7)
1000 = channel 8, (AN8)
1001 = channel 9, (AN9)
1010 = channel 10, (AN10)
1011 = channel 11, (AN11)
1100 = channel 12, (AN12) (PIC17C76X only)
1101 = channel 13, (AN13) (PIC17C76X only)
1110 = channel 14, (AN14) (PIC17C76X only)
1111 = channel 15, (AN15) (PIC17C76X only)
11xx = **RESERVED**, do not select (PIC17C75X only)

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

bit 2 **GO/$\overline{\text{DONE}}$:** A/D Conversion Status bit

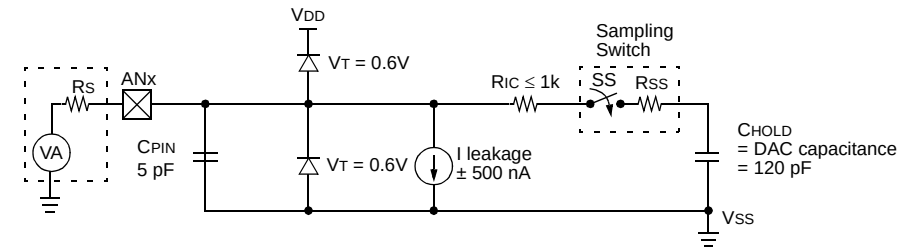
If ADON = 1:
1 = A/D conversion in progress (setting this bit starts the A/D conversion, which is automatically cleared by hardware when the A/D conversion is complete)
0 = A/D conversion not in progress

bit 1 **Unimplemented**: Read as '0'

bit 0 **ADON**: A/D On bit
1 = A/D converter module is operating
0 = A/D converter module is shut-off and consumes no operating current

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

FIGURE 16-3: ANALOG INPUT MODEL



Legend		
	C_{PIN}	= input capacitance
	V_T	= threshold voltage
	I leakage	= leakage current at the pin due to various junctions
	R_{IC}	= interconnect resistance
	SS	= sampling switch
	C_{HOLD}	= sample/hold capacitance (from DAC)

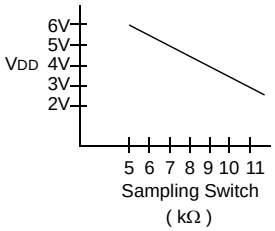


Table 18-2 lists the instructions recognized by the MPASM assembler.

Note 1:	Any unused opcode is Reserved. Use of any reserved opcode may cause unexpected operation.

All instruction examples use the following format to represent a hexadecimal number:

0xhh

where h signifies a hexadecimal digit.

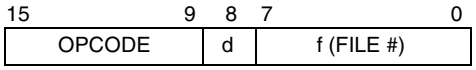
To represent a binary number:

`0000 0100b`

where b signifies a binary string.

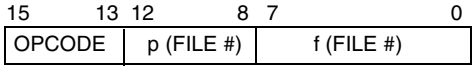
FIGURE 18-1: GENERAL FORMAT FOR INSTRUCTIONS

Byte-oriented file register operations



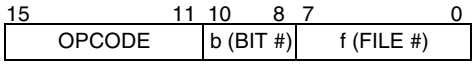
d = 0 for destination WREG
d = 1 for destination f
f = 8-bit file register address

Byte to Byte move operations



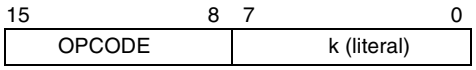
p = peripheral register file address
f = 8-bit file register address

Bit-oriented file register operations



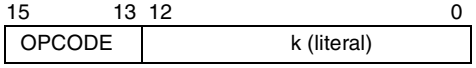
b = 3-bit address
f = 8-bit file register address

Literal and control operations



k = 8-bit immediate value

CALL and GOTO operations



k = 13-bit immediate value

18.1 Special Function Registers as Source/Destination

The PIC17C7XX's orthogonal instruction set allows read and write of all file registers, including special function registers. There are some special situations the user should be aware of:

18.1.1 ALUSTA AS DESTINATION

If an instruction writes to ALUSTA, the Z, C, DC and OV bits may be set or cleared as a result of the instruction and overwrite the original data bits written. For example, executing `CLRF ALUSTA` will clear register ALUSTA and then set the Z bit leaving `0000 0100b` in the register.

18.1.2 PCL AS SOURCE OR DESTINATION

Read, write or read-modify-write on PCL may have the following results:

Read PC:	PCH → PCLATH; PCL → dest
Write PCL:	PCLATH → PCH; 8-bit destination value → PCL
Read-Modify-Write:	PCL→ ALU operand PCLATH → PCH; 8-bit result → PCL

Where PCH = program counter high byte (not an addressable register), PCLATH = Program counter high holding latch, dest = destination, WREG or f.

18.1.3 BIT MANIPULATION

All bit manipulation instructions are done by first reading the entire register, operating on the selected bit and writing the result back (read-modify-write (R-M-W)). The user should keep this in mind when operating on some special function registers, such as ports.

Note:	Status bits that are manipulated by the device (including the interrupt flag bits) are set or cleared in the Q1 cycle. So, there is no issue on doing R-M-W instructions on registers which contain these bits

IORWF **Inclusive OR WREG with f**

Syntax: [*label*] IORWF f,d

Operands: $0 \le f \le 255$
$d \in [0,1]$

Operation: (WREG) .OR. (f) → (dest)

Status Affected: $\overline{Z}$

Encoding:

0000	100d	ffff	ffff

Description: Inclusive OR WREG with register 'f'. If 'd' is 0, the result is placed in WREG. If 'd' is 1, the result is placed back in register 'f'.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example: `IORWF RESULT, 0`

Before Instruction
RESULT = 0x13
WREG = 0x91

After Instruction
RESULT = 0x13
WREG = 0x93

LCALL **Long Call**

Syntax: [*label*] LCALL k

Operands: $0 \le k \le 255$

Operation: PC + 1 → TOS;
k → PCL, (PCLATH) → PCH

Status Affected: None

Encoding:

1011	0111	kkkk	kkkk

Description: `LCALL` allows an unconditional subroutine call to anywhere within the 64K program memory space.

First, the return address (PC + 1) is pushed onto the stack. A 16-bit destination address is then loaded into the program counter. The lower 8-bits of the destination address are embedded in the instruction. The upper 8-bits of PC are loaded from PC high holding latch, PCLATH.

Words: 1

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process Data	Write register PCL
No operation	No operation	No operation	No operation

Example:
```
MOVLW  HIGH(SUBROUTINE)
MOVPF  WREG, PCLATH
LCALL  LOW(SUBROUTINE)
```

Before Instruction
SUBROUTINE = 16-bit Address
PC = ?

After Instruction
PC = Address (SUBROUTINE)

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +125°C for extended -40°C ≤ TA ≤ +85°C for industrial 0°C ≤ TA ≤ +70°C for commercial Operating voltage VDD range as described in Section 20.1				
Param. No.	**Sym**	**Characteristic**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
		Input Leakage Current (Notes 2, 3)					
D060	IIL	I/O ports (except RA2, RA3)	–	–	±1	µA	Vss ≤ VPIN ≤ VDD, I/O Pin (in digital mode) at hi-impedance PORTB weak pull-ups disabled
D061		$\overline{\text{MCLR}}$, TEST	–	–	±2	µA	VPIN = Vss or VPIN = VDD
D062		RA2, RA3			±2	µA	Vss ≤ VRA2, VRA3 ≤ 12V
D063		OSC1 (EC, RC modes)	–	–	±1	µA	Vss ≤ VPIN ≤ VDD
D063B		OSC1 (XT, LF modes)	–	–	VPIN	µA	RF ≥ 1 MΩ
D064		$\overline{\text{MCLR}}$, TEST	–	–	25	µA	VMCLR = VPP = 12V (when not programming)
D070	IPURB	**PORTB Weak Pull-up Current**	85	130	260	µA	VPIN = VSS, $\overline{\text{RBPU}}$ = 0 4.5V ≤ VDD ≤ 5.5V
		Output Low Voltage					
D080	VOL	I/O ports					IOL = VDD/1.250 mA
			–	–	0.1VDD	V	4.5V ≤ VDD ≤ 5.5V
			–	–	0.1VDD	V	VDD = 3.0V
D081		with TTL buffer	–	–	0.4	V	IOL = 6 mA, VDD = 4.5V **(Note 6)**
D082		RA2 and RA3	–	–	3.0	V	IOL = 60.0 mA, VDD = 5.5V
			–	–	0.6	V	IOL = 60.0 mA, VDD = 4.5V
D083		OSC2/CLKOUT	–	–	0.4	V	IOL = 1 mA, VDD = 4.5V
D084		(RC and EC osc modes)	–	–	0.1VDD	V	IOL = VDD/5 mA (PIC17LC7XX only)
		Output High Voltage (Note 3)					
D090	VOH	I/O ports (except RA2 and RA3)					IOH = -VDD/2.5 mA
			0.9VDD	–	–	V	4.5V ≤ VDD ≤ 5.5V
			0.9VDD	–	–	V	VDD = 3.0V
D091		with TTL buffer	2.4	–	–	V	IOH = -6.0 mA, VDD = 4.5V **(Note 6)**
D093		OSC2/CLKOUT	2.4	–	–	V	IOH = -5 mA, VDD = 4.5V
D094		(RC and EC osc modes)	0.9VDD	–	–	V	IOH = -VDD/5 mA (PIC17LC7XX only)

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

Note 1: In RC oscillator configuration, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended that the PIC17CXXX devices be driven with external clock in RC mode.

2: The leakage current on the $\overline{\text{MCLR}}$ pin is strongly dependent on the applied voltage level. Higher leakage current may be measured at different input voltages.

3: Negative current is defined as current sourced by the pin.

4: These specifications are for the programming of the on-chip program memory EPROM through the use of the table write instructions. The complete programming specifications can be found in: PIC17C7XX Programming Specifications (Literature number DS TBD).

5: The $\overline{\text{MCLR}}$/VPP pin may be kept in this range at times other than programming, but is not recommended.

6: For TTL buffers, the better of the two specifications may be used.

FIGURE 20-9: TIMER0 EXTERNAL CLOCK TIMINGS

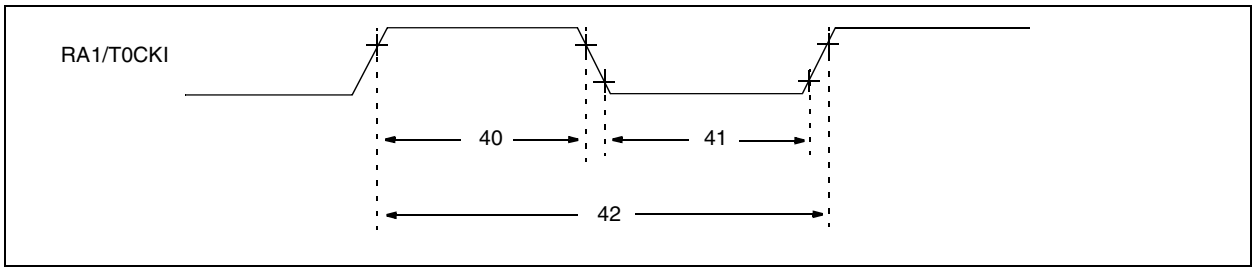


TABLE 20-4: TIMER0 EXTERNAL CLOCK REQUIREMENTS

Param No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
40	Tt0H	T0CKI High Pulse Width	No Prescaler	0.5TCY + 20	—	—	ns	
			With Prescaler	10	—	—	ns	
41	Tt0L	T0CKI Low Pulse Width	No Prescaler	0.5TCY + 20	—	—	ns	
			With Prescaler	10	—	—	ns	
42	Tt0P	T0CKI Period		Greater of: 20 ns or $\frac{T_{CY} + 40}{N}$	—	—	ns	N = prescale value (1, 2, 4, ..., 256)

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

FIGURE 20-10: TIMER1, TIMER2 AND TIMER3 EXTERNAL CLOCK TIMINGS

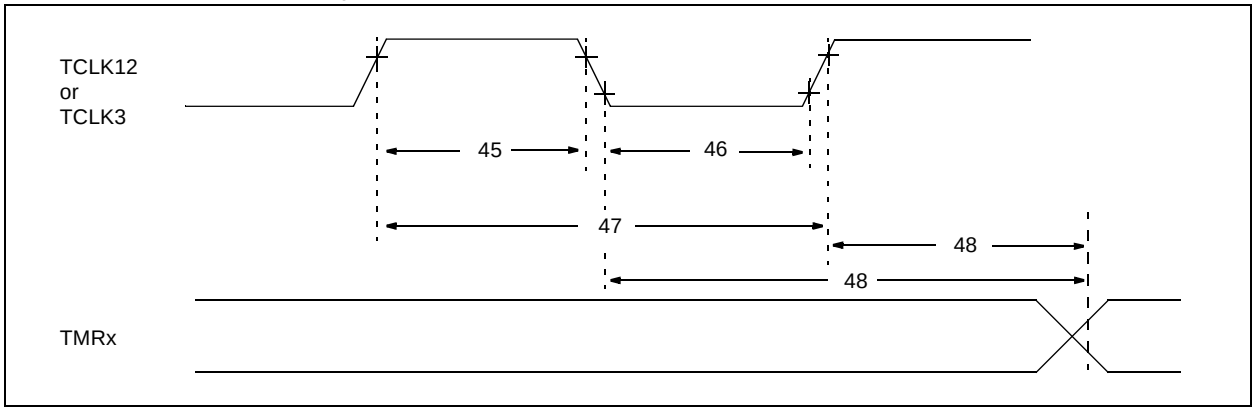


TABLE 20-5: TIMER1, TIMER2 AND TIMER3 EXTERNAL CLOCK REQUIREMENTS

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
45	Tt123H	TCLK12 and TCLK3 high time	0.5TCY + 20	—	—	ns	
46	Tt123L	TCLK12 and TCLK3 low time	0.5TCY + 20	—	—	ns	
47	Tt123P	TCLK12 and TCLK3 input period	$\frac{T_{CY} + 40}{N}$	—	—	ns	N = prescale value (1, 2, 4, 8)
48	TckE2tmrI	Delay from selected External Clock Edge to Timer increment	2TOSC	—	6Tosc	—	

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

21.0 PIC17C7XX DC AND AC CHARACTERISTICS

The graphs and tables provided in this section are for design guidance and are not tested nor guaranteed. In some graphs or tables the data presented is outside specified operating range (e.g., outside specified VDD range). This is for information only and devices are ensured to operate properly only within the specified range.

The data presented in this section is a statistical summary of data collected on units from different lots over a period of time.

- **Typ** or **Typical** represents the mean of the distribution at 25°C.
- **Max** or **Maximum** represents (mean + 3σ) over the temperature range of -40°C to 85°C.
- **Min** or **Minimum** represents (mean - 3σ) over the temperature range of -40°C to 85°C.

Note: Standard deviation is denoted by sigma (σ).

TABLE 21-1: PIN CAPACITANCE PER PACKAGE TYPE

Pin Name	Typical Capacitance (pF)	
	68-pin PLCC	64-pin TQFP
All pins, except $\overline{MCLR}$, VDD, and VSS	10	10
$\overline{MCLR}$ pin	20	20

FIGURE 21-1: TYPICAL RC OSCILLATOR FREQUENCY vs. TEMPERATURE

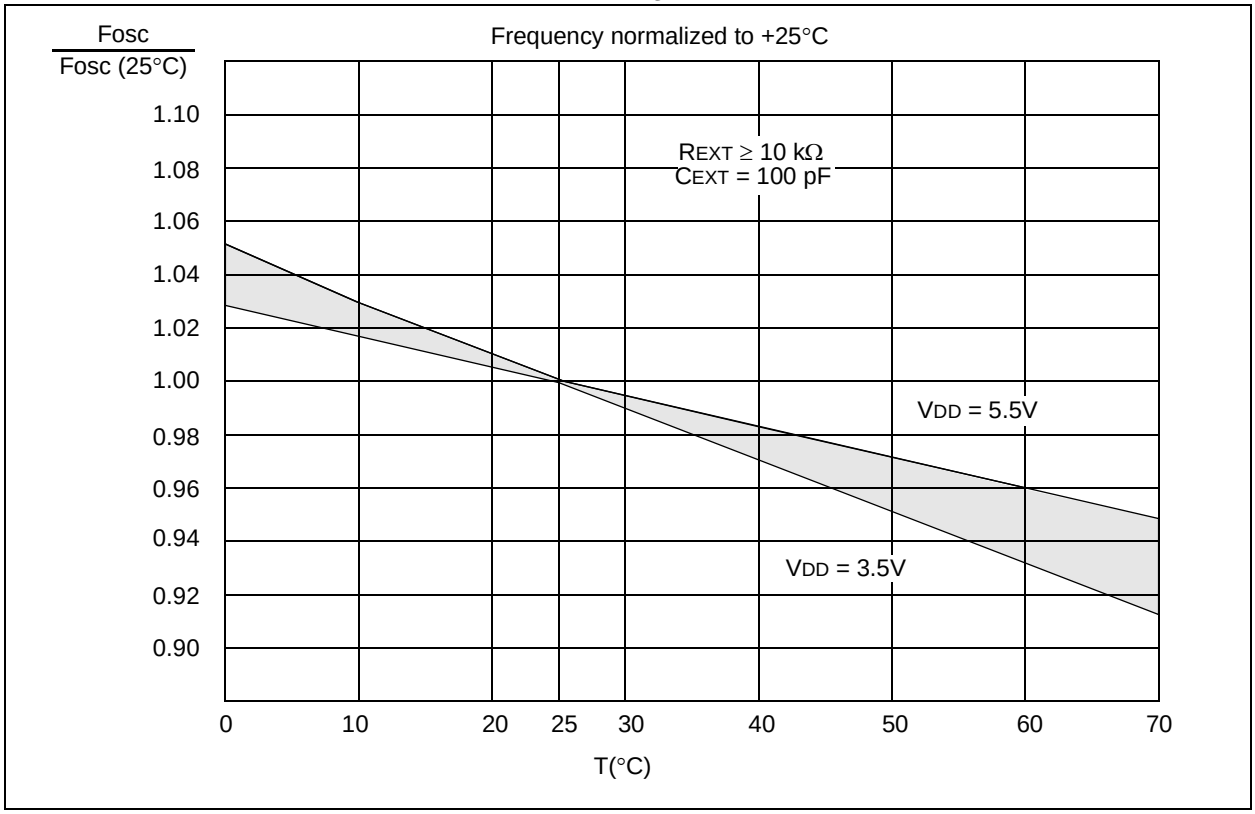


FIGURE 21-9: TYPICAL IDD vs. FOSC OVER VDD (XT MODE)

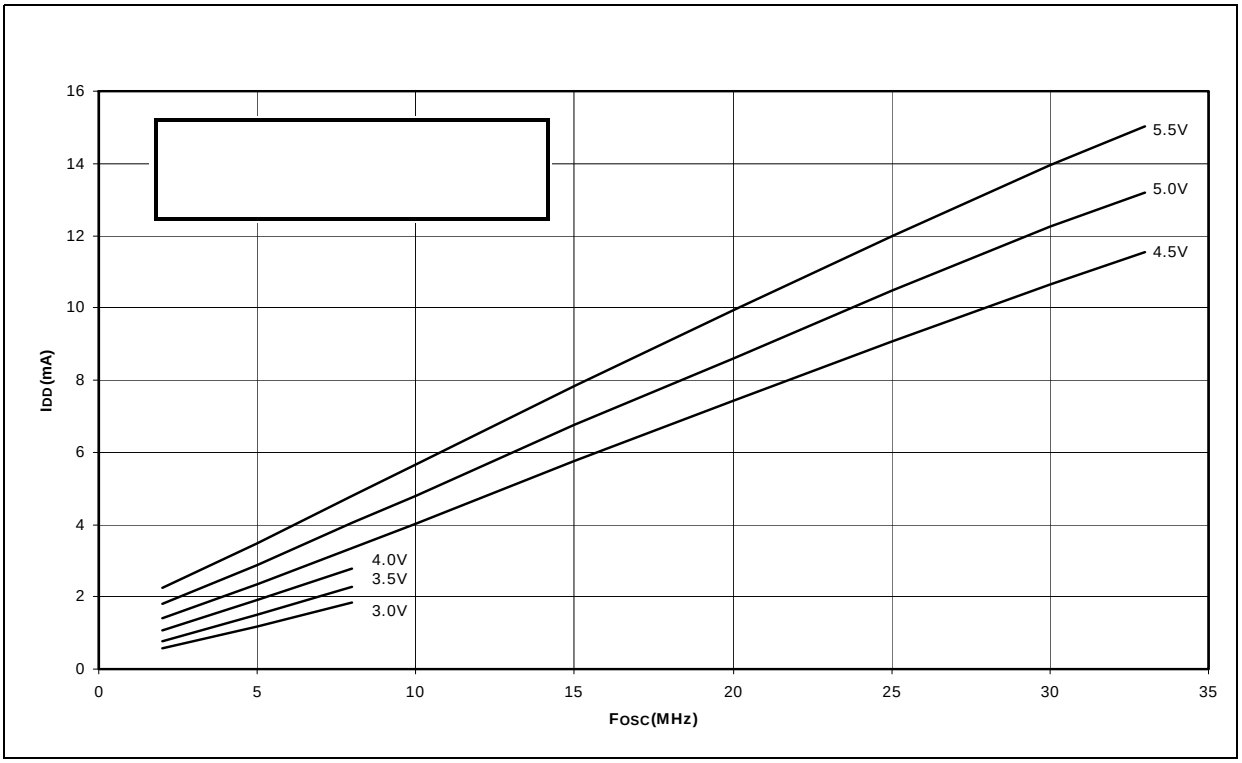


FIGURE 21-10: MAXIMUM IDD vs. FOSC OVER VDD (XT MODE)

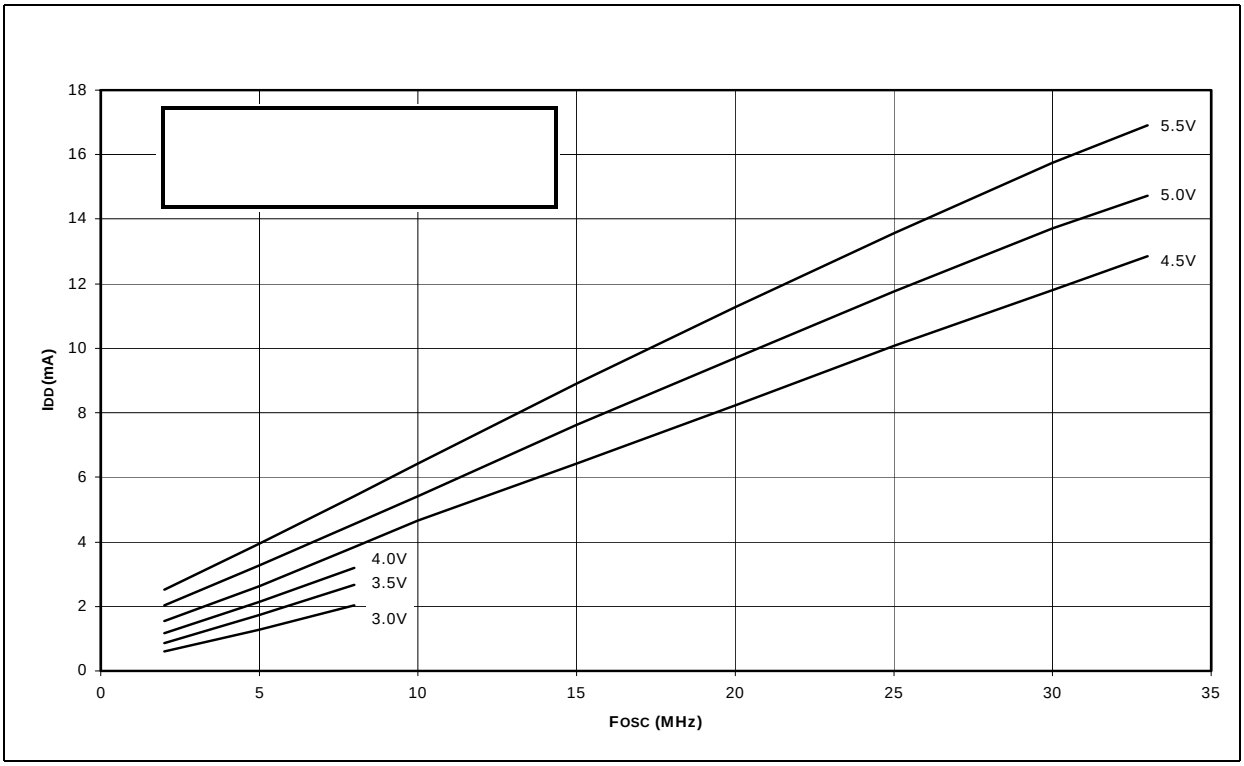
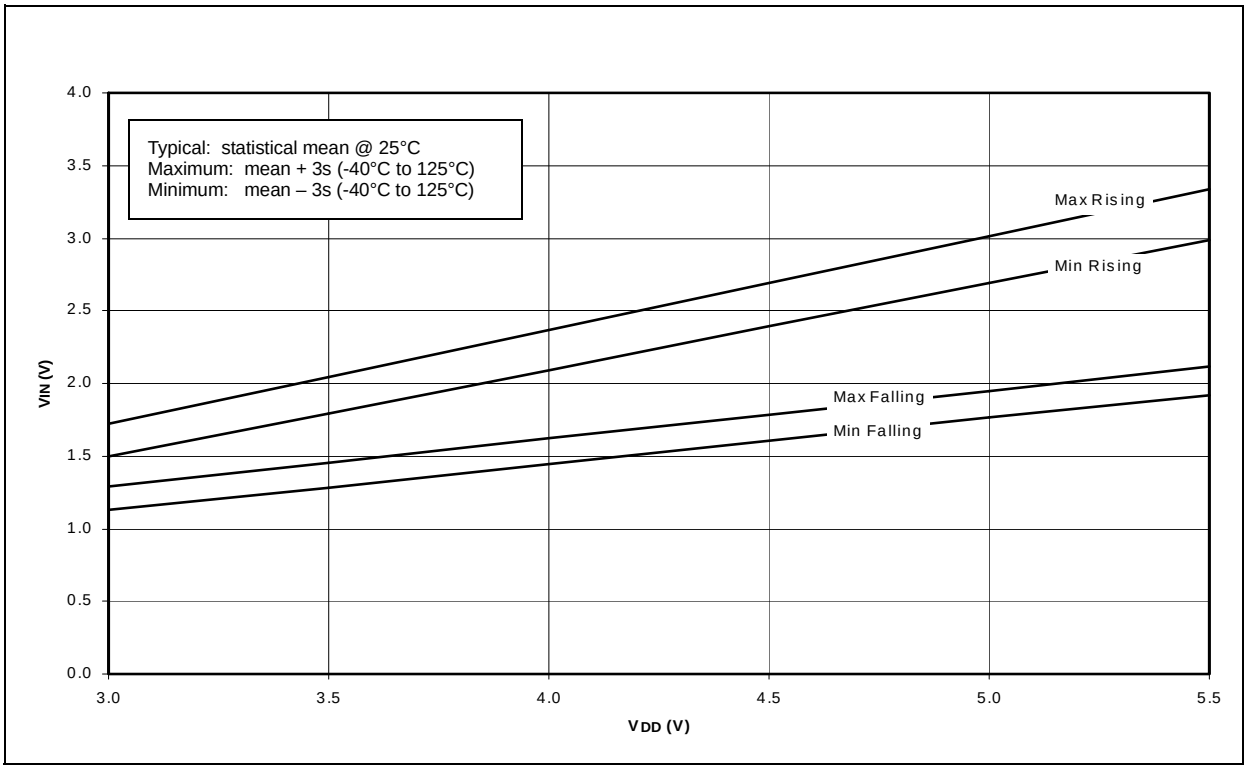
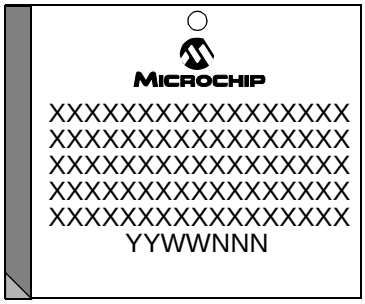


FIGURE 21-23: MAXIMUM AND MINIMUM VIN vs. VDD (I^2C Input, -40°C to +125°C)

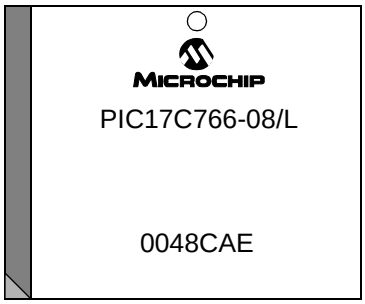


Package Marking Information (Cont.)

84-Lead PLCC



Example



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 Total Pages Sent

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?