Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

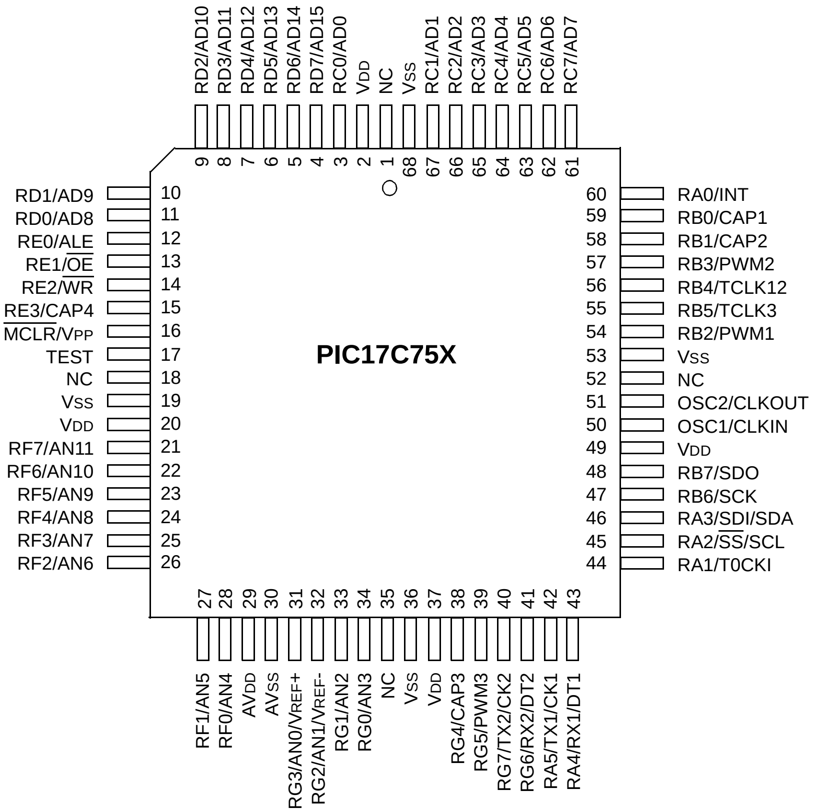
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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"

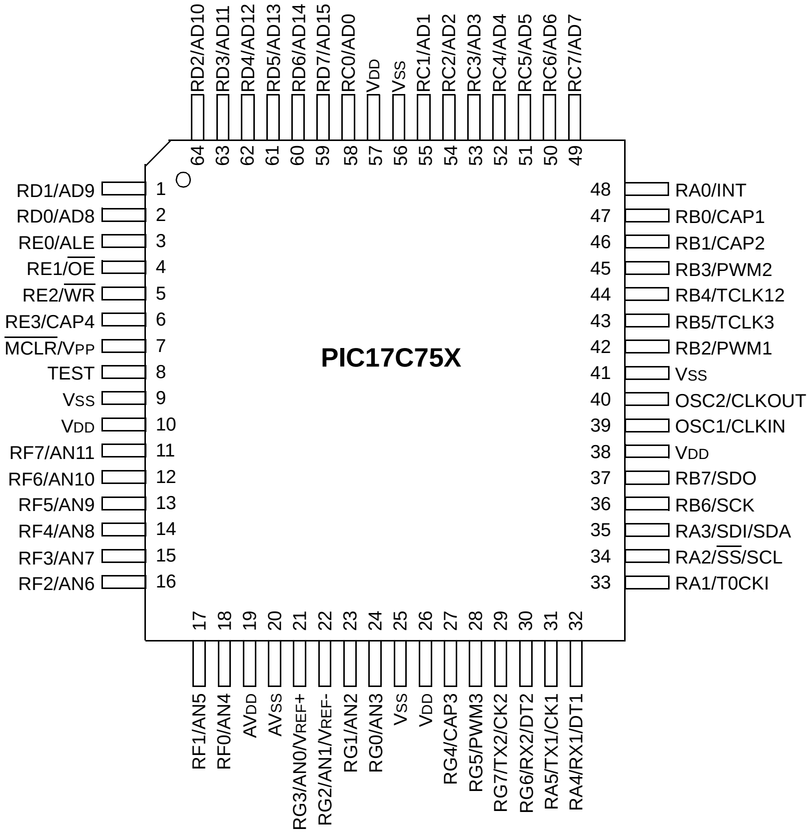
Details	
Product Status	Obsolete
Core Processor	PIC
Core Size	8-Bit
Speed	33MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	50
Program Memory Size	16KB (8K x 16)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	454 x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 12x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	68-LCC (J-Lead)
Supplier Device Package	68-PLCC (24.23x24.23)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic17c752t-33i-l

Pin Diagrams cont.'d

68-Pin PLCC



64-Pin TQFP



TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at **docerrors@mail.microchip.com** or fax the **Reader Response Form** in the back of this data sheet to (480) 792-4150. We welcome your feedback.

Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Web site at:

http://www.microchip.com

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000A is version A of document DS30000).

Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Web site; http://www.microchip.com
- Your local Microchip sales office (see last page)
- The Microchip Corporate Literature Center; U.S. FAX: (480) 792-7277

When contacting a sales office or the literature center, please specify which device, revision of silicon and data sheet (include literature number) you are using.

Customer Notification System

Register on our web site at **www.microchip.com/cn** to receive the most current information on all of our products.

NOTES:

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.

NOTES:

7.2 Data Memory Organization

Data memory is partitioned into two areas. The first is the General Purpose Registers (GPR) area, and the second is the Special Function Registers (SFR) area. The SFRs control and provide status of device operation.

Portions of data memory are banked, this occurs in both areas. The GPR area is banked to allow greater than 232 bytes of general purpose RAM.

Banking requires the use of control bits for bank selection. These control bits are located in the Bank Select Register (BSR). If an access is made to the unbanked region, the BSR bits are ignored. Figure 7-5 shows the data memory map organization.

Instructions `MOVPF` and `MOVFP` provide the means to move values from the peripheral area ("P") to any location in the register file ("F"), and vice-versa. The definition of the "P" range is from 0h to 1Fh, while the "F" range is 0h to FFh. The "P" range has six more locations than peripheral registers, which can be used as General Purpose Registers. This can be useful in some applications where variables need to be copied to other locations in the general purpose RAM (such as saving status information during an interrupt).

The entire data memory can be accessed either directly, or indirectly (through file select registers FSR0 and FSR1) (see Section 7.4). Indirect addressing uses the appropriate control bits of the BSR for access into the banked areas of data memory. The BSR is explained in greater detail in Section 7.8.

7.2.1 GENERAL PURPOSE REGISTER (GPR)

All devices have some amount of GPR area. The GPRs are 8-bits wide. When the GPR area is greater than 232, it must be banked to allow access to the additional memory space.

All the PIC17C7XX devices have banked memory in the GPR area. To facilitate switching between these banks, the `MOVLR bank` instruction has been added to the instruction set. GPRs are not initialized by a Power-on Reset and are unchanged on all other RESETS.

7.2.2 SPECIAL FUNCTION REGISTERS (SFR)

The SFRs are used by the CPU and peripheral functions to control the operation of the device (Figure 7-5). These registers are static RAM.

The SFRs can be classified into two sets, those associated with the "core" function and those related to the peripheral functions. Those registers related to the "core" are described here, while those related to a peripheral feature are described in the section for each peripheral feature.

The peripheral registers are in the banked portion of memory, while the core registers are in the unbanked region. To facilitate switching between the peripheral banks, the `MOVLB bank` instruction has been provided.

7.3 Stack Operation

PIC17C7XX devices have a 16 x 16-bit hardware stack (Figure 7-1). The stack is not part of either the program or data memory space, and the stack pointer is neither readable nor writable. The PC (Program Counter) is "PUSH'd" onto the stack when a `CALL` or `LCALL` instruction is executed, or an interrupt is acknowledged. The stack is "POP'd" in the event of a `RETURN`, `RETLW`, or a `RETFIE` instruction execution. PCLATH is not affected by a "PUSH" or a "POP" operation.

The stack operates as a circular buffer, with the stack pointer initialized to '0' after all RESETS. There is a stack available bit (STKAV) to allow software to ensure that the stack will not overflow. The STKAV bit is set after a device RESET. When the stack pointer equals Fh, STKAV is cleared. When the stack pointer rolls over from Fh to 0h, the STKAV bit will be held clear until a device RESET.

Note 1: There is not a status bit for stack underflow. The STKAV bit can be used to detect the underflow which results in the stack pointer being at the Top-of-Stack.

2: There are no instruction mnemonics called PUSH or POP. These are actions that occur from the execution of the `CALL`, `RETURN`, `RETLW` and `RETFIE` instructions, or the vectoring to an interrupt vector.

3: After a RESET, if a "POP" operation occurs before a "PUSH" operation, the STKAV bit will be cleared. This will appear as if the stack is full (underflow has occurred). If a "PUSH" operation occurs next (before another "POP"), the STKAV bit will be locked clear. Only a device RESET will cause this bit to set.

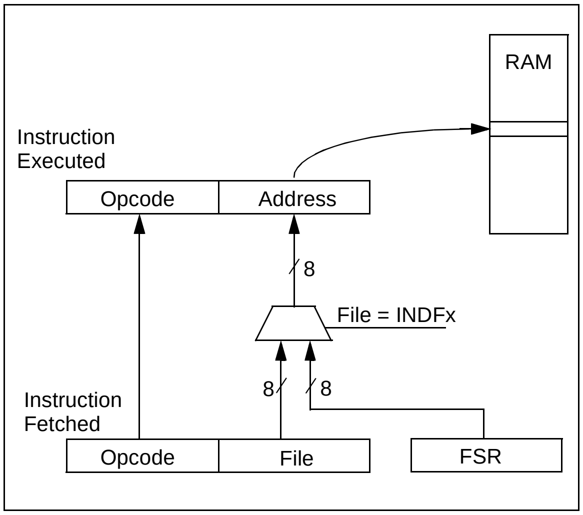
After the device is "PUSH'd" sixteen times (without a "POP"), the seventeenth push overwrites the value from the first push. The eighteenth push overwrites the second push (and so on).

7.4 Indirect Addressing

Indirect addressing is a mode of addressing data memory where the data memory address in the instruction is not fixed. That is, the register that is to be read or written can be modified by the program. This can be useful for data tables in the data memory. Figure 7-6 shows the operation of indirect addressing. This depicts the moving of the value to the data memory address specified by the value of the FSR register.

Example 7-1 shows the use of indirect addressing to clear RAM in a minimum number of instructions. A similar concept could be used to move a defined number of bytes (block) of data to the USART transmit register (TXREG). The starting address of the block of data to be transmitted could easily be modified by the program.

FIGURE 7-6: INDIRECT ADDRESSING



7.4.1 INDIRECT ADDRESSING REGISTERS

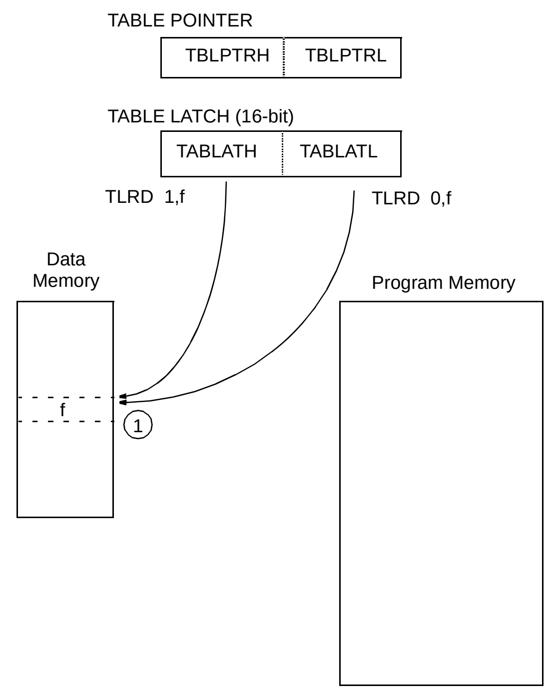
The PIC17C7XX has four registers for indirect addressing. These registers are:

- INDF0 and FSR0
- INDF1 and FSR1

Registers INDF0 and INDF1 are not physically implemented. Reading or writing to these registers activates indirect addressing, with the value in the corresponding FSR register being the address of the data. The FSR is an 8-bit register and allows addressing anywhere in the 256-byte data memory address range. For banked memory, the bank of memory accessed is specified by the value in the BSR.

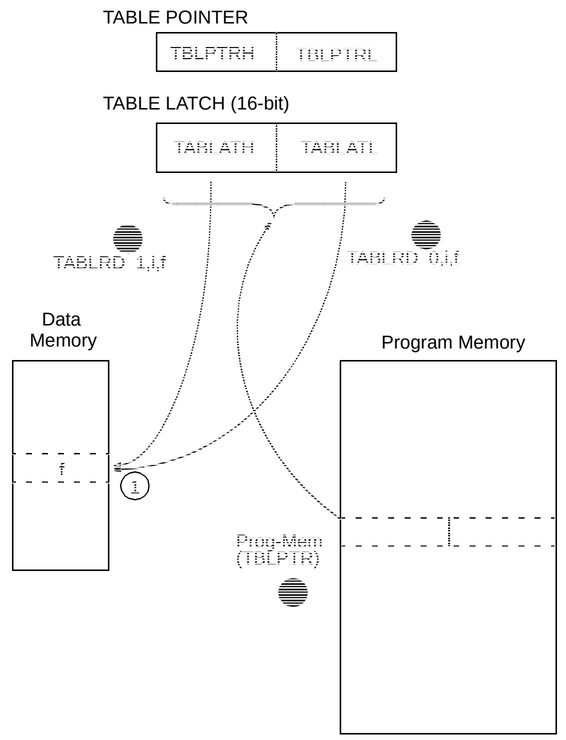
If file INDF0 (or INDF1) itself is read indirectly via an FSR, all '0's are read (Zero bit is set). Similarly, if INDF0 (or INDF1) is written to indirectly, the operation will be equivalent to a `NOP`, and the status bits are not affected.

FIGURE 8-3: `TLRD` INSTRUCTION OPERATION



Step 1: 8-bit value from TABLAT (16-bit) high or low byte, loaded into register 'f'.

FIGURE 8-4: `TABLRD` INSTRUCTION OPERATION



Step 1: 8-bit value from TABLAT (16-bit) high or low byte, loaded into register 'f'.

2: 16-bit value at Program Memory (TBLPTR), loaded into TABLAT register.

3: If "i" = 1, then TBLPTR = TBLPTR + 1,
If "i" = 0, then TBLPTR is unchanged.

8.1 Table Writes to Internal Memory

A table write operation to internal memory causes a long write operation. The long write is necessary for programming the internal EPROM. Instruction execution is halted while in a long write cycle. The long write will be terminated by any enabled interrupt. To ensure that the EPROM location has been well programmed, a minimum programming time is required (see specification #D114). Having only one interrupt enabled to terminate the long write ensures that no unintentional interrupts will prematurely terminate the long write.

The sequence of events for programming an internal program memory location should be:

1. Disable all interrupt sources, except the source to terminate EPROM program write.
2. Raise $\overline{\text{MCLR}}$/VPP pin to the programming voltage.
3. Clear the WDT.
4. Do the table write. The interrupt will terminate the long write.
5. Verify the memory location (table read).

> **Note 1:** Programming requirements must be met. See timing specification in electrical specifications for the desired device. Violating these specifications (including temperature) may result in EPROM locations that are not fully programmed and may lose their state over time.
>
> **2:** If the VPP requirement is not met, the table write is a 2-cycle write and the program memory is unchanged.

8.1.1 TERMINATING LONG WRITES

An interrupt source or RESET are the only events that terminate a long write operation. Terminating the long write from an interrupt source requires that the interrupt enable and flag bits are set. The GLINTD bit only enables the vectoring to the interrupt address.

If the T0CKI, RA0/INT, or TMR0 interrupt source is used to terminate the long write, the interrupt flag of the highest priority enabled interrupt, will terminate the long write and automatically be cleared.

> **Note 1:** If an interrupt is pending, the `TABLWT` is aborted (a `NOP` is executed). The highest priority pending interrupt, from the T0CKI, RA0/INT, or TMR0 sources that is enabled, has its flag cleared.
>
> **2:** If the interrupt is not being used for the program write timing, the interrupt should be disabled. This will ensure that the interrupt is not lost, nor will it terminate the long write prematurely.

If a peripheral interrupt source is used to terminate the long write, the interrupt enable and flag bits must be set. The interrupt flag will not be automatically cleared upon the vectoring to the interrupt vector address.

The GLINTD bit determines whether the program will branch to the interrupt vector when the long write is terminated. If GLINTD is clear, the program will vector, if GLINTD is set, the program will not vector to the interrupt address.

TABLE 8-1: INTERRUPT - TABLE WRITE INTERACTION

Interrupt Source	GLINTD	Enable Bit	Flag Bit	Action
RA0/INT, TMR0, T0CKI	0	1	1	Terminate long table write (to internal program memory), branch to interrupt vector (branch clears flag bit).
	0	1	0	None.
	1	0	x	None.
	1	1	1	Terminate long table write, do not branch to interrupt vector (flag is automatically cleared).
Peripheral	0	1	1	Terminate long table write, branch to interrupt vector.
	0	1	0	None.
	1	0	x	None.
	1	1	1	Terminate long table write, do not branch to interrupt vector (flag remains set).

10.9 PORTJ and DDRJ Registers (PIC17C76X only)

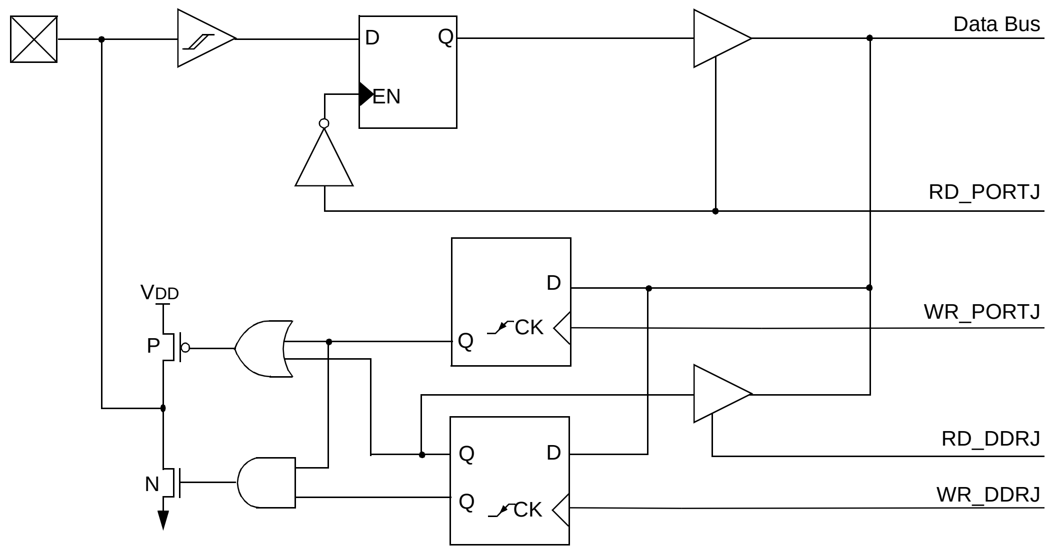
PORTJ is an 8-bit wide, bi-directional port. The corresponding data direction register is DDRJ. A '1' in DDRJ configures the corresponding port pin as an input. A '0' in the DDRJ register configures the corresponding port pin as an output. Reading PORTJ reads the status of the pins, whereas writing to PORTJ will write to the respective port latch.

PORTJ is a general purpose I/O port.

EXAMPLE 10-9: INITIALIZING PORTJ

```
MOVLB   8          ; Select Bank 8
CLRF    PORTJ, F   ; Initialize PORTJ data
                   ; latches before setting
                   ; the data direction
                   ; register
MOVLW    0xCF      ; Value used to initialize
                   ; data direction
MOVWF    DDRJ      ; Set RJ<3:0> as inputs
                   ; RJ<5:4> as outputs
                   ; RJ<7:6> as inputs
```

FIGURE 10-19: PORTJ BLOCK DIAGRAM



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

10.10.2 SUCCESSIVE OPERATIONS ON I/O PORTS

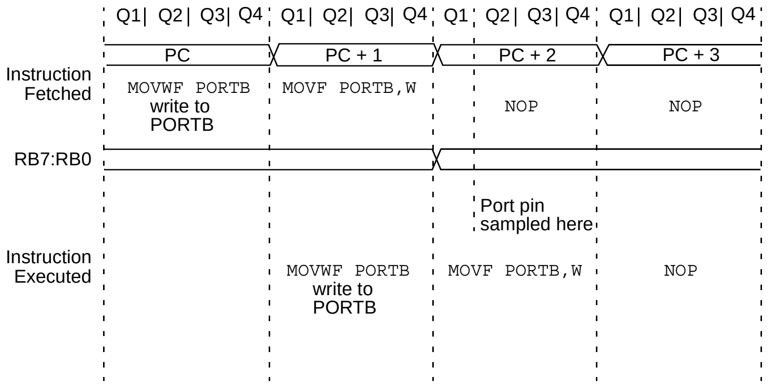
The actual write to an I/O port happens at the end of an instruction cycle, whereas for reading, the data must be valid at the beginning of the instruction cycle (Figure 10-20). Therefore, care must be exercised if a write followed by a read operation is carried out on the same I/O port. The sequence 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



Note:

This example shows a write to PORTB, followed by a read from PORTB.

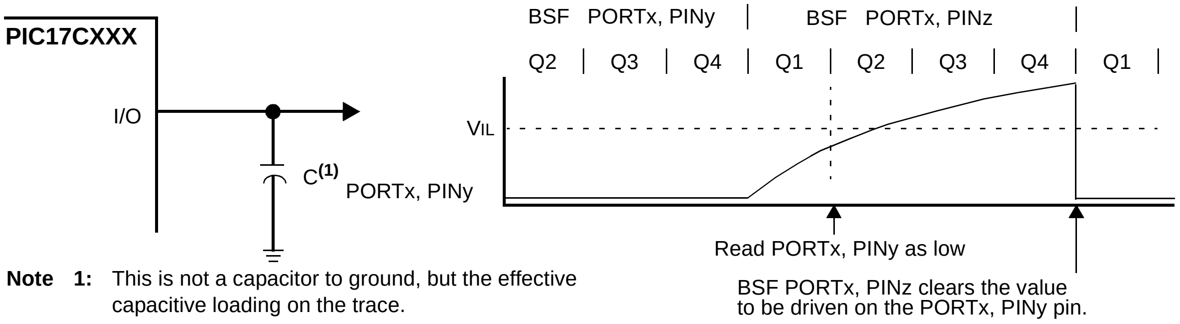
Note that:

data setup time = (0.25TCY - TPD)

where TCY = instruction cycle
TPD = propagation delay

Therefore, at higher clock frequencies, a write followed by a read may be problematic.

FIGURE 10-21: I/O CONNECTION ISSUES



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

TABLE 14-8: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER TRANSMISSION

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

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

FIGURE 14-8: SYNCHRONOUS TRANSMISSION

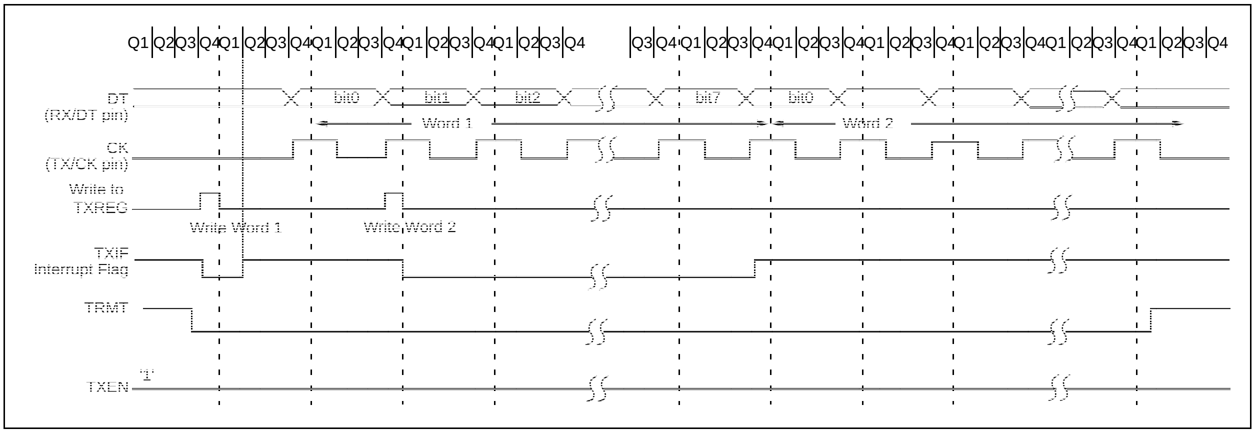
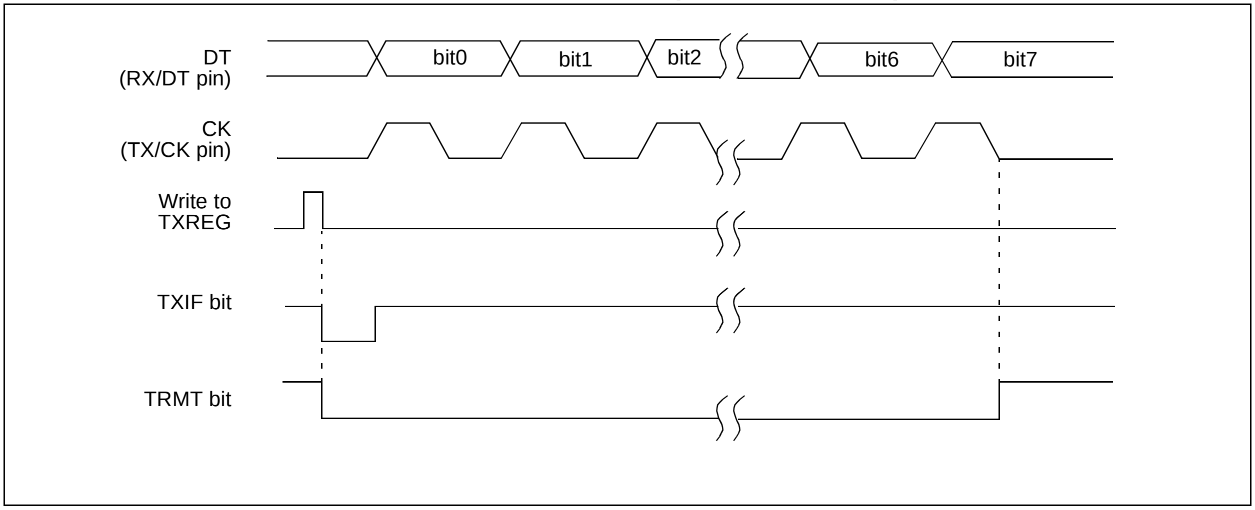


FIGURE 14-9: SYNCHRONOUS TRANSMISSION (THROUGH TXEN)



REGISTER 15-2: SSPCON1: SYNC SERIAL PORT CONTROL REGISTER1 (ADDRESS 11h, BANK 6)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	$\overline{\text{SSPM1}}$	$\overline{\text{SSPM0}}$
bit 7							bit 0

bit 7 **WCOL**: Write Collision Detect bit

Master mode:
1 = A write to the SSPBUF register was attempted while the I^{2}C conditions were not valid for a transmission to be started
0 = No collision

Slave mode:
1 = The SSPBUF register is written while it is still transmitting the previous word (must be cleared in software)
0 = No collision

bit 6 **SSPOV**: Receive Overflow Indicator bit

In SPI mode:
1 = A new byte is received while the SSPBUF register is still holding the previous data. In case of overflow, the data in SSPSR is lost. Overflow can only occur in Slave mode. In Slave mode, the user must read the SSPBUF, even if only transmitting data, to avoid setting overflow. In Master mode, the overflow bit is not set, since each new reception (and transmission) is initiated by writing to the SSPBUF register. (Must be cleared in software.)
0 = No overflow

In I^{2}C mode:
1 = A byte is received while the SSPBUF register is still holding the previous byte. SSPOV is a “don’t care” in Transmit mode. (Must be cleared in software.)
0 = No overflow

bit 5 **SSPEN**: Synchronous Serial Port Enable bit
In both modes, when enabled, these pins must be properly configured as input or output.

In SPI mode:
1 = Enables serial port and configures SCK, SDO, SDI and $\overline{\text{SS}}$ as the source of the serial port pins
0 = Disables serial port and configures these pins as I/O port pins

In I^{2}C mode:
1 = Enables the serial port and configures the SDA and SCL pins as the source of the serial port pins
0 = Disables serial port and configures these pins as I/O port pins

Note: In SPI mode, these pins must be properly configured as input or output.

bit 4 **CKP**: Clock Polarity Select bit

In SPI mode:
1 = Idle state for clock is a high level
0 = Idle state for clock is a low level

In I^{2}C Slave mode:
SCK release control
1 = Enable clock
0 = Holds clock low (clock stretch). (Used to ensure data setup time.)

In I^{2}C Master mode:
Unused in this mode

bit 3-0 **SSPM3:SSPM0**: Synchronous Serial Port Mode Select bits
0000 = SPI Master mode, clock = FOSC/4
0001 = SPI Master mode, clock = FOSC/16
0010 = SPI Master mode, clock = FOSC/64
0011 = SPI Master mode, clock = TMR2 output/2
0100 = SPI Slave mode, clock = SCK pin, $\overline{\text{SS}}$ pin control enabled
0101 = SPI Slave mode, clock = SCK pin, $\overline{\text{SS}}$ pin control disabled, $\overline{\text{SS}}$ can be used as I/O pin
0110 = I^{2}C Slave mode, 7-bit address
0111 = I^{2}C Slave mode, 10-bit address
1000 = I^{2}C Master mode, clock = FOSC / (4 * (SSPADD+1))
1xx1 = Reserved
1x1x = Reserved

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

SWAPF Swap f

Syntax:	[*label*] SWAPF f,d
Operands:	0 ≤ f ≤ 255 d ∈ [0,1]
Operation:	f<3:0> → dest<7:4>; f<7:4> → dest<3:0>
Status Affected:	None
Encoding:	`0001` `110d` `ffff` `ffff`
Description:	The upper and lower nibbles of register 'f' are exchanged. If 'd' is 0, the result is placed in WREG. If 'd' is 1, the result is placed in register 'f'.
Words:	1
Cycles:	1

Q Cycle Activity:

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

<u>Example</u>: `SWAPF REG, 0`

Before Instruction

REG = 0x53

After Instruction

REG = 0x35

TABLRD Table Read

Syntax:	[*label*] TABLRD t,i,f
Operands:	0 ≤ f ≤ 255 i ∈ [0,1] t ∈ [0,1]
Operation:	If t = 1, TBLATH → f; If t = 0, TBLATL → f; Prog Mem (TBLPTR) → TBLAT; If i = 1, TBLPTR + 1 → TBLPTR If i = 0, TBLPTR is unchanged
Status Affected:	None
Encoding:	`1010` `10ti` `ffff` `ffff`
Description:	1. A byte of the table latch (TBLAT) is moved to register file 'f'. If t = 1: the high byte is moved; If t = 0: the low byte is moved. 2. Then, the contents of the program memory location pointed to by the 16-bit Table Pointer (TBLPTR) are loaded into the 16-bit Table Latch (TBLAT). 3. If i = 1: TBLPTR is incremented; If i = 0: TBLPTR is not incremented.
Words:	1
Cycles:	2 (3-cycle if f = PCL)

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register TBLATH or TBLATL	Process Data	Write register 'f'
No operation	No operation (Table Pointer on Address bus)	No operation	No operation ($\overline{\text{OE}}$ goes low)

19.13 PICDEM 3 Low Cost PIC16CXXX Demonstration Board

The PICDEM 3 demonstration board is a simple demonstration board that supports the PIC16C923 and PIC16C924 in the PLCC package. It will also support future 44-pin PLCC microcontrollers with an LCD Module. All the necessary hardware and software is included to run the basic demonstration programs. The user can program the sample microcontrollers provided with the PICDEM 3 demonstration board on a PRO MATE II device programmer, or a PICSTART Plus development programmer with an adapter socket, and easily test firmware. The MPLAB ICE in-circuit emulator may also be used with the PICDEM 3 demonstration board to test firmware. A prototype area has been provided to the user for adding hardware and connecting it to the microcontroller socket(s). Some of the features include a RS-232 interface, push button switches, a potentiometer for simulated analog input, a thermistor and separate headers for connection to an external LCD module and a keypad. Also provided on the PICDEM 3 demonstration board is a LCD panel, with 4 commons and 12 segments, that is capable of displaying time, temperature and day of the week. The PICDEM 3 demonstration board provides an additional RS-232 interface and Windows software for showing the demultiplexed LCD signals on a PC. A simple serial interface allows the user to construct a hardware demultiplexer for the LCD signals.

19.14 PICDEM 17 Demonstration Board

The PICDEM 17 demonstration board is an evaluation board that demonstrates the capabilities of several Microchip microcontrollers, including PIC17C752, PIC17C756A, PIC17C762 and PIC17C766. All necessary hardware is included to run basic demo programs, which are supplied on a 3.5-inch disk. A programmed sample is included and the user may erase it and program it with the other sample programs using the PRO MATE II device programmer, or the PICSTART Plus development programmer, and easily debug and test the sample code. In addition, the PICDEM 17 demonstration board supports downloading of programs to and executing out of external FLASH memory on board. The PICDEM 17 demonstration board is also usable with the MPLAB ICE in-circuit emulator, or the PICMASTER emulator and all of the sample programs can be run and modified using either emulator. Additionally, a generous prototype area is available for user hardware.

19.15 KEELOQ Evaluation and Programming Tools

KEELOQ evaluation and programming tools support Microchip's HCS Secure Data Products. The HCS evaluation kit includes a LCD display to show changing codes, a decoder to decode transmissions and a programming interface to program test transmitters.

20.4 Timing Diagrams and Specifications

FIGURE 20-6: EXTERNAL CLOCK TIMING

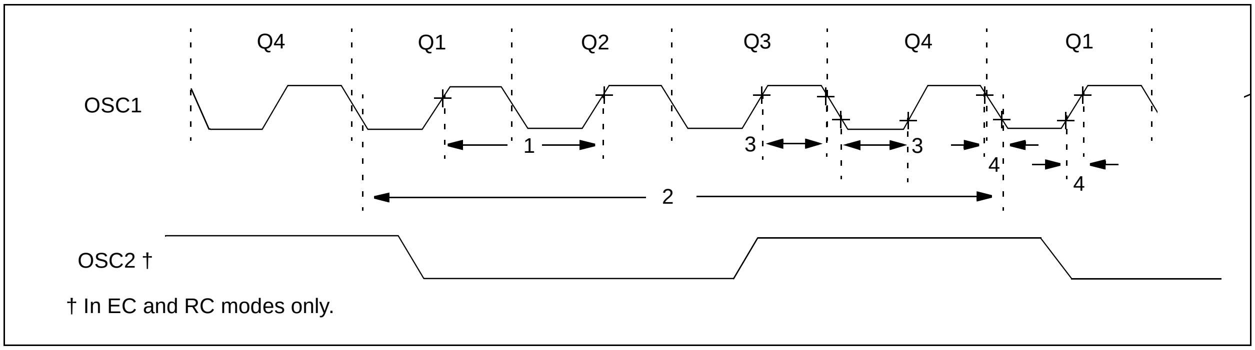


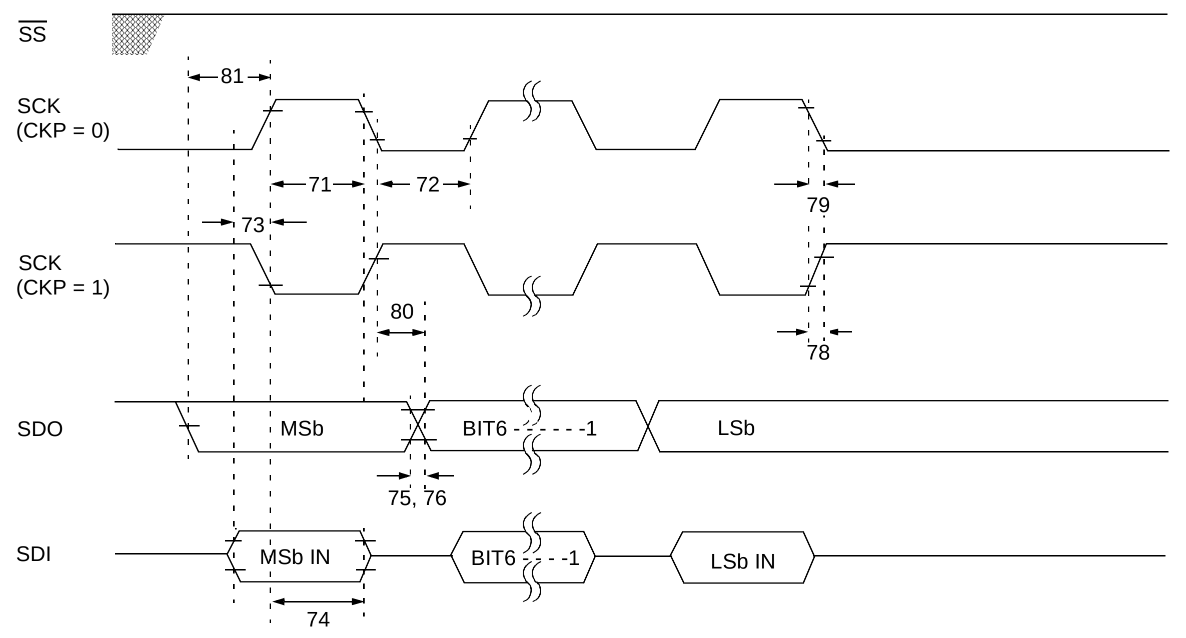
TABLE 20-1: EXTERNAL CLOCK TIMING REQUIREMENTS

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
	FOSC	**External CLKIN Frequency (Note 1)**	DC DC DC	— — —	8 16 33	MHz MHz MHz	EC osc mode - 08 devices (8 MHz devices) - 16 devices (16 MHz devices) - 33 devices (33 MHz devices)
		Oscillator Frequency (Note 1)	DC 2 2 2 DC	— — — — —	4 8 16 33 2	MHz MHz MHz MHz MHz	RC osc mode XT osc mode - 08 devices (8 MHz devices) - 16 devices (16 MHz devices) - 33 devices (33 MHz devices) LF osc mode
1	TOSC	**External CLKIN Period (Note 1)**	125 62.5 30.3	— — —	— — —	ns ns ns	EC osc mode - 08 devices (8 MHz devices) - 16 devices (16 MHz devices) - 33 devices (33 MHz devices)
		Oscillator Period (Note 1)	250 125 62.5 30.3 500	— — — — —	— 1,000 1,000 1,000 —	ns ns ns ns ns	RC osc mode XT osc mode - 08 devices (8 MHz devices) - 16 devices (16 MHz devices) - 33 devices (33 MHz devices) LF osc mode
2	TCY	**Instruction Cycle Time (Note 1)**	121.2	4/FOSC	DC	ns	
3	TosL, TosH	**Clock in (OSC1) High or Low Time**	10	—	—	ns	EC oscillator
4	TosR, TosF	**Clock in (OSC1) Rise or Fall Time**	—	—	5	ns	EC oscillator

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

Note 1: Instruction cycle period (TCY) equals four times the input oscillator time base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at “min.” values with an external clock applied to the OSC1/CLKIN pin. When an external clock input is used, the “max.” cycle time limit is “DC” (no clock) for all devices.

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



Note: Refer to Figure 20-5 for load conditions.

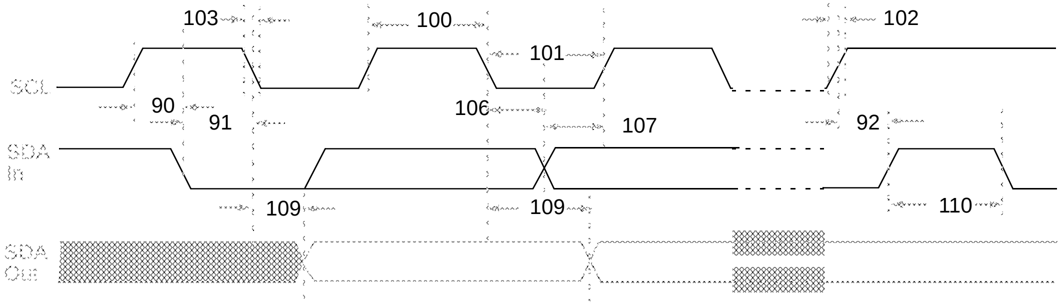
TABLE 20-9: SPI MODE REQUIREMENTS (MASTER MODE, CKE = 1)

Param. No.	Symbol	Characteristic		Min	Typ†	Max	Units	Conditions
71	TscH	SCK input high time (Slave mode)	Continuous	1.25TCY + 30	—	—	ns	
71A			Single Byte	40	—	—	ns	**(Note 1)**
72	TscL	SCK input low time (Slave mode)	Continuous	1.25 TCY + 30	—	—	ns	
72A			Single Byte	40	—	—	ns	**(Note 1)**
73	TdiV2scH, TdiV2scL	Setup time of SDI data input to SCK edge		100	—	—	ns	
73A	TB2B	Last clock edge of Byte1 to the 1st clock edge of Byte2		1.5TCY + 40	—	—	ns	**(Note 1)**
74	TscH2diL, TscL2diL	Hold time of SDI data input to SCK edge		100	—	—	ns	
75	TdoR	SDO data output rise time		—	10	25	ns	
76	TdoF	SDO data output fall time		—	10	25	ns	
78	TscR	SCK output rise time (Master mode)		—	10	25	ns	
79	TscF	SCK output fall time (Master mode)		—	10	25	ns	
80	TscH2doV, TscL2doV	SDO data output valid after SCK edge		—	—	50	ns	
81	TdoV2scH, TdoV2scL	SDO data output setup to SCK edge		Tcy	—	—	ns	

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

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

FIGURE 20-18: I^2C BUS DATA TIMING



Note: Refer to Figure 20-5 for load conditions.

TABLE 20-13: I^2C BUS DATA REQUIREMENTS

Param No.	Sym	Characteristic		Min	Max	Units	Conditions
100	Thigh	Clock high time	100 kHz mode	2(Tosc)(BRG + 1)	—	ms	
			400 kHz mode	2(Tosc)(BRG + 1)	—	ms	
			1 MHz mode(1)	2(Tosc)(BRG + 1)	—	ms	
101	Tlow	Clock low time	100 kHz mode	2(Tosc)(BRG + 1)	—	ms	
			400 kHz mode	2(Tosc)(BRG + 1)	—	ms	
			1 MHz mode(1)	2(Tosc)(BRG + 1)	—	ms	
102	Tr	SDA and SCL rise time	100 kHz mode	—	1000	ns	Cb is specified to be from 10 to 400 pF
			400 kHz mode	20 + 0.1Cb	300	ns	
			1 MHz mode(1)	—	300	ns	
103	Tf	SDA and SCL fall time	100 kHz mode	—	300	ns	Cb is specified to be from 10 to 400 pF
			400 kHz mode	20 + 0.1Cb	300	ns	
			1 MHz mode(1)	—	10	ns	
90	Tsu:sta	START condition setup time	100 kHz mode	2(Tosc)(BRG + 1)	—	ms	Only relevant for Repeated Start condition
			400 kHz mode	2(Tosc)(BRG + 1)	—	ms	
			1 MHz mode(1)	2(Tosc)(BRG + 1)	—	ms	
91	Thd:sta	START condition hold time	100 kHz mode	2(Tosc)(BRG + 1)	—	ms	After this period, the first clock pulse is generated
			400 kHz mode	2(Tosc)(BRG + 1)	—	ms	
			1 MHz mode(1)	2(Tosc)(BRG + 1)	—	ms	
106	Thd:dat	Data input hold time	100 kHz mode	0	—	ns	
			400 kHz mode	0	0.9	ms	
			1 MHz mode(1)	0	—	ns	
107	Tsu:dat	Data input setup time	100 kHz mode	250	—	ns	**(Note 2)**
			400 kHz mode	100	—	ns	
			1 MHz mode(1)	100	—	ns	
92	Tsu:sto	STOP condition setup time	100 kHz mode	2(Tosc)(BRG + 1)	—	ms	
			400 kHz mode	2(Tosc)(BRG + 1)	—	ms	
			1 MHz mode(1)	2(Tosc)(BRG + 1)	—	ms	
109	Taa	Output valid from clock	100 kHz mode	—	3500	ns	
			400 kHz mode	—	1000	ns	
			1 MHz mode(1)	—	400	ns	

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 $\geq$ 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 (VDD=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.

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^{2}C pins.

2: A fast mode (400 KHz) I^{2}C bus device can be used in a standard mode I^{2}C 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 (VDD=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

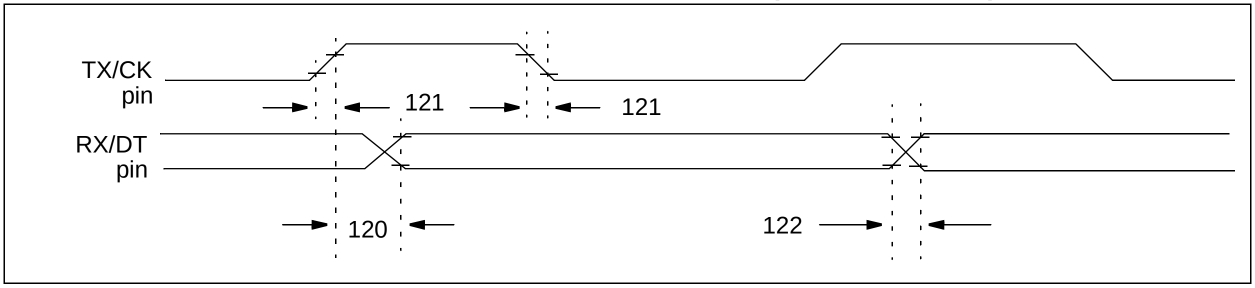


TABLE 20-14: USART SYNCHRONOUS TRANSMISSION REQUIREMENTS

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.

APPENDIX C: WHAT'S NEW

This is a new Data Sheet for the Following Devices:

- PIC17C752
- PIC17C756A
- PIC17C762
- PIC17C766

This Data Sheet is based on the PIC17C75X Data Sheet (DS30246A).

APPENDIX D: WHAT'S CHANGED

Clarified the TAD vs. device maximum operating frequency tables in Section 16.2.

Added device characteristic graphs and charts in Section 21.

Removed the "Preliminary" status from the entire document.

Revision C (January 2013)

Added a note to each package outline drawing.