Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

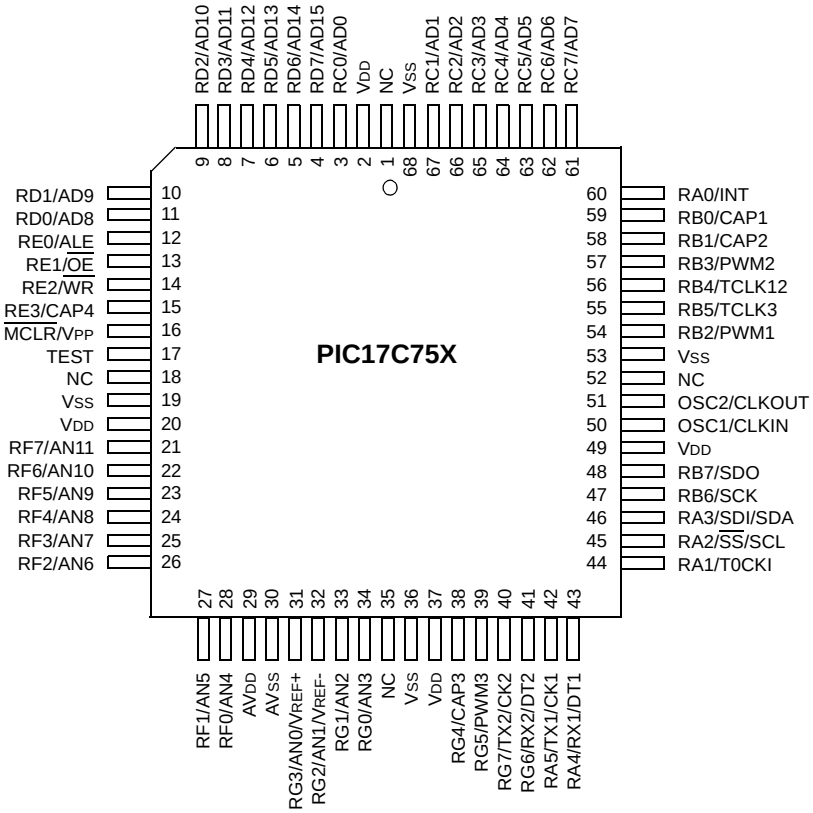
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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	16MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	66
Program Memory Size	16KB (8K x 16)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	678 x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 16x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	84-LCC (J-Lead)
Supplier Device Package	84-PLCC (29.31x29.31)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic17c762t-16i-l

Pin Diagrams cont.'d

68-Pin PLCC



64-Pin TQFP

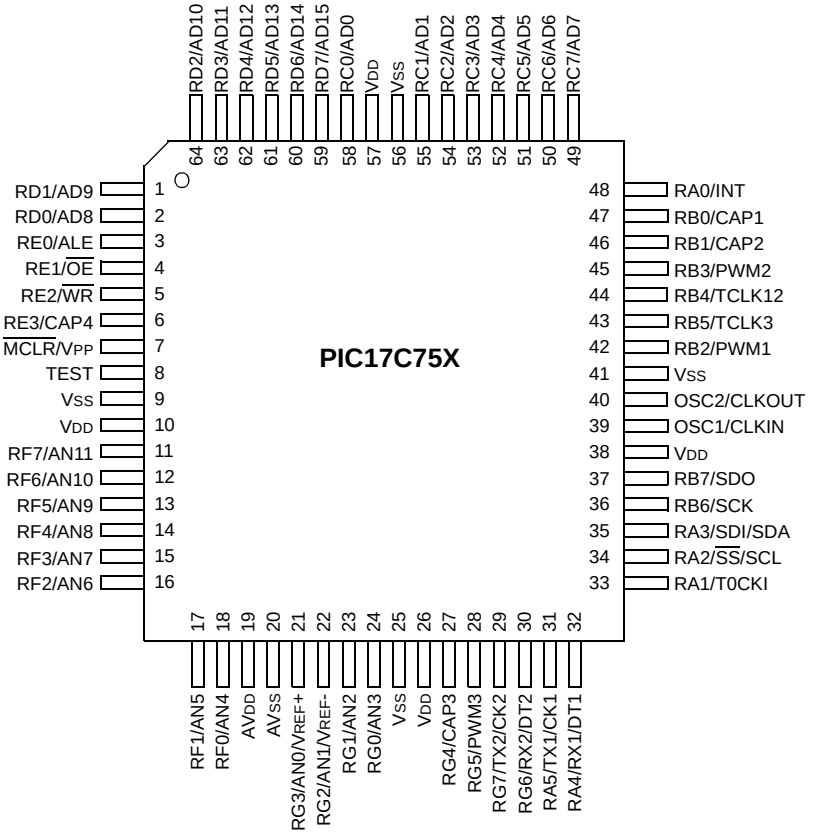


FIGURE 3-1: PIC17C752/756A BLOCK DIAGRAM

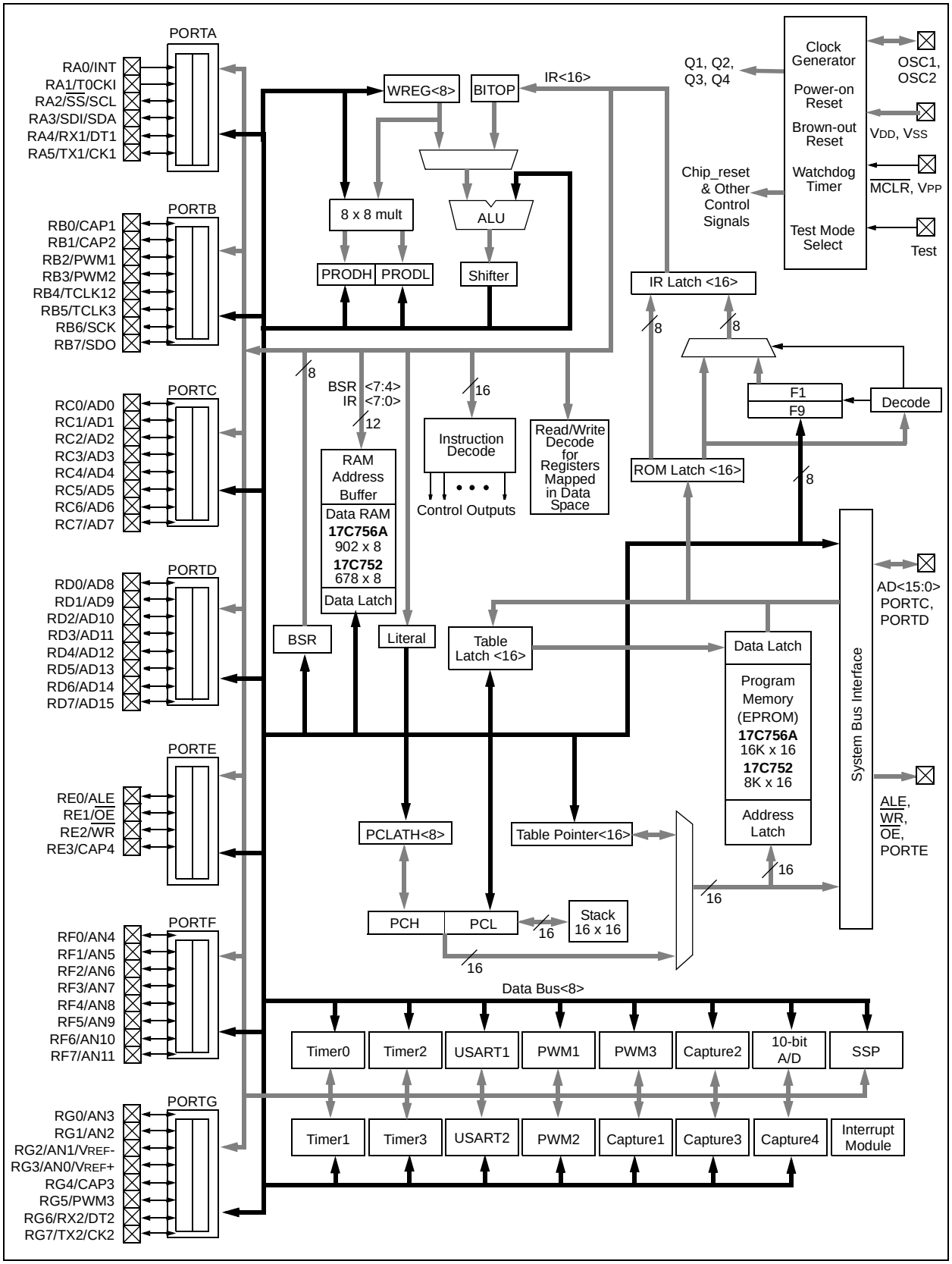


TABLE 3-1: PINOUT DESCRIPTIONS

Name	PIC17C75X			PIC17C76X		I/O/P Type	Buffer Type	Description
	DIP No.	PLCC No.	TQFP No.	PLCC No.	QFP No.			
OSC1/CLKIN	47	50	39	62	49	I	ST	Oscillator input in Crystal/Resonator or RC Oscillator mode. External clock input in External Clock mode.
OSC2/CLKOUT	48	51	40	63	50	O	—	Oscillator output. Connects to crystal or resonator in Crystal Oscillator mode. In RC Oscillator or External Clock modes, OSC2 pin outputs CLKOUT which has one fourth the frequency (FOSC/4) of OSC1 and denotes the instruction cycle rate.
$\overline{MCLR}$/VPP	15	16	7	20	9	I/P	ST	Master clear (RESET) input or Programming Voltage (VPP) input. This is the active low RESET input to the device.
								PORTA pins have individual differentiations that are listed in the following descriptions:
RA0/INT	56	60	48	72	58	I	ST	RA0 can also be selected as an external interrupt input. Interrupt can be configured to be on positive or negative edge. Input only pin.
RA1/T0CKI	41	44	33	56	43	I	ST	RA1 can also be selected as an external interrupt input and the interrupt can be configured to be on positive or negative edge. RA1 can also be selected to be the clock input to the Timer0 timer/counter. Input only pin.
RA2/$\overline{SS}$/SCL	42	45	34	57	44	I/O(2)	ST	RA2 can also be used as the slave select input for the SPI or the clock input for the I^2C bus. High voltage, high current, open drain port pin.
RA3/SDI/SDA	43	46	35	58	45	I/O(2)	ST	RA3 can also be used as the data input for the SPI or the data for the I^2C bus. High voltage, high current, open drain port pin.
RA4/RX1/DT1	40	43	32	51	38	I/O(1)	ST	RA4 can also be selected as the USART1 (SCI) Asynchronous Receive or USART1 (SCI) Synchronous Data. Output available from USART only.
RA5/TX1/CK1	39	42	31	50	37	I/O(1)	ST	RA5 can also be selected as the USART1 (SCI) Asynchronous Transmit or USART1 (SCI) Synchronous Clock. Output available from USART only.
								PORTB is a bi-directional I/O Port with software configurable weak pull-ups.
RB0/CAP1	55	59	47	71	57	I/O	ST	RB0 can also be the Capture1 input pin.
RB1/CAP2	54	58	46	70	56	I/O	ST	RB1 can also be the Capture2 input pin.
RB2/PWM1	50	54	42	66	52	I/O	ST	RB2 can also be the PWM1 output pin.
RB3/PWM2	53	57	45	69	55	I/O	ST	RB3 can also be the PWM2 output pin.
RB4/TCLK12	52	56	44	68	54	I/O	ST	RB4 can also be the external clock input to Timer1 and Timer2.
RB5/TCLK3	51	55	43	67	53	I/O	ST	RB5 can also be the external clock input to Timer3.
RB6/SCK	44	47	36	59	46	I/O	ST	RB6 can also be used as the master/slave clock for the SPI.
RB7/SDO	45	48	37	60	47	I/O	ST	RB7 can also be used as the data output for the SPI.

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.

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.			
								PORTG is a bi-directional I/O Port.
RG0/AN3	32	34	24	42	30	I/O	ST	RG0 can also be analog input 3.
RG1/AN2	31	33	23	41	29	I/O	ST	RG1 can also be analog input 2.
RG2/AN1/VREF-	30	32	22	40	28	I/O	ST	RG2 can also be analog input 1, or the ground reference voltage.
RG3/AN0/VREF+	29	31	21	39	27	I/O	ST	RG3 can also be analog input 0, or the positive reference voltage.
RG4/CAP3	35	38	27	46	33	I/O	ST	RG4 can also be the Capture3 input pin.
RG5/PWM3	36	39	28	47	34	I/O	ST	RG5 can also be the PWM3 output pin.
RG6/RX2/DT2	38	41	30	49	36	I/O	ST	RG6 can also be selected as the USART2 (SCI) Asynchronous Receive or USART2 (SCI) Synchronous Data.
RG7/TX2/CK2	37	40	29	48	35	I/O	ST	RG7 can also be selected as the USART2 (SCI) Asynchronous Transmit or USART2 (SCI) Synchronous Clock.
								PORTH is a bi-directional I/O Port. PORTH is only available on the PIC17C76X devices.
RH0	—	—	—	10	79	I/O	ST	
RH1	—	—	—	11	80	I/O	ST	
RH2	—	—	—	12	1	I/O	ST	
RH3	—	—	—	13	2	I/O	ST	
RH4/AN12	—	—	—	31	19	I/O	ST	RH4 can also be analog input 12.
RH5/AN13	—	—	—	32	20	I/O	ST	RH5 can also be analog input 13.
RH6/AN14	—	—	—	33	21	I/O	ST	RH6 can also be analog input 14.
RH7/AN15	—	—	—	34	22	I/O	ST	RH7 can also be analog input 15.
								PORTJ is a bi-directional I/O Port. PORTJ is only available on the PIC17C76X devices.
RJ0	—	—	—	52	39	I/O	ST	
RJ1	—	—	—	53	40	I/O	ST	
RJ2	—	—	—	54	41	I/O	ST	
RJ3	—	—	—	55	42	I/O	ST	
RJ4	—	—	—	73	59	I/O	ST	
RJ5	—	—	—	74	60	I/O	ST	
RJ6	—	—	—	75	61	I/O	ST	
RJ7	—	—	—	76	62	I/O	ST	
TEST	16	17	8	21	10	I	ST	Test mode selection control input. Always tie to VSS for normal operation.
VSS	17, 33, 49, 64	19, 36, 53, 68	9, 25, 41, 56	23, 44, 65, 84	11, 31, 51, 70	P		Ground reference for logic and I/O pins.
VDD	1, 18, 34, 46	2, 20, 37, 49,	10, 26, 38, 57	24, 45, 61, 2	12, 32, 48, 71	P		Positive supply for logic and I/O pins.
AVSS	28	30	20	38	26	P		Ground reference for A/D converter. This pin **MUST** be at the same potential as VSS.
AVDD	27	29	19	37	25	P		Positive supply for A/D converter. This pin **MUST** be at the same potential as VDD.
NC	—	1, 18, 35, 52	—	1, 22, 43, 64	—			No Connect. Leave these pins unconnected.

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.

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.

6.4 Interrupt Operation

Global Interrupt Disable bit, GLINTD (CPUSTA<4>), enables all unmasked interrupts (if clear), or disables all interrupts (if set). Individual interrupts can be disabled through their corresponding enable bits in the INTSTA register. Peripheral interrupts need either the global peripheral enable PEIE bit disabled, or the specific peripheral enable bit disabled. Disabling the peripherals via the global peripheral enable bit, disables all peripheral interrupts. GLINTD is set on RESET (interrupts disabled).

The `RETFIE` instruction clears the GLINTD bit while forcing the Program Counter (PC) to the value loaded at the Top-of-Stack.

When an interrupt is responded to, the GLINTD bit is automatically set to disable any further interrupt, the return address is pushed onto the stack and the PC is loaded with the interrupt vector. There are four interrupt vectors which help reduce interrupt latency.

The peripheral interrupt vector has multiple interrupt sources. Once in the peripheral Interrupt Service Routine, the source(s) of the interrupt can be determined by polling the interrupt flag bits. The peripheral interrupt flag bit(s) must be cleared in software before re-enabling interrupts to avoid continuous interrupts.

The PIC17C7XX devices have four interrupt vectors. These vectors and their hardware priority are shown in Table 6-1. If two enabled interrupts occur “at the same time”, the interrupt of the highest priority will be serviced first. This means that the vector address of that interrupt will be loaded into the program counter (PC).

TABLE 6-1: INTERRUPT VECTORS/ PRIORITIES

Address	Vector	Priority
0008h	External Interrupt on RA0/ INT pin (INTF)	1 (Highest)
0010h	TMR0 Overflow Interrupt (T0IF)	2
0018h	External Interrupt on T0CKI (T0CKIF)	3
0020h	Peripherals (PEIF)	4 (Lowest)

Note 1: Individual interrupt flag bits are set, regardless of the status of their corresponding mask bit or the GLINTD bit.

2: Before disabling any of the INTSTA enable bits, the GLINTD bit should be set (disabled).

6.5 RA0/INT Interrupt

The external interrupt on the RA0/INT pin is edge triggered. Either the rising edge if the INTEDG bit (T0STA<7>) is set, or the falling edge if the INTEDG bit is clear. When a valid edge appears on the RA0/INT pin, the INTF bit (INTSTA<4>) is set. This interrupt can be disabled by clearing the INTE control bit (INTSTA<0>). The INT interrupt can wake the processor from SLEEP. See Section 17.4 for details on SLEEP operation.

6.6 T0CKI Interrupt

The external interrupt on the RA1/T0CKI pin is edge triggered. Either the rising edge if the T0SE bit (T0STA<6>) is set, or the falling edge if the T0SE bit is clear. When a valid edge appears on the RA1/T0CKI pin, the T0CKIF bit (INTSTA<6>) is set. This interrupt can be disabled by clearing the T0CKIE control bit (INTSTA<2>). The T0CKI interrupt can wake up the processor from SLEEP. See Section 17.4 for details on SLEEP operation.

6.7 Peripheral Interrupt

The peripheral interrupt flag indicates that at least one of the peripheral interrupts occurred (PEIF is set). The PEIF bit is a read only bit and is a bit wise OR of all the flag bits in the PIR registers AND’d with the corresponding enable bits in the PIE registers. Some of the peripheral interrupts can wake the processor from SLEEP. See Section 17.4 for details on SLEEP operation.

6.8 Context Saving During Interrupts

During an interrupt, only the returned PC value is saved on the stack. Typically, users may wish to save key registers during an interrupt; e.g. WREG, ALUSTA and the BSR registers. This requires implementation in software.

Example 6-2 shows the saving and restoring of information for an Interrupt Service Routine. This is for a simple interrupt scheme, where only one interrupt may occur at a time (no interrupt nesting). The SFRs are stored in the non-banked GPR area.

Example 6-2 shows the saving and restoring of information for a more complex Interrupt Service Routine. This is useful where nesting of interrupts is required. A maximum of 6 levels can be done by this example. The BSR is stored in the non-banked GPR area, while the other registers would be stored in a particular bank. Therefore, 6 saves may be done with this routine (since there are 6 non-banked GPR registers). These routines require a dedicated indirect addressing register, FSR0, to be selected for this.

The PUSH and POP code segments could either be in each Interrupt Service Routine, or could be subroutines that were called. Depending on the application, other registers may also need to be saved.

TABLE 7-3: SPECIAL FUNCTION REGISTERS (CONTINUED)

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
Bank 2											
10h	TMR1	Timer1's Register								xxxx xxxx	uuuu uuuu
11h	TMR2	Timer2's Register								xxxx xxxx	uuuu uuuu
12h	TMR3L	Timer3's Register; Low Byte								xxxx xxxx	uuuu uuuu
13h	TMR3H	Timer3's Register; High Byte								xxxx xxxx	uuuu uuuu
14h	PR1	Timer1's Period Register								xxxx xxxx	uuuu uuuu
15h	PR2	Timer2's Period Register								xxxx xxxx	uuuu uuuu
16h	PR3L/CA1L	Timer3's Period Register - Low Byte/Capture1 Register; Low Byte								xxxx xxxx	uuuu uuuu
17h	PR3H/CA1H	Timer3's Period Register - High Byte/Capture1 Register; High Byte								xxxx xxxx	uuuu uuuu
Bank 3											
10h	PW1DCL	DC1	DC0	—	—	—	—	—	—	xx-- ----	uu-- ----
11h	PW2DCL	DC1	DC0	TM2PW2	—	—	—	—	—	xx0- ----	uu0- ----
12h	PW1DCH	DC9	DC8	DC7	DC6	DC5	DC4	DC3	DC2	xxxx xxxx	uuuu uuuu
13h	PW2DCH	DC9	DC8	DC7	DC6	DC5	DC4	DC3	DC2	xxxx xxxx	uuuu uuuu
14h	CA2L	Capture2 Low Byte								xxxx xxxx	uuuu uuuu
15h	CA2H	Capture2 High Byte								xxxx xxxx	uuuu uuuu
16h	TCON1	CA2ED1	CA2ED0	CA1ED1	CA1ED0	T16	TMR3CS	TMR2CS	TMR1CS	0000 0000	0000 0000
17h	TCON2	CA2OVF	CA1OVF	PWM2ON	PWM1ON	CA1/$\overline{\text{PR3}}$	TMR3ON	TMR2ON	TMR1ON	0000 0000	0000 0000
Bank 4											
10h	PIR2	SSPIF	BCLIF	ADIF	—	CA4IF	CA3IF	TX2IF	RC2IF	000- 0010	000- 0010
11h	PIE2	SSPIE	BCLIE	ADIE	—	CA4IE	CA3IE	TX2IE	RC2IE	000- 0000	000- 0000
12h	Unimplemented	—	—	—	—	—	—	—	—	---- ----	---- ----
13h	RCSTA2	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00u
14h	RCREG2	Serial Port Receive Register for USART2								xxxx xxxx	uuuu uuuu
15h	TXSTA2	CSRC	TX9	TXEN	SYNC	—	—	TRMT	TX9D	0000 --1x	0000 --1u
16h	TXREG2	Serial Port Transmit Register for USART2								xxxx xxxx	uuuu uuuu
17h	SPBRG2	Baud Rate Generator for USART2								0000 0000	0000 0000
Bank 5:											
10h	DDRF	Data Direction Register for PORTF								1111 1111	1111 1111
11h	PORTF(4)	RF7/ AN11	RF6/ AN10	RF5/ AN9	RF4/ AN8	RF3/ AN7	RF2/ AN6	RF1/ AN5	RF0/ AN4	0000 0000	0000 0000
12h	DDRG	Data Direction Register for PORTG								1111 1111	1111 1111
13h	PORTG(4)	RG7/ TX2/CK2	RG6/ RX2/DT2	RG5/ PWM3	RG4/ CAP3	RG3/ AN0	RG2/ AN1	RG1/ AN2	RG0/ AN3	xxxx 0000	uuuu 0000
14h	ADCON0	CHS3	CHS2	CHS1	CHS0	—	GO/$\overline{\text{DONE}}$	—	ADON	0000 -0-0	0000 -0-0
15h	ADCON1	ADCS1	ADCS0	ADFM	—	PCFG3	PCFG2	PCFG1	PCFG0	000- 0000	000- 0000
16h	ADRESL	A/D Result Register Low Byte								xxxx xxxx	uuuu uuuu
17h	ADRESH	A/D Result Register High Byte								xxxx xxxx	uuuu uuuu

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

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for PC<15:8> whose contents are updated from, or transferred to, the upper byte of the program counter.

2: The $\overline{\text{TO}}$ and $\overline{\text{PD}}$ status bits in CPUSTA are not affected by a $\overline{\text{MCLR}}$ Reset.

3: Bank 8 and associated registers are only implemented on the PIC17C76X devices.

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.

9.0 HARDWARE MULTIPLIER

All PIC17C7XX devices have an 8 x 8 hardware multiplier included in the ALU of the device. By making the multiply a hardware operation, it completes in a single instruction cycle. This is an unsigned multiply that gives a 16-bit result. The result is stored into the 16-bit Product register (PRODH:PRODL). The multiplier does not affect any flags in the ALUSTA register.

Making the 8 x 8 multiplier execute in a single cycle gives the following advantages:

- Higher computational throughput
- Reduces code size requirements for multiply algorithms

The performance increase allows the device to be used in applications previously reserved for Digital Signal Processors.

Table 9-1 shows a performance comparison between PIC17CXXX devices using the single cycle hardware multiply and performing the same function without the hardware multiply.

Example 9-1 shows the sequence to do an 8 x 8 unsigned multiply. Only one instruction is required when one argument of the multiply is already loaded in the WREG register.

Example 9-2 shows the sequence to do an 8 x 8 signed multiply. To account for the sign bits of the arguments, each argument's most significant bit (MSb) is tested and the appropriate subtractions are done.

EXAMPLE 9-1: 8 x 8 UNSIGNED MULTIPLY ROUTINE

```
MOVFP    ARG1, WREG  ;
MULWF    ARG2        ; ARG1 * ARG2 ->
                     ;   PRODH:PRODL
```

EXAMPLE 9-2: 8 x 8 SIGNED MULTIPLY ROUTINE

```
MOVFP    ARG1, WREG
MULWF    ARG2        ; ARG1 * ARG2 ->
                     ;   PRODH:PRODL
BTFSC    ARG2, SB    ; Test Sign Bit
SUBWF    PRODH, F    ; PRODH = PRODH
                     ;         - ARG1
MOVFP    ARG2, WREG
BTFSC    ARG1, SB    ; Test Sign Bit
SUBWF    PRODH, F    ; PRODH = PRODH
                     ;         - ARG2
```

TABLE 9-1: PERFORMANCE COMPARISON

Routine	Multiply Method	Program Memory (Words)	Cycles (Max)	Time		
				@ 33 MHz	@ 16 MHz	@ 8 MHz
8 x 8 unsigned	Without hardware multiply	13	69	8.364 μs	17.25 μs	34.50 μs
	Hardware multiply	1	1	0.121 μs	0.25 μs	0.50 μs
8 x 8 signed	Without hardware multiply	—	—	—	—	—
	Hardware multiply	6	6	0.727 μs	1.50 μs	3.0 μs
16 x 16 unsigned	Without hardware multiply	21	242	29.333 μs	60.50 μs	121.0 μs
	Hardware multiply	24	24	2.91 μs	6.0 μs	12.0 μs
16 x 16 signed	Without hardware multiply	52	254	30.788 μs	63.50 μs	127.0 μs
	Hardware multiply	36	36	4.36 μs	9.0 μs	18.0 μs

10.4 PORTD and DDRD Registers

PORTD is an 8-bit bi-directional port. The corresponding data direction register is DDRD. A '1' in DDRD configures the corresponding port pin as an input. A '0' in the DDRD register configures the corresponding port pin as an output. Reading PORTD reads the status of the pins, whereas writing to PORTD will write to the port latch. PORTD is multiplexed with the system bus. When operating as the system bus, PORTD is the high order byte of the address/data bus (AD15:AD8). The timing for the system bus is shown in the Electrical Specifications section.

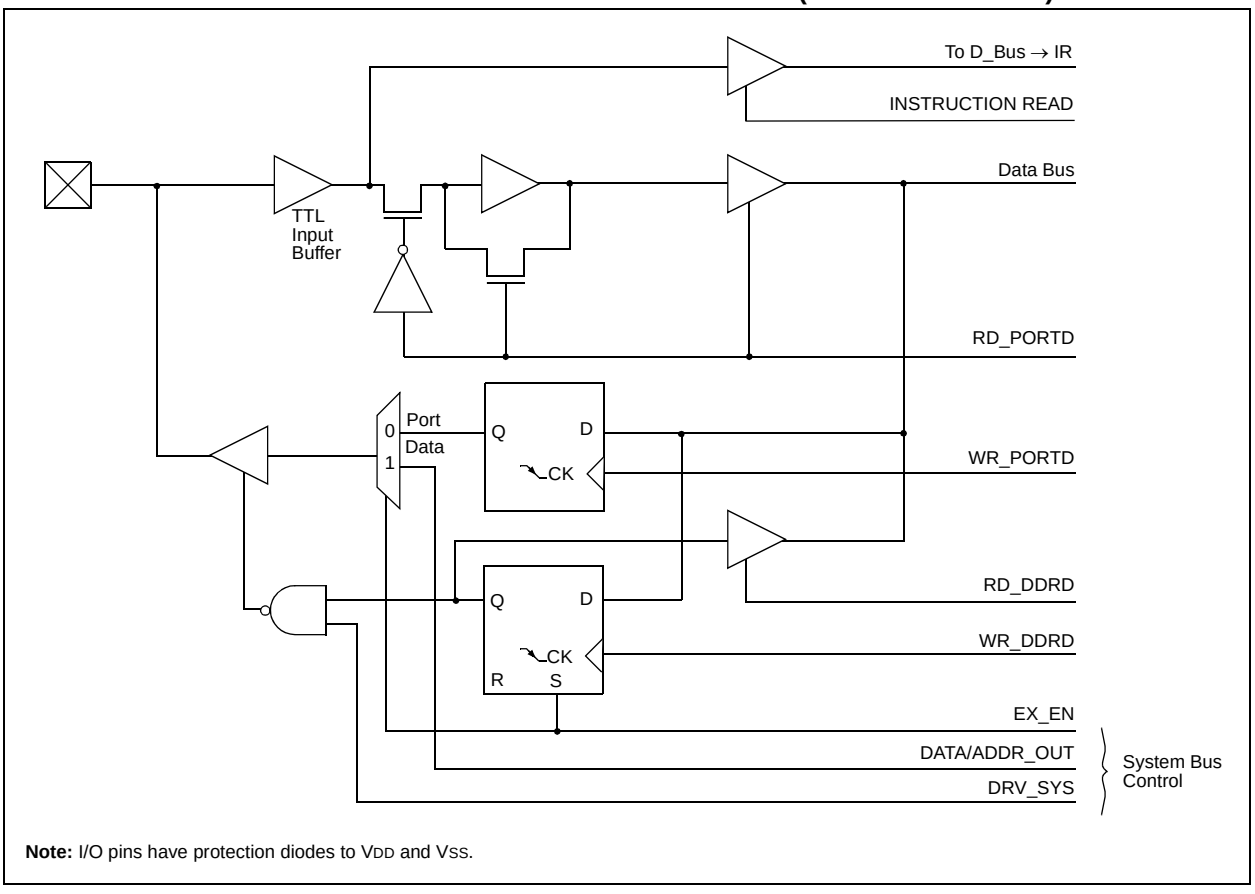
Note:	This port is configured as the system bus when the device's configuration bits are selected to Microprocessor or Extended Microcontroller modes. In the two other microcontroller modes, this port is a general purpose I/O.

Example 10-4 shows an instruction sequence to initialize PORTD. 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 10-4: INITIALIZING PORTD

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

FIGURE 10-10: BLOCK DIAGRAM OF RD7:RD0 PORT PINS (IN I/O PORT MODE)



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

10.10.2 SUCCESSIVE OPERATIONS ON I/O PORTS

The actual write to an I/O port happens at the end of an instruction cycle, whereas for reading, the data must be valid at the beginning of the instruction cycle (Figure 10-20). Therefore, care must be exercised if a write followed by a read operation is carried out on the same I/O port. The sequence of instructions should be such to allow the pin voltage to stabilize (load dependent) before executing the instruction that reads the values on that I/O port. Otherwise, the previous state of that pin may be read into the CPU, rather than the "new" state. When in doubt, it is better to separate these instructions with a `NOP`, or another instruction not accessing this I/O port.

Figure 10-21 shows the I/O model which causes this situation. As the effective capacitance (C) becomes larger, the rise/fall time of the I/O pin increases. As the device frequency increases, or the effective capacitance increases, the possibility of this subsequent PORTx read-modify-write instruction issue increases. This effective capacitance includes the effects of the board traces.

The best way to address this is to add a series resistor at the I/O pin. This resistor allows the I/O pin to get to the desired level before the next instruction.

The use of `NOP` instructions between the subsequent PORTx read-modify-write instructions, is a lower cost solution, but has the issue that the number of `NOP` instructions is dependent on the effective capacitance C and the frequency of the device.

FIGURE 10-20: SUCCESSIVE I/O OPERATION

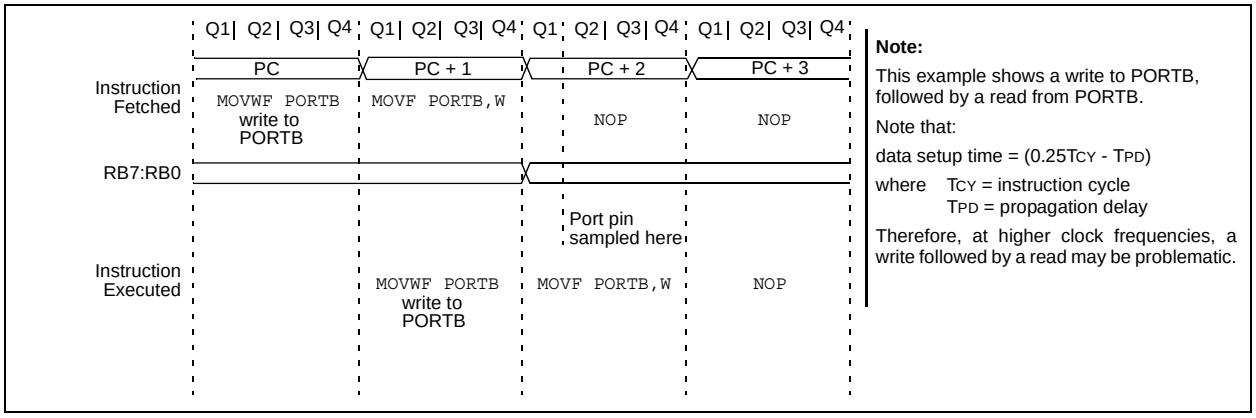
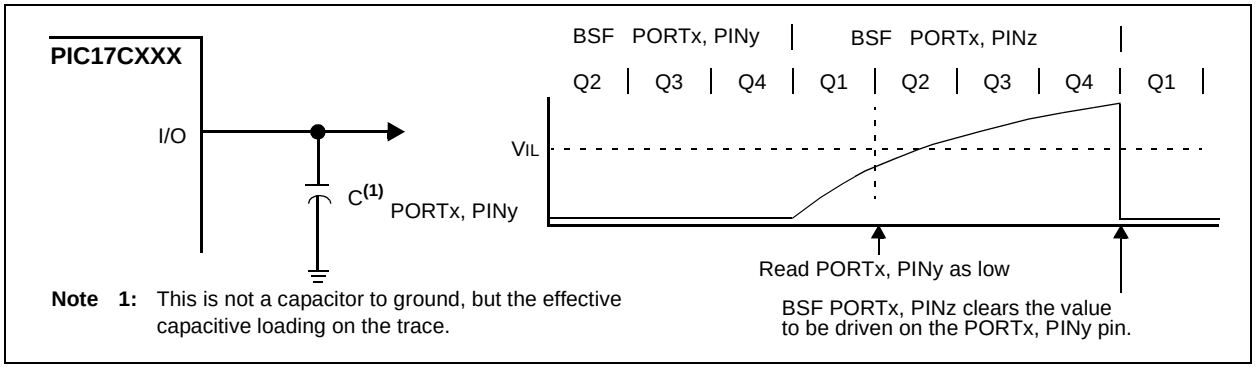


FIGURE 10-21: I/O CONNECTION ISSUES



Note 1: This is not a capacitor to ground, but the effective capacitive loading on the trace.

FIGURE 14-1: USART TRANSMIT

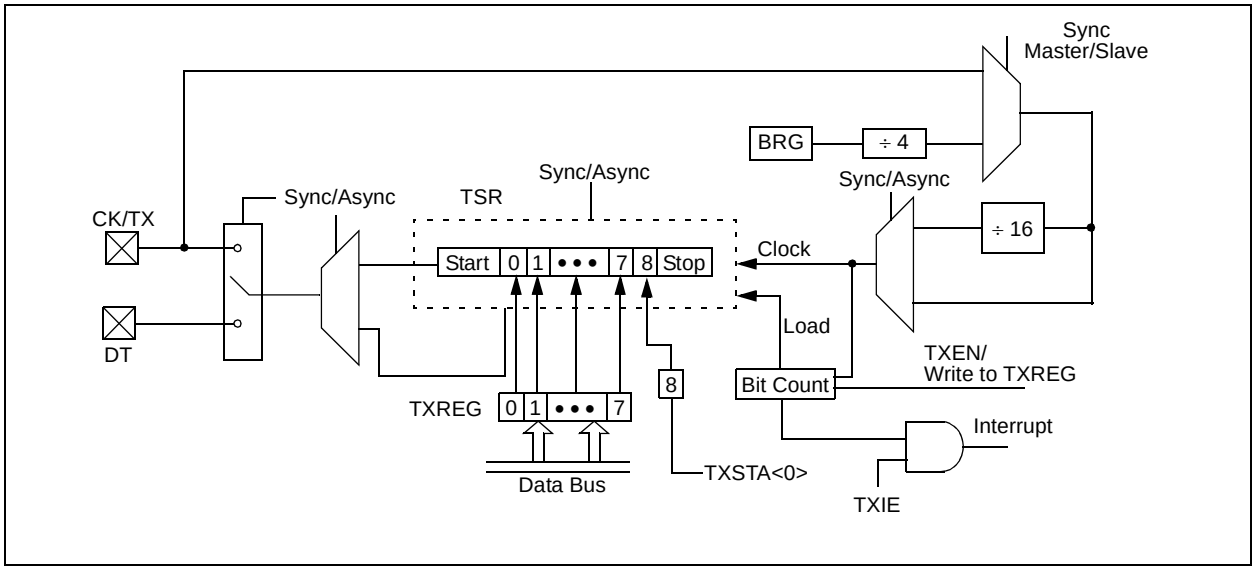
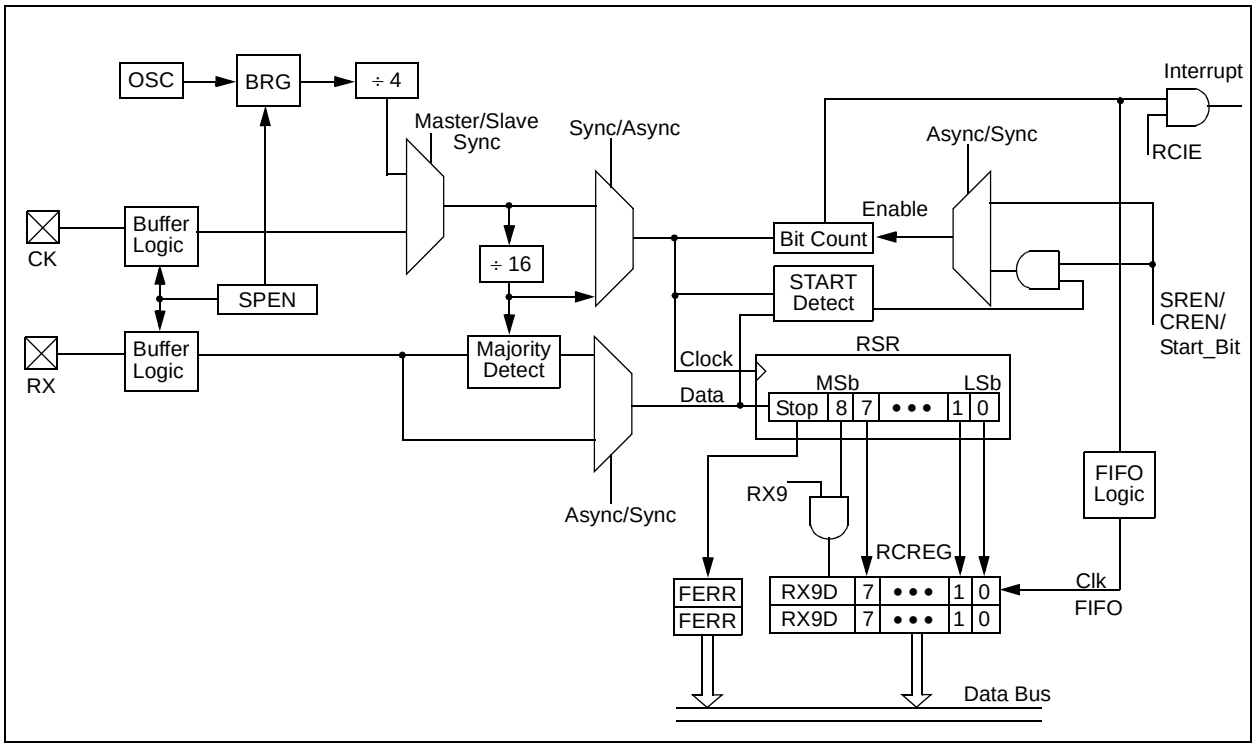


FIGURE 14-2: USART RECEIVE



A typical transmit sequence would go as follows:

a) The user generates a START Condition by setting the START enable bit (SEN) in SSPCON2.
b) SSPIF is set. The module will wait the required START time before any other operation takes place.
c) The user loads the SSPBUF with address to transmit.
d) Address is shifted out the SDA pin until all 8 bits are transmitted.
e) The MSSP Module shifts in the ACK bit from the slave device and writes its value into the SSPCON2 register (SSPCON2<6>).
f) The module generates an interrupt at the end of the ninth clock cycle by setting SSPIF.
g) The user loads the SSPBUF with eight bits of data.
h) DATA is shifted out the SDA pin until all 8 bits are transmitted.
i) The MSSP Module shifts in the ACK bit from the slave device, and writes its value into the SSPCON2 register (SSPCON2<6>).
j) The MSSP module generates an interrupt at the end of the ninth clock cycle by setting the SSPIF bit.
k) The user generates a STOP condition by setting the STOP enable bit PEN in SSPCON2.
l) Interrupt is generated once the STOP condition is complete.

15.2.8 BAUD RATE GENERATOR

In I^2C Master mode, the reload value for the BRG is located in the lower 7 bits of the SSPADD register (Figure 15-18). When the BRG is loaded with this value, the BRG counts down to 0 and stops until another reload has taken place. The BRG count is decremented twice per instruction cycle (TCY), on the Q2 and Q4 clock.

In I^2C Master mode, the BRG is reloaded automatically. If Clock Arbitration is taking place, for instance, the BRG will be reloaded when the SCL pin is sampled high (Figure 15-19).

FIGURE 15-18: BAUD RATE GENERATOR BLOCK DIAGRAM

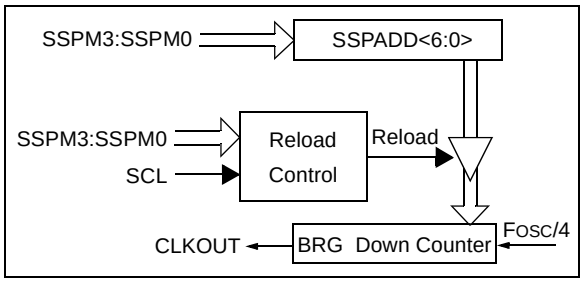
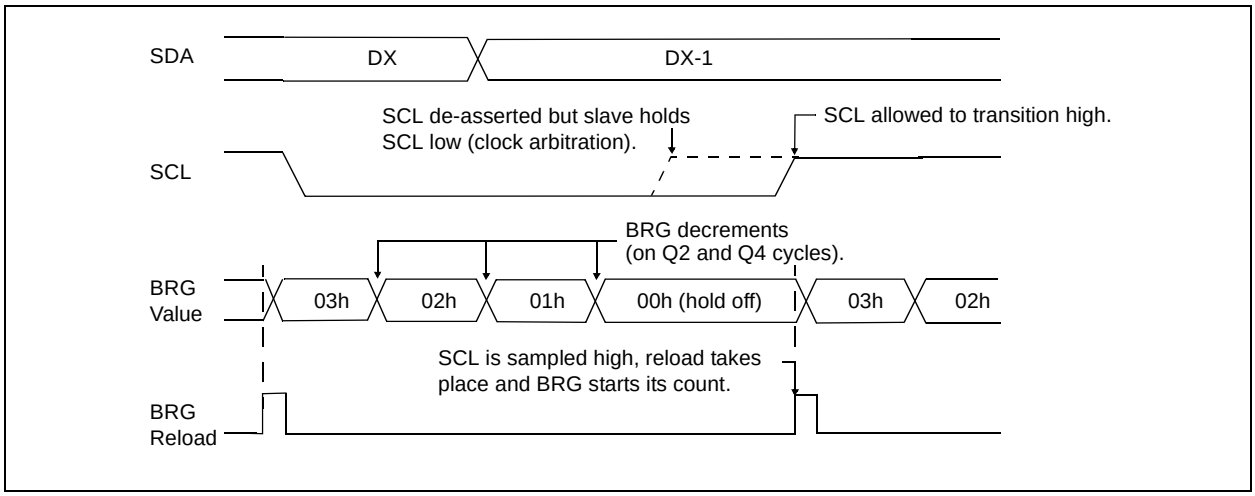


FIGURE 15-19: BAUD RATE GENERATOR TIMING WITH CLOCK ARBITRATION



15.2.12 I^2C 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).

REGISTER 16-2: ADCON1 REGISTER (ADDRESS 15h, BANK 5)

R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADCS1	ADCS0	ADFM	—	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

bit 7-6 **ADCS1:ADCS0**: A/D Conversion Clock Select bits
00 = FOSC/8
01 = FOSC/32
10 = FOSC/64
11 = FRC (clock derived from an internal RC oscillator)

bit 5 **ADFM**: A/D Result Format Select
1 = Right justified. 6 Most Significant bits of ADRESH are read as '0'.
0 = Left justified. 6 Least Significant bits of ADRESL are read as '0'.

bit 4 **Unimplemented**: Read as '0'

bit 3-1 **PCFG3:PCFG1**: A/D Port Configuration Control bits

bit 0 **PCFG0**: A/D Voltage Reference Select bit
1 = A/D reference is the VREF+ and VREF- pins
0 = A/D reference is AVDD and AVSS

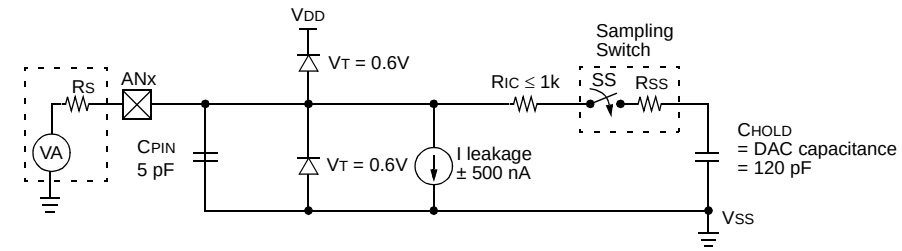
Note: When this bit is set, ensure that the A/D voltage reference specifications are met.

PCFG3:PCFG0	AN15	AN14	AN13	AN12	AN11	AN10	AN9	AN8	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0
000x	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
001x	D	A	A	A	A	A	A	A	D	A	A	A	A	A	A	A
010x	D	D	A	A	A	A	A	A	D	D	A	A	A	A	A	A
011x	D	D	D	A	A	A	A	A	D	D	D	A	A	A	A	A
100x	D	D	D	D	A	A	A	A	D	D	D	D	A	A	A	A
101x	D	D	D	D	D	A	A	A	D	D	D	D	D	A	A	A
110x	D	D	D	D	D	D	A	A	D	D	D	D	D	D	A	A
111x	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D

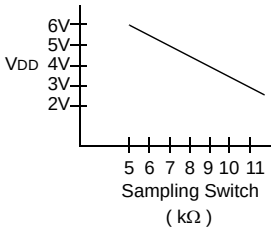
A = Analog input D = Digital I/O

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)



INCF **Increment f**

Syntax: [*label*] INCF f,d

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

Operation: (f) + 1 → (dest)

Status Affected: OV, C, DC, Z

Encoding:

0001	010d	ffff	ffff

Description: The contents of register 'f' are incremented. 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: `INCF CNT, 1`

Before Instruction
CNT = 0xFF
Z = 0
C = ?

After Instruction
CNT = 0x00
Z = 1
C = 1

INCFSZ **Increment f, skip if 0**

Syntax: [*label*] INCFSZ f,d

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

Operation: (f) + 1 → (dest)
skip if result = 0

Status Affected: None

Encoding:

0001	111d	ffff	ffff

Description: The contents of register 'f' are incremented. If 'd' is 0, the result is placed in WREG. If 'd' is 1, the result is placed back in register 'f'.
If the result is 0, the next instruction, which is already fetched is discarded and a NOP is executed instead, making it a two-cycle instruction.

Words: 1

Cycles: 1(2)

Q Cycle Activity:

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

If skip:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation

Example:
```
HERE    INCFSZ  CNT, 1
NZERO   :
ZERO    :
```

Before Instruction
PC = Address (HERE)

After Instruction
CNT = CNT + 1
If CNT = 0;
PC = Address (ZERO)
If CNT ≠ 0;
PC = Address (NZERO)

20.1 DC Characteristics

PIC17LC7XX-08 (Commercial, Industrial)			**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial				
PIC17C7XX-16 (Commercial, Industrial, Extended) **PIC17C7XX-33** (Commercial, Industrial, Extended)			**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				
Param. No.	**Sym**	**Characteristic**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
D001	VDD	**Supply Voltage**					
		PIC17LC7XX	3.0	—	5.5	V	
D001		PIC17C7XX-33 PIC17C7XX-16	4.5 VBOR	— —	5.5 5.5	V V	(BOR enabled) **(Note 5)**
D002	VDR	**RAM Data Retention Voltage (Note 1)**	1.5	—	—	V	Device in SLEEP mode
D003	VPOR	**VDD Start Voltage** to ensure internal Power-on Reset signal	—	Vss	—	V	See section on Power-on Reset for details
D004	SVDD	**VDD Rise Rate** to ensure proper operation					
		PIC17LCXX	0.010	—	—	V/ms	See section on Power-on Reset for details
D004		PIC17CXX	0.085	—	—	V/ms	See section on Power-on Reset for details
D005	VBOR	**Brown-out Reset** voltage trip point	3.65	—	4.35	V	
D006	VPORTP	**Power-on Reset** trip point	—	2.2	—	V	VDD = VPORTP

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

Note 1: This is the limit to which VDD can be lowered in SLEEP mode without losing RAM data.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors such as I/O pin loading and switching rate, oscillator type, internal code execution pattern and temperature also have an impact on the current consumption.

The test conditions for all IDD measurements in active operation mode are:

OSC1 = external square wave, from rail to rail; all I/O pins tri-stated, pulled to VDD or VSS, T0CKI = VDD, MCLR = VDD; WDT disabled.

Current consumed from the oscillator and I/O's driving external capacitive or resistive loads needs to be considered.

For the RC oscillator, the current through the external pull-up resistor (R) can be estimated as: VDD/(2 • R).

For capacitive loads, the current can be estimated (for an individual I/O pin) as (CL • VDD) • f

CL = Total capacitive load on the I/O pin; f = average frequency the I/O pin switches.

The capacitive currents are most significant when the device is configured for external execution (includes Extended Microcontroller mode).

3: The power-down current in SLEEP mode does not depend on the oscillator type. Power-down current is measured with the part in SLEEP mode, with all I/O pins in hi-impedance state and tied to VDD or VSS.

4: For RC osc configuration, current through REXT is not included. The current through the resistor can be estimated by the formula IR = VDD/2REXT (mA) with REXT in kOhm.

5: This is the voltage where the device enters the Brown-out Reset. When BOR is enabled, the device (-16) will operate correctly to this trip point.

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.

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

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

2: A fast mode (400 KHz) I^2C bus device can be used in a standard mode I^2C bus system, but the parameter # 107 ≥ 250 ns must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCL signal. If such a device does stretch the LOW period of the SCL signal, it must output the next data bit to the SDA line. Parameter #102 + #107 = 1000 + 250 = 1250 ns (for 100 kHz mode) before the SCL line is released.

3: C_b is specified to be from 10-400pF. The minimum specifications are characterized with C_b=10pF. The rise time spec (t_r) is characterized with R_p=R_p min. The minimum fall time specification (t_f) is characterized with C_b=10pF,and R_p=R_p max. These are only valid for fast mode operation (V_{DD}=4.5-5.5V) and where the SPM bit (SSPSTAT<7>) =1.)

4: Max specifications for these parameters are valid for falling edge only. Specs are characterized with R_p=R_p min and C_b=400pF for standard mode, 200pF for fast mode, and 10pF for 1MHz mode.

FIGURE 20-19: USART SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING

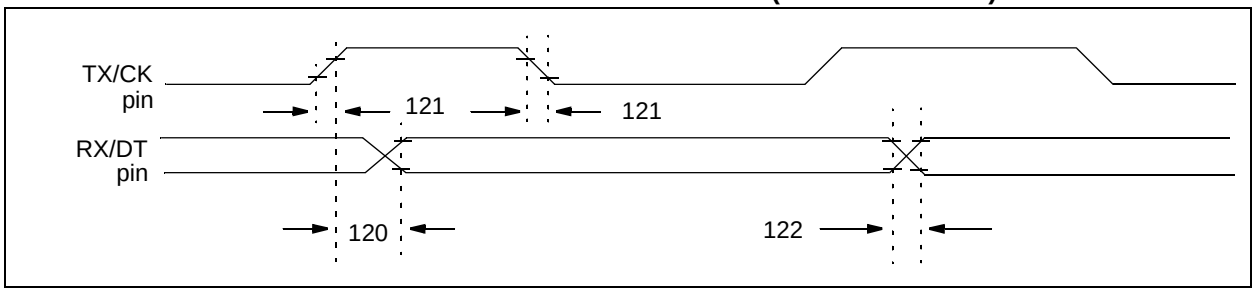


TABLE 20-14: USART SYNCHRONOUS TRANSMISSION REQUIREMENTS

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

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