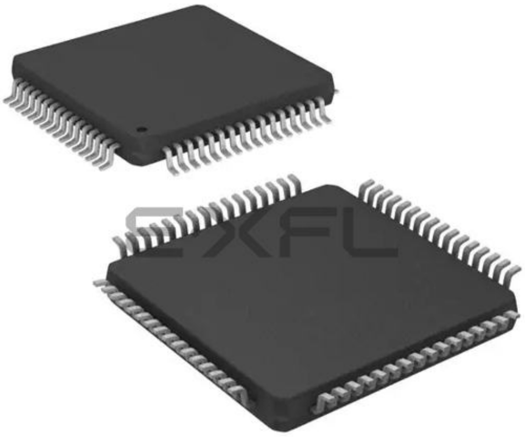
Microchip Technology - PIC17C752T-33I/PT Datasheet



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	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	-40°C ~ 85°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-33i-pt

2.0 DEVICE VARIETIES

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

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

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

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

TABLE 2-1: DEVICE MEMORY VARIETIES

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

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

2.1 UV Erasable Devices

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

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

2.2 One-Time-Programmable (OTP) Devices

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

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

2.3 Quick-Turnaround-Production (QTP) Devices

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

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

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

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

2.5 Read Only Memory (ROM) Devices

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

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

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

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

3.0 ARCHITECTURAL OVERVIEW

The high performance of the PIC17CXXX can be attributed to a number of architectural features, commonly found in RISC microprocessors. To begin with, the PIC17CXXX uses a modified Harvard architecture. This architecture has the program and data accessed from separate memories. So, the device has a program memory bus and a data memory bus. This improves bandwidth over traditional von Neumann architecture, where program and data are fetched from the same memory (accesses over the same bus). Separating program and data memory further allows instructions to be sized differently than the 8-bit wide data word. PIC17CXXX opcodes are 16-bits wide, enabling single word instructions. The full 16-bit wide program memory bus fetches a 16-bit instruction in a single cycle. A two-stage pipeline overlaps fetch and execution of instructions. Consequently, all instructions execute in a single cycle (121 ns @ 33 MHz), except for program branches and two special instructions that transfer data between program and data memory.

The PIC17CXXX can address up to 64K x 16 of program memory space.

The **PIC17C752** and **PIC17C762** integrate 8K x 16 of EPROM program memory on-chip.

The **PIC17C756A** and **PIC17C766** integrate 16K x 16 EPROM program memory on-chip.

A simplified block diagram is shown in Figure 3-1. The descriptions of the device pins are listed in Table 3-1.

Program execution can be internal only (Microcontroller or Protected Microcontroller mode), external only (Microprocessor mode), or both (Extended Microcontroller mode). Extended Microcontroller mode does not allow code protection.

The PIC17CXXX can directly or indirectly address its register files or data memory. All special function registers, including the Program Counter (PC) and Working Register (WREG), are mapped in data memory. The PIC17CXXX has an orthogonal (symmetrical) instruction set that makes it possible to carry out any operation on any register using any addressing mode. This symmetrical nature and lack of 'special optimal situations' make programming with the PIC17CXXX simple, yet efficient. In addition, the learning curve is reduced significantly.

One of the PIC17CXXX family architectural enhancements from the PIC16CXX family, allows two file registers to be used in some two operand instructions. This allows data to be moved directly between two registers without going through the WREG register, thus increasing performance and decreasing program memory usage.

The PIC17CXXX devices contain an 8-bit ALU and working register. The ALU is a general purpose arithmetic unit. It performs arithmetic and Boolean functions between data in the working register and any register file.

The WREG register is an 8-bit working register used for ALU operations.

All PIC17CXXX devices have an 8 x 8 hardware multiplier. This multiplier generates a 16-bit result in a single cycle.

The ALU is 8-bits wide and capable of addition, subtraction, shift and logical operations. Unless otherwise mentioned, arithmetic operations are two's complement in nature.

Depending on the instruction executed, the ALU may affect the values of the Carry (C), Digit Carry (DC), Zero (Z) and Overflow (OV) bits in the ALUSTA register. The C and DC bits operate as a $\overline{\text{borrow}}$ and $\overline{\text{digit borrow}}$ out bit, respectively, in subtraction. See the `SUBLW` and `SUBWF` instructions for examples.

Signed arithmetic is comprised of a magnitude and a sign bit. The overflow bit indicates if the magnitude overflows and causes the sign bit to change state. That is, if the result of 8-bit signed operations is greater than 127 (7Fh), or less than -128 (80h).

Signed math can have greater than 7-bit values (magnitude), if more than one byte is used. The overflow bit only operates on bit6 (MSb of magnitude) and bit7 (sign bit) of each byte value in the ALU. That is, the overflow bit is not useful if trying to implement signed math where the magnitude, for example, is 11-bits.

If the signed math values are greater than 7-bits (such as 15-, 24-, or 31-bit), the algorithm must ensure that the low order bytes of the signed value ignore the overflow status bit.

Example 3-1 shows two cases of doing signed arithmetic. The Carry (C) bit and the Overflow (OV) bit are the most important status bits for signed math operations.

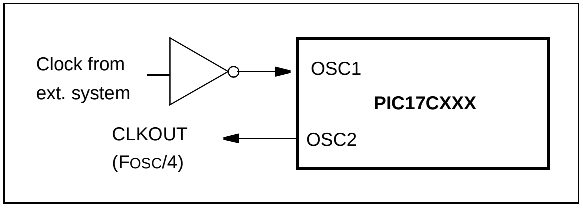
EXAMPLE 3-1: 8-BIT MATH ADDITION

Hex Value	Signed Values	Unsigned Values
FFh	-1	255
+ 01h	+ 1	+ 1
= 00h	= 0 (FEh)	= 256 → 00h
C bit = 1	C bit = 1	C bit = 1
OV bit = 0	OV bit = 0	OV bit = 0
DC bit = 1	DC bit = 1	DC bit = 1
Z bit = 1	Z bit = 1	Z bit = 1
Hex Value	**Signed Values**	**Unsigned Values**
7Fh	127	127
+ 01h	+ 1	+ 1
= 80h	= 128 → 00h	= 128
C bit = 0	C bit = 0	C bit = 0
OV bit = 1	OV bit = 1	OV bit = 1
DC bit = 1	DC bit = 1	DC bit = 1
Z bit = 0	Z bit = 0	Z bit = 0

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

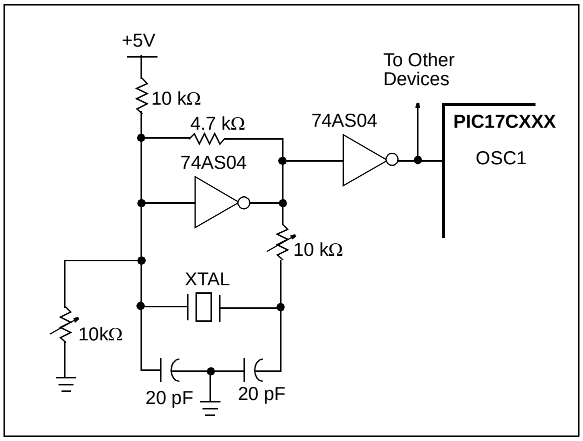
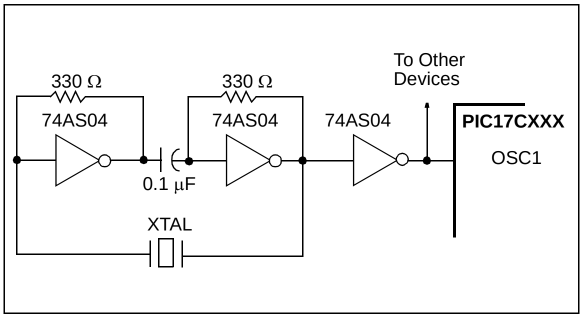


Figure 4-6 shows a series resonant oscillator circuit. This circuit is also designed to use the fundamental frequency of the crystal. The inverter performs a 180-degree phase shift in a series resonant oscillator circuit. The 330 Ω resistors provide the negative feedback to bias the inverters in their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

FIGURE 4-6: EXTERNAL SERIES RESONANT CRYSTAL OSCILLATOR CIRCUIT



NOTES:

5.1.4 TIME-OUT SEQUENCE

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

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

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

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

TABLE 5-1: TIME-OUT IN VARIOUS SITUATIONS

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

TABLE 5-2: STATUS BITS AND THEIR SIGNIFICANCE

$\overline{\text{POR}}$	$\overline{\text{BOR}}$[1]	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Event
0	0	1	1	Power-on Reset
1	1	1	0	$\overline{\text{MCLR}}$ Reset during SLEEP or interrupt wake-up from SLEEP
1	1	0	1	WDT Reset during normal operation
1	1	0	0	WDT Wake-up during SLEEP
1	1	1	1	$\overline{\text{MCLR}}$ Reset during normal operation
1	0	1	1	Brown-out Reset
0	0	0	x	Illegal, $\overline{\text{TO}}$ is set on $\overline{\text{POR}}$
0	0	x	0	Illegal, $\overline{\text{PD}}$ is set on $\overline{\text{POR}}$
x	x	1	1	CLRWDT instruction executed

Note 1: When BODEN is enabled, else the $\overline{\text{BOR}}$ status bit is unknown.

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

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

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

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

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

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

4: When BODEN is enabled, else the $\overline{\text{BOR}}$ status bit is unknown.

TABLE 5-4: INITIALIZATION CONDITIONS FOR SPECIAL FUNCTION REGISTERS

Register	Address	Power-on Reset Brown-out Reset	$\overline{\text{MCLR}}$ Reset WDT Reset	Wake-up from SLEEP through Interrupt
Unbanked				
INDF0	00h	N/A	N/A	N/A
FSR0	01h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCL	02h	0000h	0000h	PC + 1(2)
PCLATH	03h	0000 0000	uuuu uuuu	uuuu uuuu
ALUSTA	04h	1111 xxxx	1111 uuuu	1111 uuuu
T0STA	05h	0000 000-	0000 000-	0000 000-
CPUSTA(3)	06h	--11 11qq	--11 qquu	--uu qquu
INTSTA	07h	0000 0000	0000 0000	uuuu uuuu(1)
INDF1	08h	N/A	N/A	N/A
FSR1	09h	xxxx xxxx	uuuu uuuu	uuuu uuuu
WREG	0Ah	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR0L	0Bh	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR0H	0Ch	xxxx xxxx	uuuu uuuu	uuuu uuuu
TBLPTRL	0Dh	0000 0000	0000 0000	uuuu uuuu
TBLPTRH	0Eh	0000 0000	0000 0000	uuuu uuuu
BSR	0Fh	0000 0000	0000 0000	uuuu uuuu
Bank 0				
PORTA(4,6)	10h	0-xx 11xx	0-uu 11uu	u-uu uuuu
DDRB	11h	1111 1111	1111 1111	uuuu uuuu
PORTB(4)	12h	xxxx xxxx	uuuu uuuu	uuuu uuuu
RCSTA1	13h	0000 -00x	0000 -00u	uuuu -uuu
RCREG1	14h	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXSTA1	15h	0000 --1x	0000 --1u	uuuu --uu
TXREG1	16h	xxxx xxxx	uuuu uuuu	uuuu uuuu
SPBRG1	17h	0000 0000	0000 0000	uuuu uuuu

Legend: u = unchanged, x = unknown, - = unimplemented, read as '0', q = value depends on condition

Note 1: One or more bits in INTSTA, PIR1, PIR2 will be affected (to cause wake-up).

2: When the wake-up is due to an interrupt and the GLINTD bit is cleared, the PC is loaded with the interrupt vector.

3: See Table 5-3 for RESET value of specific condition.

4: This is the value that will be in the port output latch.

5: When the device is configured for Microprocessor or Extended Microcontroller mode, the operation of this port does not rely on these registers.

6: On any device RESET, these pins are configured as inputs.

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

REGISTER 13-3: TCON3 REGISTER (ADDRESS: 16h, BANK 7)

U-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	CA4OVF	CA3OVF	CA4ED1	CA4ED0	CA3ED1	CA3ED0	PWM3ON
bit 7							bit 0

bit 7 **Unimplemented:** Read as ‘0’

bit 6 **CA4OVF**: Capture4 Overflow Status bit
This bit indicates that the capture value had not been read from the capture register pair (CA4H:CA4L) before the next capture event occurred. The capture register retains the oldest unread capture value (last capture before overflow). Subsequent capture events will not update the capture register with the TMR3 value until the capture register has been read (both bytes).
`1` = Overflow occurred on Capture4 registers
`0` = No overflow occurred on Capture4 registers

bit 5 **CA3OVF**: Capture3 Overflow Status bit
This bit indicates that the capture value had not been read from the capture register pair (CA3H:CA3L) before the next capture event occurred. The capture register retains the oldest unread capture value (last capture before overflow). Subsequent capture events will not update the capture register with the TMR3 value until the capture register has been read (both bytes).
`1` = Overflow occurred on Capture3 registers
`0` = No overflow occurred on Capture3 registers

bit 4-3 **CA4ED1:CA4ED0**: Capture4 Mode Select bits
`00` = Capture on every falling edge
`01` = Capture on every rising edge
`10` = Capture on every 4th rising edge
`11` = Capture on every 16th rising edge

bit 2-1 **CA3ED1:CA3ED0**: Capture3 Mode Select bits
`00` = Capture on every falling edge
`01` = Capture on every rising edge
`10` = Capture on every 4th rising edge
`11` = Capture on every 16th rising edge

bit 0 **PWM3ON**: PWM3 On bit
`1` = PWM3 is enabled (the RG5/PWM3 pin ignores the state of the DDRG<5> bit)
`0` = PWM3 is disabled (the RG5/PWM3 pin uses the state of the DDRG<5> bit for data direction)

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

13.2 Timer3

Timer3 is a 16-bit timer consisting of the TMR3H and TMR3L registers. TMR3H is the high byte of the timer and TMR3L is the low byte. This timer has an associated 16-bit period register (PR3H/CA1H:PR3L/CA1L). This period register can be software configured to be a another 16-bit capture register.

When the TMR3CS bit (TCON1<2>) is clear, the timer increments every instruction cycle (FOSC/4). When TMR3CS is set, the counter increments on every falling edge of the RB5/TCLK3 pin. In either mode, the TMR3ON bit must be set for the timer/counter to increment. When TMR3ON is clear, the timer will not increment or set flag bit TMR3IF.

Timer3 has two modes of operation, depending on the CA1/$\overline{\text{PR3}}$ bit (TCON2<3>). These modes are:

- Three capture and one period register mode
- Four capture register mode

The PIC17C7XX has up to four 16-bit capture registers that capture the 16-bit value of TMR3 when events are detected on capture pins. There are four capture pins (RB0/CAP1, RB1/CAP2, RG4/CAP3, and RE3/CAP4), one for each capture register pair. The capture pins are multiplexed with the I/O pins. An event can be:

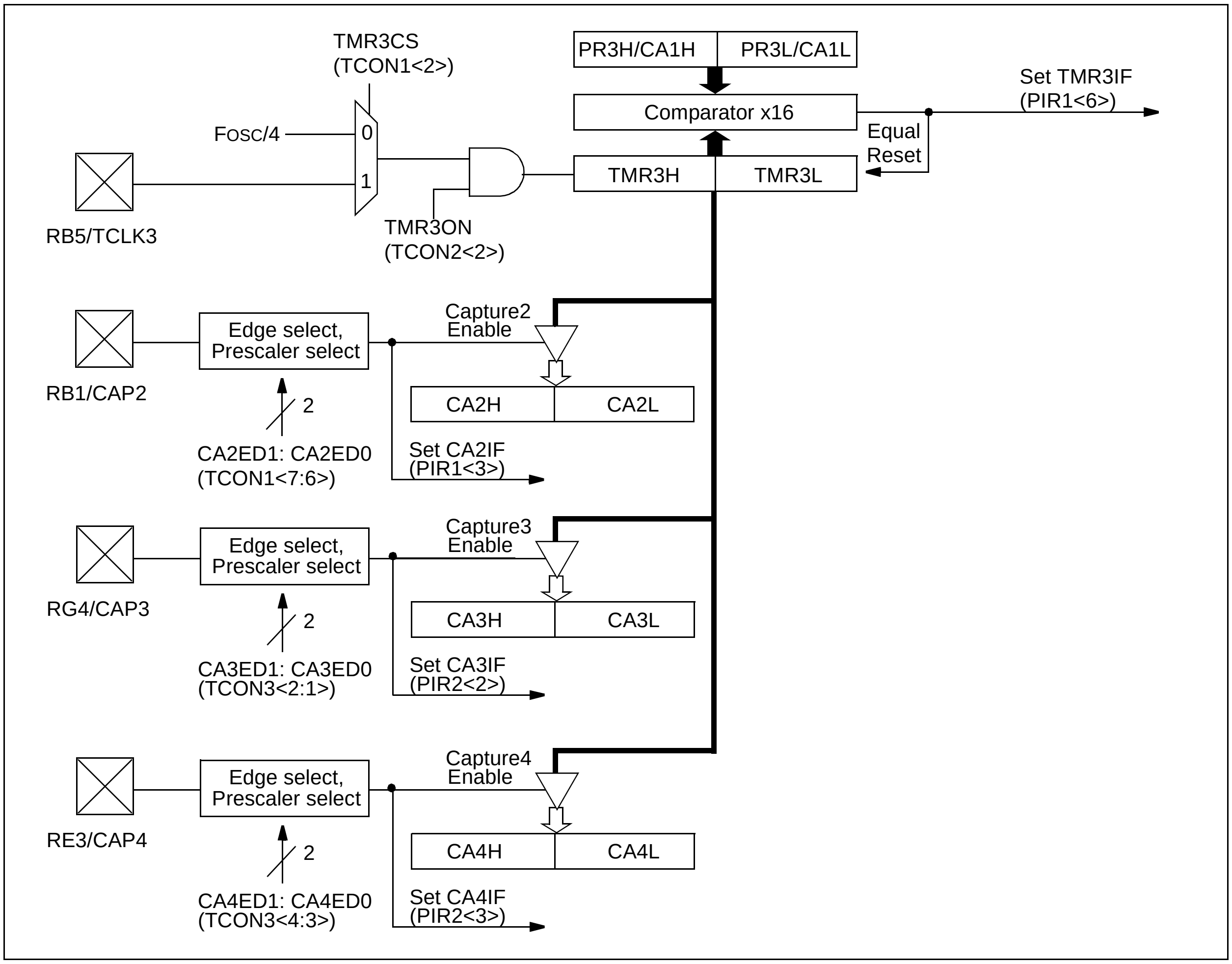
- A rising edge
- A falling edge
- Every 4th rising edge
- Every 16th rising edge

Each 16-bit capture register has an interrupt flag associated with it. The flag is set when a capture is made. The capture modules are truly part of the Timer3 block. Figure 13-5 and Figure 13-6 show the block diagrams for the two modes of operation.

13.2.1 THREE CAPTURE AND ONE PERIOD REGISTER MODE

In this mode, registers PR3H/CA1H and PR3L/CA1L constitute a 16-bit period register. A block diagram is shown in Figure 13-5. The timer increments until it equals the period register and then resets to 0000h on the next timer clock. TMR3 Interrupt Flag bit (TMR3IF) is set at this point. This interrupt can be disabled by clearing the TMR3 Interrupt Enable bit (TMR3IE). TMR3IF must be cleared in software.

FIGURE 13-5: TIMER3 WITH THREE CAPTURE AND ONE PERIOD REGISTER BLOCK DIAGRAM



15.2.6 MULTI-MASTER MODE

In Multi-Master mode, the interrupt generation on the detection of the START and STOP conditions allows the determination of when the bus is free. The STOP (P) and START (S) bits are cleared from a RESET, or when the MSSP module is disabled. Control of the I^{2}C bus may be taken when bit P (SSPSTAT<4>) is set, or the bus is idle, with both the S and P bits clear. When the bus is busy, enabling the SSP interrupt will generate the interrupt when the STOP condition occurs.

In Multi-Master operation, the SDA line must be monitored for arbitration, to see if the signal level is the expected output level. This check is performed in hardware, with the result placed in the BCLIF bit.

The states where arbitration can be lost are:

- Address Transfer
- Data Transfer
- A START Condition
- A Repeated Start Condition
- An Acknowledge Condition

15.2.7 I^{2}C MASTER MODE SUPPORT

Master mode is enabled by setting and clearing the appropriate SSPM bits in SSPCON1 and by setting the SSPEN bit. Once Master mode is enabled, the user has six options.

- Assert a START condition on SDA and SCL.
- Assert a Repeated Start condition on SDA and SCL.
- Write to the SSPBUF register initiating transmission of data/address.
- Generate a STOP condition on SDA and SCL.
- Configure the I^{2}C port to receive data.
- Generate an Acknowledge condition at the end of a received byte of data.

Note:	The MSSP Module, when configured in I^{2}C Master mode, does not allow queueing of events. For instance: The user is not allowed to initiate a START condition and immediately write the SSPBUF register to initiate transmission before the START condition is complete. In this case, the SSPBUF will not be written to and the WCOL bit will be set, indicating that a write to the SSPBUF did not occur.

15.2.7.1 I^{2}C Master Mode Operation

The master device generates all of the serial clock pulses and the START and STOP conditions. A transfer is ended with a STOP condition or with a Repeated Start condition. Since the Repeated Start condition is also the beginning of the next serial transfer, the I^{2}C bus will not be released.

In Master Transmitter mode, serial data is output through SDA, while SCL outputs the serial clock. The first byte transmitted contains the slave address of the receiving device (7 bits) and the Read/Write (R/$\overline{W}$) bit. In this case, the R/$\overline{W}$ bit will be logic '0'. Serial data is transmitted 8 bits at a time. After each byte is transmitted, an acknowledge bit is received. START and STOP conditions are output to indicate the beginning and the end of a serial transfer.

In Master Receive mode, the first byte transmitted contains the slave address of the transmitting device (7 bits) and the R/$\overline{W}$ bit. In this case, the R/$\overline{W}$ bit will be logic '1'. Thus, the first byte transmitted is a 7-bit slave address, followed by a '1' to indicate receive bit. Serial data is received via SDA, while SCL outputs the serial clock. Serial data is received 8 bits at a time. After each byte is received, an acknowledge bit is transmitted. START and STOP conditions indicate the beginning and end of transmission.

The baud rate generator used for SPI mode operation is now used to set the SCL clock frequency for either 100 kHz, 400 kHz, or 1 MHz I^{2}C operation. The baud rate generator reload value is contained in the lower 7 bits of the SSPADD register. The baud rate generator will automatically begin counting on a write to the SSPBUF. Once the given operation is complete (i.e., transmission of the last data bit is followed by ACK), the internal clock will automatically stop counting and the SCL pin will remain in its last state

15.3 Connection Considerations for I^2C Bus

For standard mode I^2C bus devices, the values of resistors ***R_p R_s*** in Figure 15-42 depends on the following parameters:

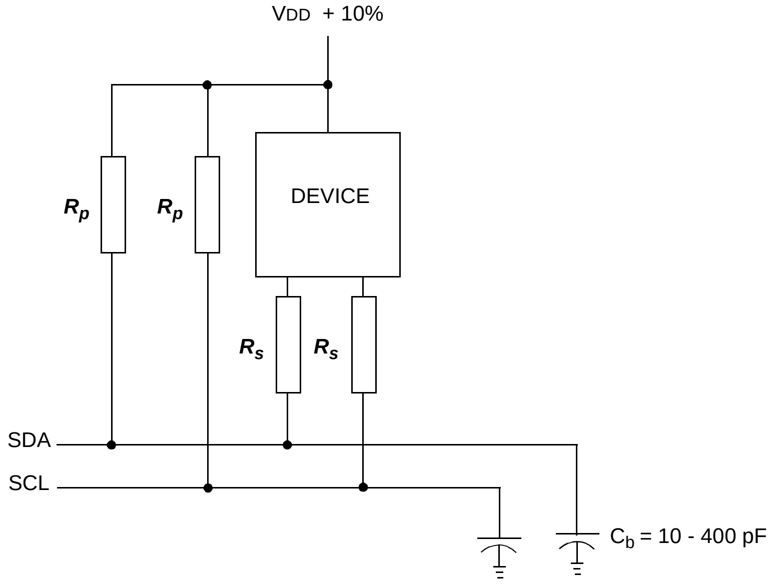
- Supply voltage
- Bus capacitance
- Number of connected devices (input current + leakage current)

The supply voltage limits the minimum value of resistor ***R_p*** due to the specified minimum sink current of 3 mA at VOL max = 0.4V for the specified output stages. For example, with a supply voltage of VDD = 5V ±10% and VOL max = 0.4V at 3 mA, ***R_p*** min = (5.5-0.4)/0.003 = 1.7 kΩ. VDD as a function of ***R_p*** is shown in Figure 15-42. The desired noise margin of 0.1 VDD for the low level, limits the maximum value of ***R_s***. Series resistors are optional and used to improve ESD susceptibility.

The bus capacitance is the total capacitance of wire, connections and pins. This capacitance limits the maximum value of ***R_p*** due to the specified rise time (Figure 15-42).

The SMP bit is the slew rate control enabled bit. This bit is in the SSPSTAT register and controls the slew rate of the I/O pins when in I^2C mode (master or slave).

FIGURE 15-42: SAMPLE DEVICE CONFIGURATION FOR I^2C BUS



Note: I^2C devices with input levels related to VDD must have one common supply line to which the pull-up resistor is also connected.

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

```
// Writes the byte data to 24LC01B at the specified address
void ByteWrite(static unsigned char address, static unsigned char data)
{
    StartI2C();                      // Send start bit
    IdleI2C();                       // Wait for idle condition
    WriteI2C(CONTROL);               // Send control byte
    IdleI2C();                       // Wait for idle condition
    if (!SSPCON2bits.ACKSTAT)        // If 24LC01B ACKs
    {
        WriteI2C(address);           // Send control byte
        IdleI2C();                   // Wait for idle condition

        if (!SSPCON2bits.ACKSTAT)    // If 24LC01B ACKs
            WriteI2C(data);          // Send data
    }
    IdleI2C();                       // Wait for idle condition
    StopI2C();                       // Send stop bit
    IdleI2C();                       // Wait for idle condition
    return;
}


// Reads a byte of data from 24LC01B at the specified address
unsigned char ByteRead(static unsigned char address)
{
    StartI2C();                      // Send start bit
    IdleI2C();                       // Wait for idle condition
    WriteI2C(CONTROL);               // Send control byte
    IdleI2C();                       // Wait for idle condition
    if (!SSPCON2bits.ACKSTAT)        // If the 24LC01B ACKs
    {
        WriteI2C(address);           // Send address
        IdleI2C();                   // Wait for idle condition
        if (!SSPCON2bits.ACKSTAT)    // If the 24LC01B ACKs
        {
            RestartI2C();            // Send restart
            IdleI2C();               // Wait for idle condition
            WriteI2C(CONTROL+1);     // Send control byte with R/W set
            IdleI2C();               // Wait for idle condition
            if (!SSPCON2bits.ACKSTAT)    // If the 24LC01B ACKs
            {
                getcI2C();               // Read a byte of data from 24LC01B
                IdleI2C();               // Wait for idle condition
                NotAckI2C();             // Send a NACK to 24LC01B
                IdleI2C();               // Wait for idle condition
                StopI2C();               // Send stop bit
                IdleI2C();               // Wait for idle condition
             }
        }
    }
    return(SSPBUF);
}
```

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

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

17.0 SPECIAL FEATURES OF THE CPU

What sets a microcontroller apart from other processors are special circuits to deal with the needs of real-time applications. The PIC17CXXX family has a host of such features intended to maximize system reliability, minimize cost through elimination of external components, provide power saving operating modes and offer code protection. These are:

- Oscillator Selection (Section 4.0)
- RESET (Section 5.0)
  - Power-on Reset (POR)
  - Power-up Timer (PWRT)
  - Oscillator Start-up Timer (OST)
  - Brown-out Reset (BOR)
- Interrupts (Section 6.0)
- Watchdog Timer (WDT)
- SLEEP mode
- Code protection

The PIC17CXXX has a Watchdog Timer which can be shut-off only through EPROM bits. It runs off its own RC oscillator for added reliability. There are two timers that offer necessary delays on POR and BOR. One is the Oscillator Start-up Timer (OST), intended to keep the chip in RESET until the crystal oscillator is stable. The other is the Power-up Timer (PWRT), which provides a fixed delay of 96 ms (nominal) on power-up only, designed to keep the part in RESET while the power supply stabilizes. With these two timers on-chip, most applications need no external RESET circuitry.

The SLEEP mode is designed to offer a very low current power-down mode. The user can wake from SLEEP through external RESET, Watchdog Timer Reset, or through an interrupt. Several oscillator options are also made available to allow the part to fit the application. The RC oscillator option saves system cost, while the LF crystal option saves power. Configuration bits are used to select various options. This configuration word has the format shown in Figure 17-1.

REGISTER 17-1: CONFIGURATION WORDS

High (H) Table Read Addr.	U-x	R/P-1	R/P-1	U-x	U-x	U-x	U-x	U-x	U-x
FE0Fh - FE08h	—	PM2	BODEN	—	—	—	—	—	—
	bit 15 bit 8	bit 7							bit 0

Low (L) Table Read Addr.	U-x	U-x	R/P-1	U-x	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
FE07h - FE00h	—	—	PM1	—	PM0	WDTPS1	WDTPS0	FOSC1	FOSC0
	bit 15 bit 8	bit 7							bit 0

bits 7H, 6L, 4L **PM2, PM1, PM0:** Processor Mode Select bits
`111` = Microprocessor mode
`110` = Microcontroller mode
`101` = Extended Microcontroller mode
`000` = Code Protected Microcontroller mode

bit 6H **BODEN:** Brown-out Detect Enable
`1` = Brown-out Detect circuitry is enabled
`0` = Brown-out Detect circuitry is disabled

bits 3L:2L **WDTPS1:WDTPS0**: WDT Postscaler Select bits
`11` = WDT enabled, postscaler = 1
`10` = WDT enabled, postscaler = 256
`01` = WDT enabled, postscaler = 64
`00` = WDT disabled, 16-bit overflow timer

bits 1L:0L **FOSC1:FOSC0**: Oscillator Select bits
`11` = EC oscillator
`10` = XT oscillator
`01` = RC oscillator
`00` = LF oscillator

Shaded bits (—) **Reserved**

MULLW Multiply Literal with WREG

Syntax:	[*label*] MULLW k
Operands:	0 ≤ k ≤ 255
Operation:	(k x WREG) → PRODH:PRODL
Status Affected:	None

Encoding:

`1011`	`1100`	`kkkk`	`kkkk`

Description: An unsigned multiplication is carried out between the contents of WREG and the 8-bit literal 'k'. The 16-bit result is placed in PRODH:PRODL register pair. PRODH contains the high byte.

WREG is unchanged.

None of the status flags are affected.

Note that neither overflow, nor carry is possible in this operation. A zero result is possible, but not detected.

Words:	1
Cycles:	1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process Data	Write registers PRODH: PRODL

Example: `MULLW 0xC4`

Before Instruction

WREG	=	0xE2
PRODH	=	?
PRODL	=	?

After Instruction

WREG	=	0xC4
PRODH	=	0xAD
PRODL	=	0x08

MULWF Multiply WREG with f

Syntax:	[*label*] MULWF f
Operands:	0 ≤ f ≤ 255
Operation:	(WREG x f) → PRODH:PRODL
Status Affected:	None

Encoding:

`0011`	`0100`	`ffff`	`ffff`

Description: An unsigned multiplication is carried out between the contents of WREG and the register file location 'f'. The 16-bit result is stored in the PRODH:PRODL register pair. PRODH contains the high byte.

Both WREG and 'f' are unchanged.

None of the status flags are affected.

Note that neither overflow, nor carry is possible in this operation. A zero result is possible, but not detected.

Words:	1
Cycles:	1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write registers PRODH: PRODL

Example: `MULWF REG`

Before Instruction

WREG	=	0xC4
REG	=	0xB5
PRODH	=	?
PRODL	=	?

After Instruction

WREG	=	0xC4
REG	=	0xB5
PRODH	=	0x8A
PRODL	=	0x94

FIGURE 20-3: PIC17LC7XX-08 VOLTAGE-FREQUENCY GRAPH

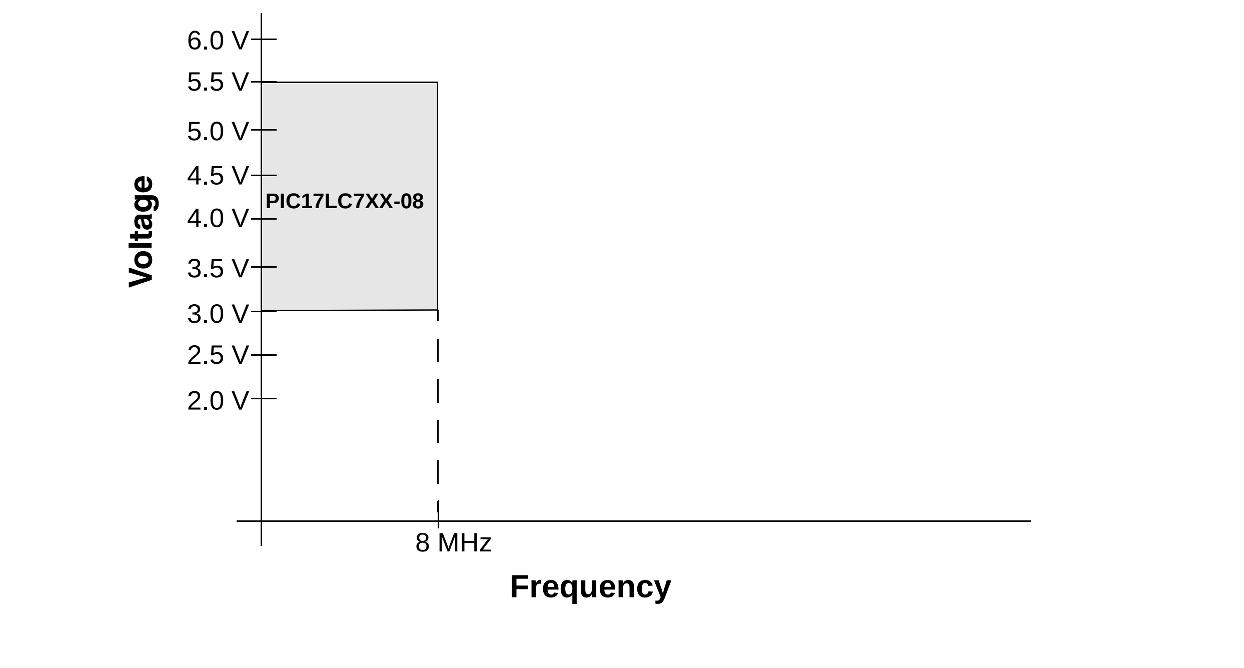
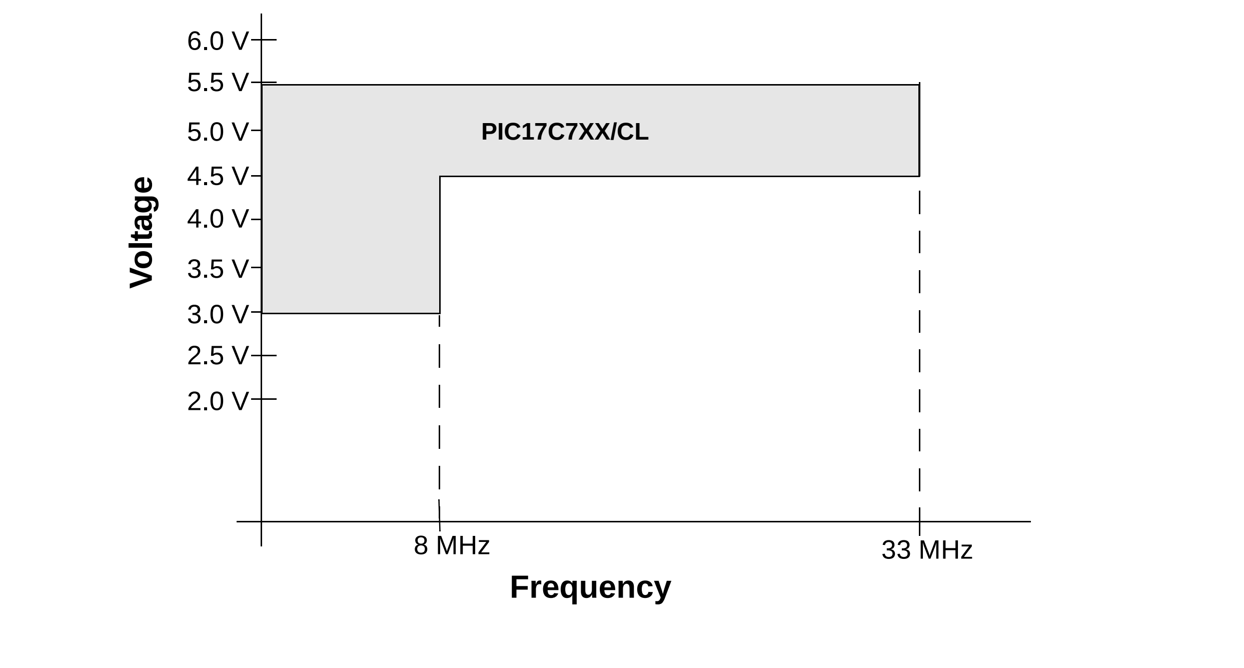


FIGURE 20-4: PIC17C7XX/CL VOLTAGE-FREQUENCY GRAPH



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

† 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 20-21: USART ASYNCHRONOUS MODE START BIT DETECT

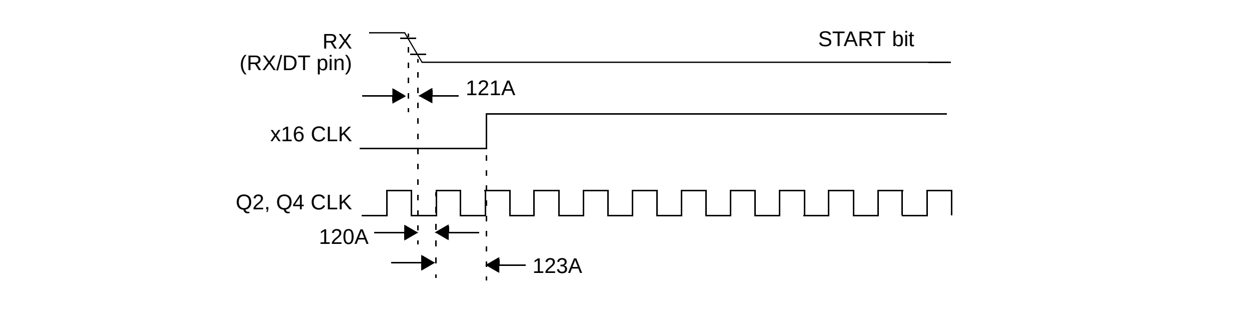


TABLE 20-16: USART ASYNCHRONOUS MODE START BIT DETECT REQUIREMENTS

Param No.	Sym	Characteristic		Min	Typ	Max	Units	Conditions
120A	TdtL2ckH	Time to ensure that the RX pin is sampled low		—	—	TCY	ns	
121A	TdtRF	Data rise time and fall time	Receive	—	—	**(Note 1)**	ns	
			Transmit	—	—	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

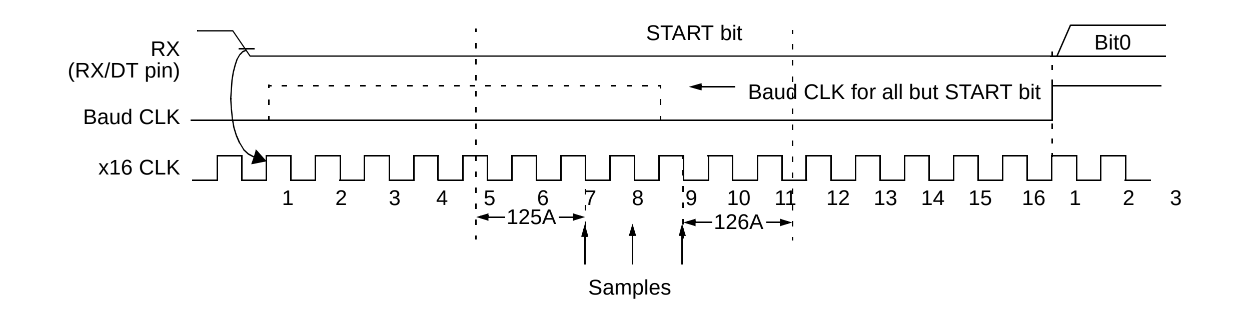


TABLE 20-17: USART ASYNCHRONOUS RECEIVE SAMPLING REQUIREMENTS

Param No.	Sym	Characteristic	Min	Typ	Max	Units	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	

INDEX