Microchip Technology - PIC17C752T-16I/L Datasheet

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	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-16i-l

TABLE 5-4: INITIALIZATION CONDITIONS FOR SPECIAL FUNCTION REGISTERS (CONTINUED)

Register	Address	Power-on Reset Brown-out Reset	$\overline{\text{MCLR}}$ Reset WDT Reset	Wake-up from SLEEP through Interrupt
Bank 4				
PIR2	10h	000- 0010	000- 0010	uuu- uuuu(1)
PIE2	11h	000- 0000	000- 0000	uuu- uuuu
Unimplemented	12h	---- ----	---- ----	---- ----
RCSTA2	13h	0000 -00x	0000 -00u	uuuu -uuu
RCREG2	14h	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXSTA2	15h	0000 --1x	0000 --1u	uuuu --uu
TXREG2	16h	xxxx xxxx	uuuu uuuu	uuuu uuuu
SPBRG2	17h	0000 0000	0000 0000	uuuu uuuu
Bank 5				
DDRF	10h	1111 1111	1111 1111	uuuu uuuu
PORTF(4)	11h	0000 0000	0000 0000	uuuu uuuu
DDRG	12h	1111 1111	1111 1111	uuuu uuuu
PORTG(4)	13h	xxxx 0000	uuuu 0000	uuuu uuuu
ADCON0	14h	0000 -0-0	0000 -0-0	uuuu uuuu
ADCON1	15h	000- 0000	000- 0000	uuuu uuuu
ADRESL	16h	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADRESH	17h	xxxx xxxx	uuuu uuuu	uuuu uuuu
Bank 6				
SSPADD	10h	0000 0000	0000 0000	uuuu uuuu
SSPCON1	11h	0000 0000	0000 0000	uuuu uuuu
SSPCON2	12h	0000 0000	0000 0000	uuuu uuuu
SSPSTAT	13h	0000 0000	0000 0000	uuuu uuuu
SSPBUF	14h	xxxx xxxx	uuuu uuuu	uuuu uuuu
Unimplemented	15h	---- ----	---- ----	---- ----
Unimplemented	16h	---- ----	---- ----	---- ----
Unimplemented	17h	---- ----	---- ----	---- ----

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.

TABLE 5-4: INITIALIZATION CONDITIONS FOR SPECIAL FUNCTION REGISTERS (CONTINUED)

Register	Address	Power-on Reset Brown-out Reset	$\overline{\text{MCLR}}$ Reset WDT Reset	Wake-up from SLEEP through Interrupt
Bank 7				
PW3DCL	10h	`xx0- ----`	`uu0- ----`	`uuu- ----`
PW3DCH	11h	`xxxx xxxx`	`uuuu uuuu`	`uuuu uuuu`
CA3L	12h	`xxxx xxxx`	`uuuu uuuu`	`uuuu uuuu`
CA3H	13h	`xxxx xxxx`	`uuuu uuuu`	`uuuu uuuu`
CA4L	14h	`xxxx xxxx`	`uuuu uuuu`	`uuuu uuuu`
CA4H	15h	`xxxx xxxx`	`uuuu uuuu`	`uuuu uuuu`
TCON3	16h	`-000 0000`	`-000 0000`	`-uuu uuuu`
Unimplemented	17h	`---- ----`	`---- ----`	`---- ----`
Bank 8				
DDRH	10h	`1111 1111`	`1111 1111`	`uuuu uuuu`
PORTH[4]	11h	`xxxx xxxx`	`uuuu uuuu`	`uuuu uuuu`
DDRJ	12h	`1111 1111`	`1111 1111`	`uuuu uuuu`
PORTJ[4]	13h	`xxxx xxxx`	`uuuu uuuu`	`uuuu uuuu`
Unbanked				
PRODL	18h	`xxxx xxxx`	`uuuu uuuu`	`uuuu uuuu`
PRODH	19h	`xxxx xxxx`	`uuuu uuuu`	`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.

7.2.2.3 TMR0 Status/Control Register (T0STA)

This register contains various control bits. Bit7 (INTEDG) is used to control the edge upon which a signal on the RA0/INT pin will set the RA0/INT interrupt flag. The other bits configure Timer0, it’s prescaler and clock source.

REGISTER 7-3: T0STA REGISTER (ADDRESS: 05h, UNBANKED)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
INTEDG	T0SE	T0CS	T0PS3	T0PS2	T0PS1	T0PS0	—
bit 7							bit 0

bit 7 **INTEDG**: RA0/INT Pin Interrupt Edge Select bit
This bit selects the edge upon which the interrupt is detected.
1 = Rising edge of RA0/INT pin generates interrupt
0 = Falling edge of RA0/INT pin generates interrupt

bit 6 **T0SE**: Timer0 External Clock Input Edge Select bit
This bit selects the edge upon which TMR0 will increment.

When T0CS = 0 (External Clock):
1 = Rising edge of RA1/T0CKI pin increments TMR0 and/or sets the T0CKIF bit
0 = Falling edge of RA1/T0CKI pin increments TMR0 and/or sets a T0CKIF bit

When T0CS = 1 (Internal Clock):
Don’t care

bit 5 **T0CS**: Timer0 Clock Source Select bit
This bit selects the clock source for Timer0.
1 = Internal instruction clock cycle (TCY)
0 = External clock input on the T0CKI pin

bit 4-1 **T0PS3:T0PS0**: Timer0 Prescale Selection bits
These bits select the prescale value for Timer0.

T0PS3:T0PS0	Prescale Value
0000	1:1
0001	1:2
0010	1:4
0011	1:8
0100	1:16
0101	1:32
0110	1:64
0111	1:128
1xxx	1:256

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

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

7.8 Bank Select Register (BSR)

The BSR is used to switch between banks in the data memory area (Figure 7-9). In the PIC17C7XX devices, the entire byte is implemented. The lower nibble is used to select the peripheral register bank. The upper nibble is used to select the general purpose memory bank.

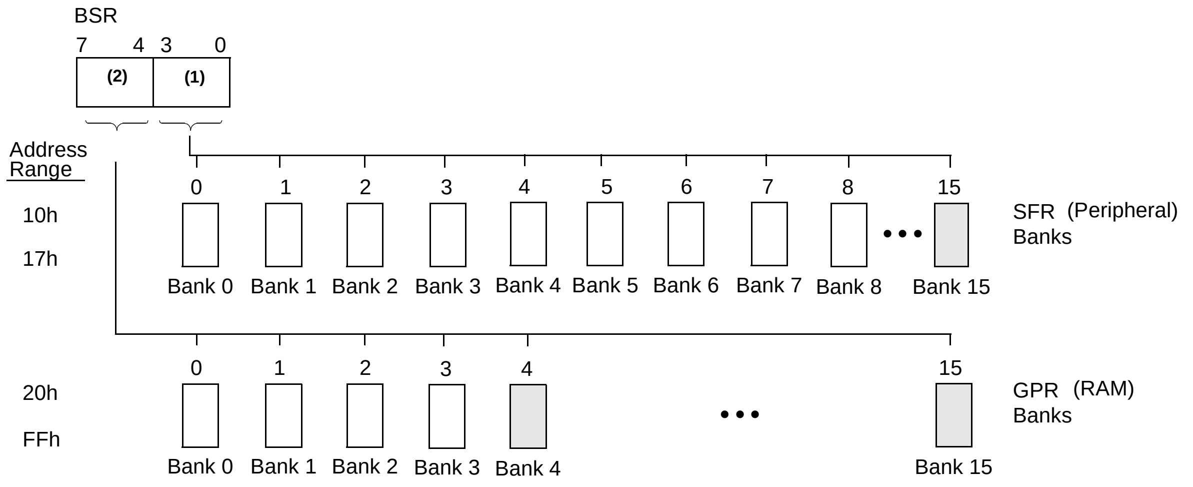
All the Special Function Registers (SFRs) are mapped into the data memory space. In order to accommodate the large number of registers, a banking scheme has been used. A segment of the SFRs, from address 10h to address 17h, is banked. The lower nibble of the bank select register (BSR) selects the currently active "peripheral bank." Effort has been made to group the peripheral registers of related functionality in one bank. However, it will still be necessary to switch from bank to bank in order to address all peripherals related to a single task. To assist this, a `MOVLB bank` instruction has been included in the instruction set.

The need for a large general purpose memory space dictated a general purpose RAM banking scheme. The upper nibble of the BSR selects the currently active general purpose RAM bank. To assist this, a `MOVLR bank` instruction has been provided in the instruction set.

If the currently selected bank is not implemented (such as Bank 13), any read will read all '0's. Any write is completed to the bit bucket and the ALU status bits will be set/cleared as appropriate.

Note:	Registers in Bank 15 in the Special Function Register area, are reserved for Microchip use. Reading of registers in this bank may cause random values to be read.

FIGURE 7-9: BSR OPERATION



Note 1: For the SFRs only Banks 0 through 8 are implemented. Selection of an unimplemented bank is not recommended. **Bank 15 is reserved for Microchip use, reading of registers in this bank may cause random values to be read.**

2: For the GPRs, Bank 3 is unimplemented on the PIC17C752 and the PIC17C762. Selection of an unimplemented bank is not recommended.

3: SFR Bank 8 is only implemented on the PIC17C76X.

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.

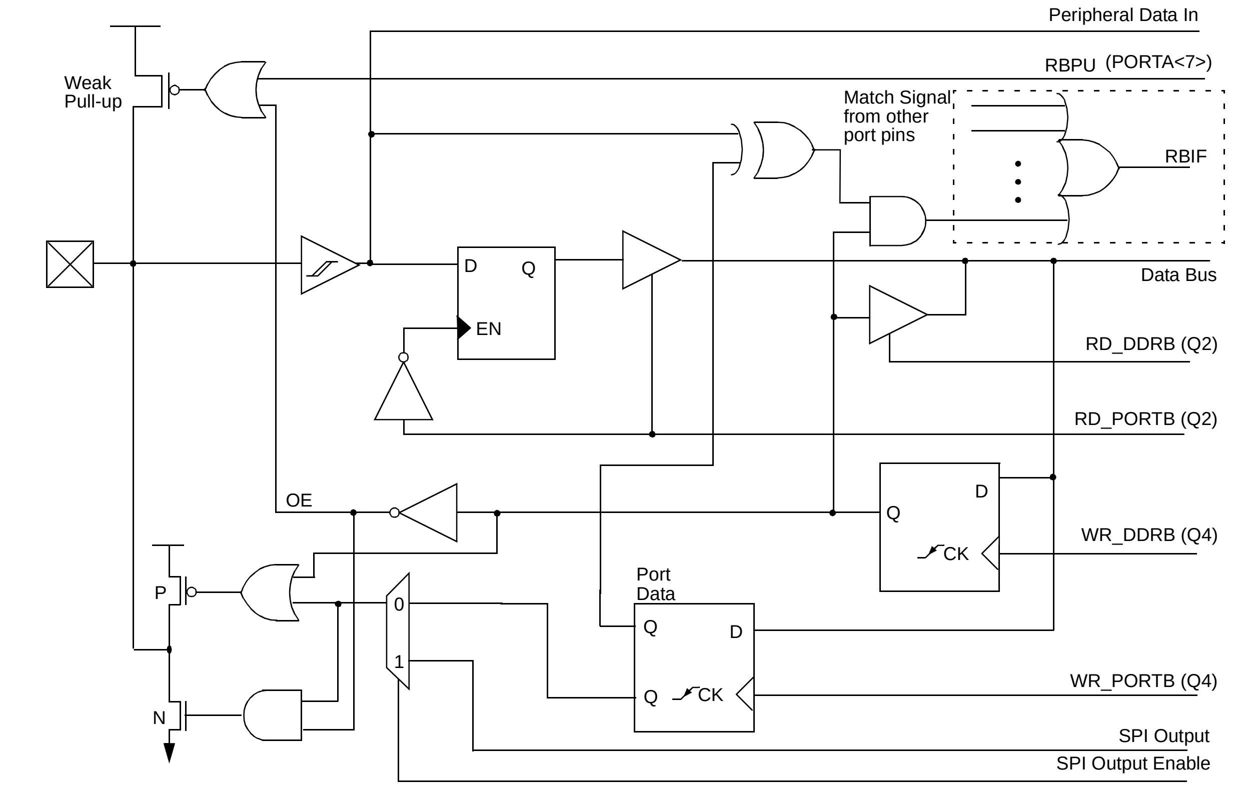
If a peripheral interrupt source is used to terminate the long write, the interrupt enable and flag bits must be set. The interrupt flag will not be automatically cleared upon the vectoring to the interrupt vector address.

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

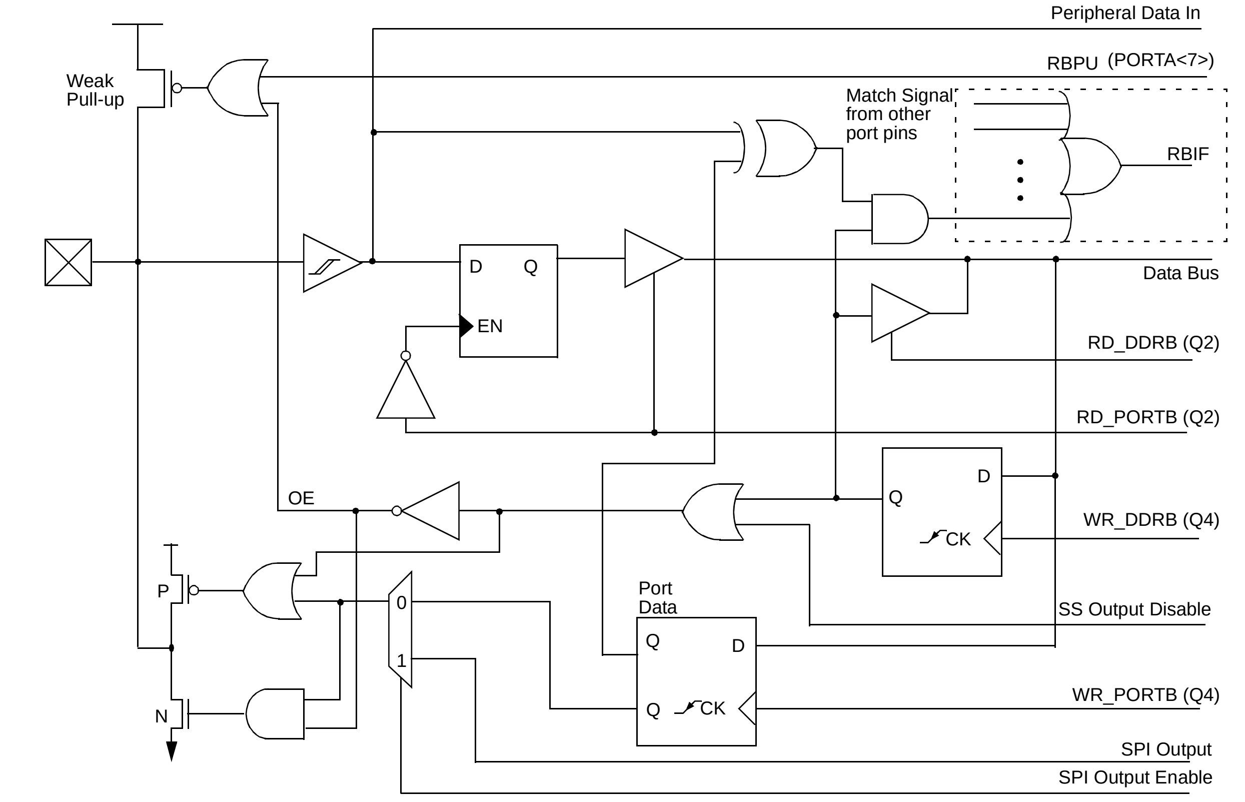
Interrupt Source	GLINTD	Enable Bit	Flag Bit	Action
RA0/INT, TMR0, T0CKI	0	1	1	Terminate long table write (to internal program memory), branch to interrupt vector (branch clears flag bit).
	0	1	0	None.
	1	0	x	None.
	1	1	1	Terminate long table write, do not branch to interrupt vector (flag is automatically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

FIGURE 10-7: BLOCK DIAGRAM OF RB6 PORT PIN



Note: I/O pin has protection diodes to Vdd and Vss.

FIGURE 10-8: BLOCK DIAGRAM OF RB7 PORT PIN



Note: I/O pin has protection diodes to VDD and VSS.

10.5 PORTE and DDRE Register

PORTE is a 4-bit bi-directional port. The corresponding data direction register is DDRE. A '1' in DDRE configures the corresponding port pin as an input. A '0' in the DDRE register configures the corresponding port pin as an output. Reading PORTE reads the status of the pins, whereas writing to PORTE will write to the port latch. PORTE is multiplexed with the system bus. When operating as the system bus, PORTE contains the control signals for the address/data bus (AD15:AD0). These control signals are Address Latch Enable (ALE), Output Enable ($\overline{OE}$) and Write ($\overline{WR}$). The control signals $\overline{OE}$ and $\overline{WR}$ are active low signals. The timing for the system bus is shown in the Electrical Specifications section.

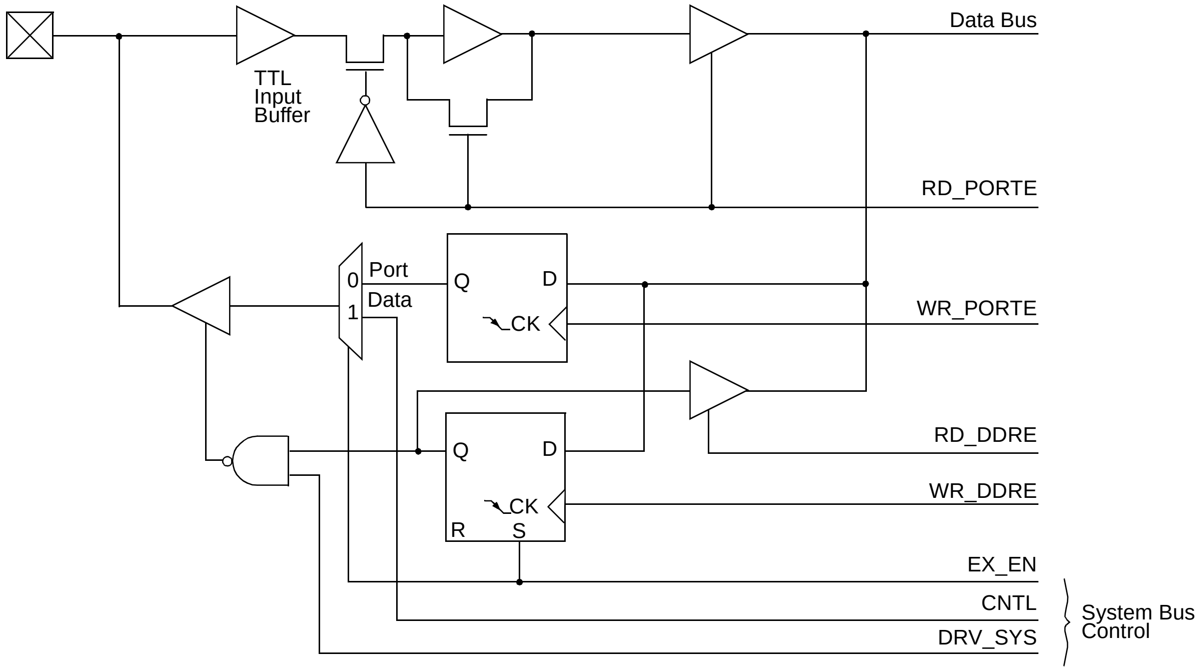
> **Note:** Three pins of this port are configured as the system bus when the device's configuration bits are selected to Microprocessor or Extended Microcontroller modes. The other pin is a general purpose I/O or Capture4 pin. In the two other microcontroller modes, RE2:RE0 are general purpose I/O pins.

Example 10-5 shows an instruction sequence to initialize PORTE. 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 10-5: INITIALIZING PORTE

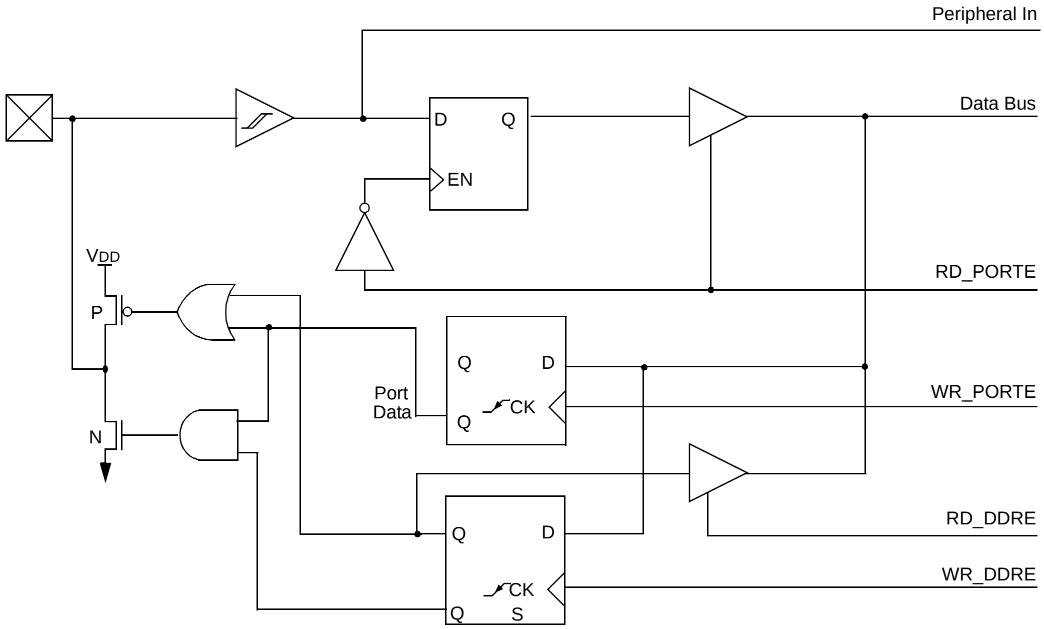
```
MOVLB  1          ; Select Bank 1
CLRF   PORTE, F   ; Initialize PORTE data
                  ; latches before setting
                  ; the data direction
                  ; register
MOVLW   0x03      ; Value used to initialize
                  ; data direction
MOVWF   DDRE      ; Set RE<1:0> as inputs
                  ; RE<3:2> as outputs
                  ; RE<7:4> are always
                  ; read as '0'
```

FIGURE 10-11: BLOCK DIAGRAM OF RE2:RE0 (IN I/O PORT MODE)



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

FIGURE 10-12: BLOCK DIAGRAM OF RE3/CAP4 PORT PIN



Note: I/O pin has protection diodes to V_{DD} and V_{SS}.

TABLE 10-9: PORTE FUNCTIONS

Name	Bit	Buffer Type	Function
RE0/$\overline{\text{ALE}}$	bit0	TTL	Input/output or system bus Address Latch Enable ($\overline{\text{ALE}}$) control pin.
RE1/$\overline{\text{OE}}$	bit1	TTL	Input/output or system bus Output Enable ($\overline{\text{OE}}$) control pin.
RE2/$\overline{\text{WR}}$	bit2	TTL	Input/output or system bus Write ($\overline{\text{WR}}$) control pin.
RE3/CAP4	bit3	ST	Input/output or Capture4 input pin.

Legend: TTL = TTL input, ST = Schmitt Trigger input

TABLE 10-10: REGISTERS/BITS ASSOCIATED WITH PORTE

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
15h, Bank 1	PORTE	—	—	—	—	RE3/CAP4	RE2/$\overline{\text{WR}}$	RE1/$\overline{\text{OE}}$	RE0/ALE	`---- xxxx`	`---- uuuu`
14h, Bank 1	DDRE	Data Direction Register for PORTE								`---- 1111`	`---- 1111`
14h, Bank 7	CA4L	Capture4 Low Byte								`xxxx xxxx`	`uuuu uuuu`
15h, Bank 7	CA4H	Capture4 High Byte								`xxxx xxxx`	`uuuu uuuu`
16h, Bank 7	TCON3	—	CA4OVF	CA3OVF	CA4ED1	CA4ED0	CA3ED1	CA3ED0	PWM3ON	`-000 0000`	`-000 0000`

Legend: `x` = unknown, `u` = unchanged, `-` = unimplemented, read as '0'. Shaded cells are not used by PORTE.

10.8 PORTH and DDRH Registers (PIC17C76X only)

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

The upper four bits of PORTH are multiplexed with 4 channels of the 10-bit A/D converter.

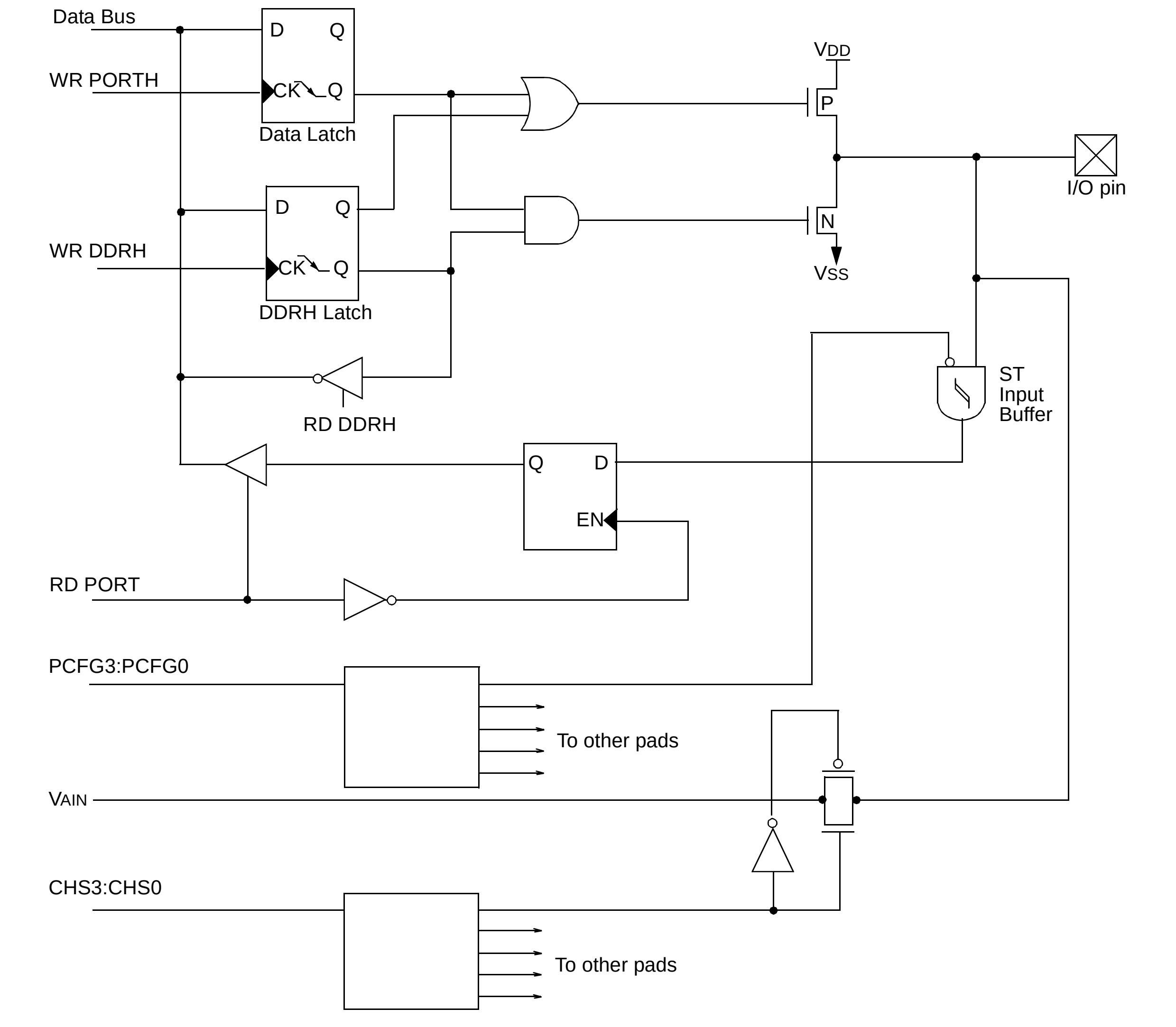
The remaining bits of PORTH are general purpose I/O.

Upon RESET, RH7:RH4 are automatically configured as analog inputs and must be configured in software to be a digital I/O.

EXAMPLE 10-8: INITIALIZING PORTH

```
MOVLB   8           ; Select Bank 8
MOVLW   0x0E        ; Configure PORTH as
MOVPF   ADCON1      ; digital
CLRF    PORTH, F    ; Initialize PORTH data
                    ;   latches before
                    ;   the data direction
                    ;   register
MOVLW    0x03       ; Value used to init
                    ;   data direction
MOVWF    DDRH       ; Set RH<1:0> as inputs
                    ;   RH<7:2> as outputs
```

FIGURE 10-17: BLOCK DIAGRAM OF RH7:RH4



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

12.3 Read/Write Consideration for TMR0

Although TMR0 is a 16-bit timer/counter, only 8-bits at a time can be read or written during a single instruction cycle. Care must be taken during any read or write.

12.3.1 READING 16-BIT VALUE

The problem in reading the entire 16-bit value is that after reading the low (or high) byte, its value may change from FFh to 00h.

Example 12-1 shows a 16-bit read. To ensure a proper read, interrupts must be disabled during this routine.

EXAMPLE 12-1: 16-BIT READ

```
MOVPF   TMR0L, TMPLO    ;read low tmr0
MOVPF   TMR0H, TMPHI    ;read high tmr0
MOVFP   TMPLO, WREG     ;tmplo —> wreg
CPFSLT  TMR0L           ;tmr0l < wreg?
RETURN                  ;no then return
MOVPF   TMR0L, TMPLO    ;read low tmr0
MOVPF   TMR0H, TMPHI    ;read high tmr0
RETURN                  ;return
```

12.3.2 WRITING A 16-BIT VALUE TO TMR0

Since writing to either TMR0L or TMR0H will effectively inhibit increment of that half of the TMR0 in the next cycle (following write), but not inhibit increment of the other half, the user must write to TMR0L first and TMR0H second, in two consecutive instructions, as shown in Example 12-2. The interrupt must be disabled. Any write to either TMR0L or TMR0H clears the prescaler.

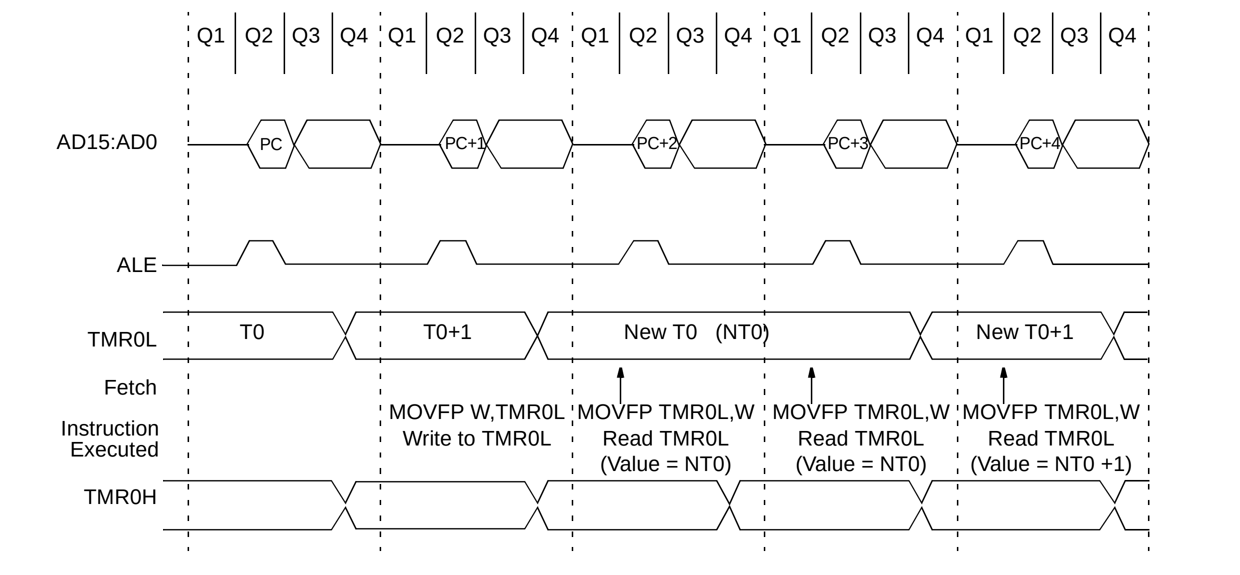
EXAMPLE 12-2: 16-BIT WRITE

```
BSF     CPUSTA, GLINTD ; Disable interrupts
MOVFP   RAM_L, TMR0L   ;
MOVFP   RAM_H, TMR0H   ;
BCF     CPUSTA, GLINTD ; Done, enable
                       ;    interrupts
```

12.4 Prescaler Assignments

Timer0 has an 8-bit prescaler. The prescaler selection is fully under software control; i.e., it can be changed “on the fly” during program execution. Clearing the prescaler is recommended before changing its setting. The value of the prescaler is “unknown” and assigning a value that is less than the present value, makes it difficult to take this unknown time into account.

FIGURE 12-3: TMR0 TIMING: WRITE HIGH OR LOW BYTE



13.2.2 FOUR CAPTURE MODE

This mode is selected by setting bit CA1/$\overline{\text{PR3}}$. A block diagram is shown in Figure 13-6. In this mode, TMR3 runs without a period register and increments from 0000h to FFFFh and rolls over to 0000h. The TMR3 interrupt Flag (TMR3IF) is set on this rollover. The TMR3IF bit must be cleared in software.

Registers PR3H/CA1H and PR3L/CA1L make a 16-bit capture register (Capture1). It captures events on pin RB0/CAP1. Capture mode is configured by the CA1ED1 and CA1ED0 bits. Capture1 Interrupt Flag bit (CA1IF) is set upon detection of the capture event. The corresponding interrupt mask bit is CA1IE. The Capture1 Overflow Status bit is CA1OVF.

All the captures operate in the same manner. Refer to Section 13.2.1 for the operation of capture.

FIGURE 13-6: TIMER3 WITH FOUR CAPTURES BLOCK DIAGRAM

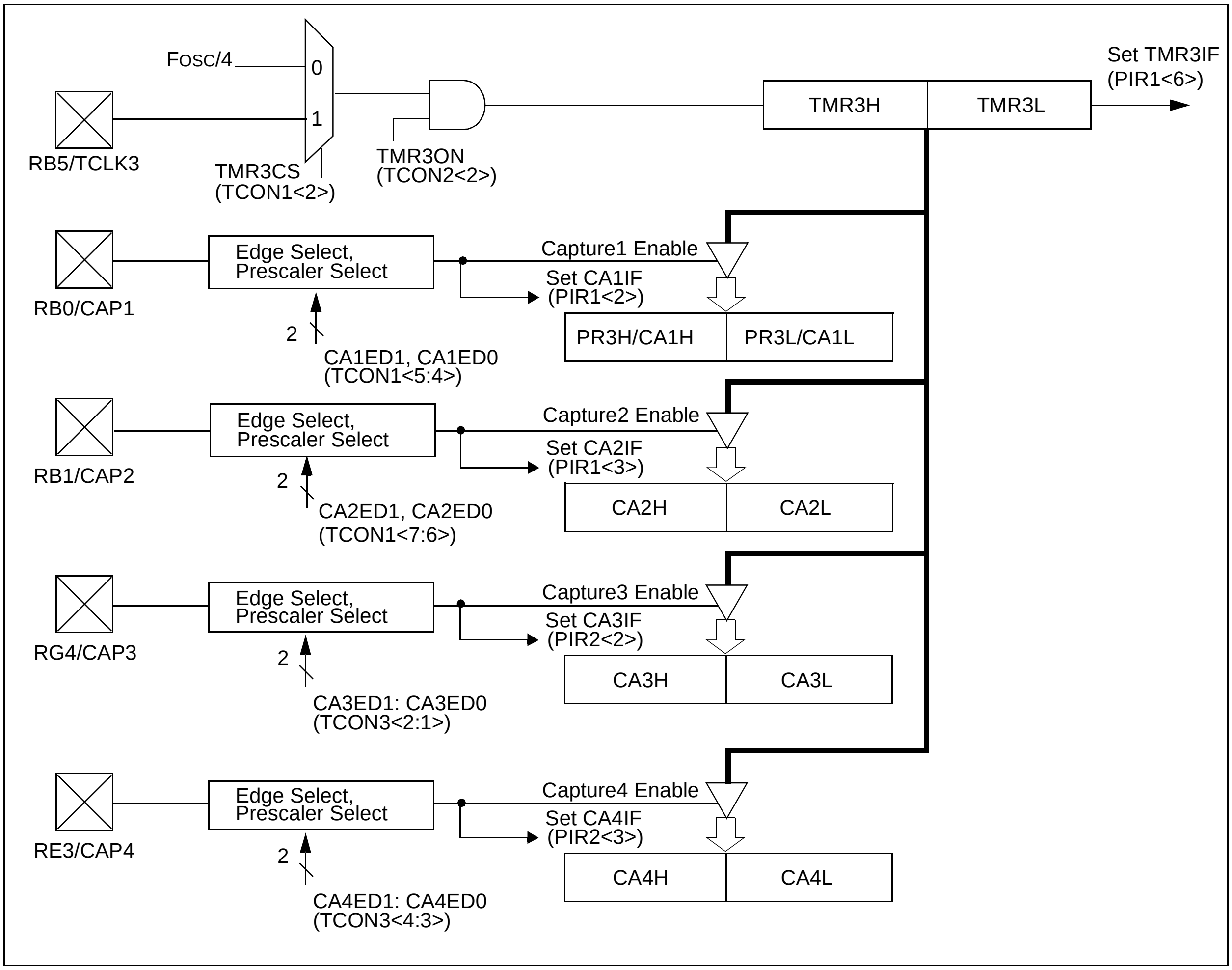
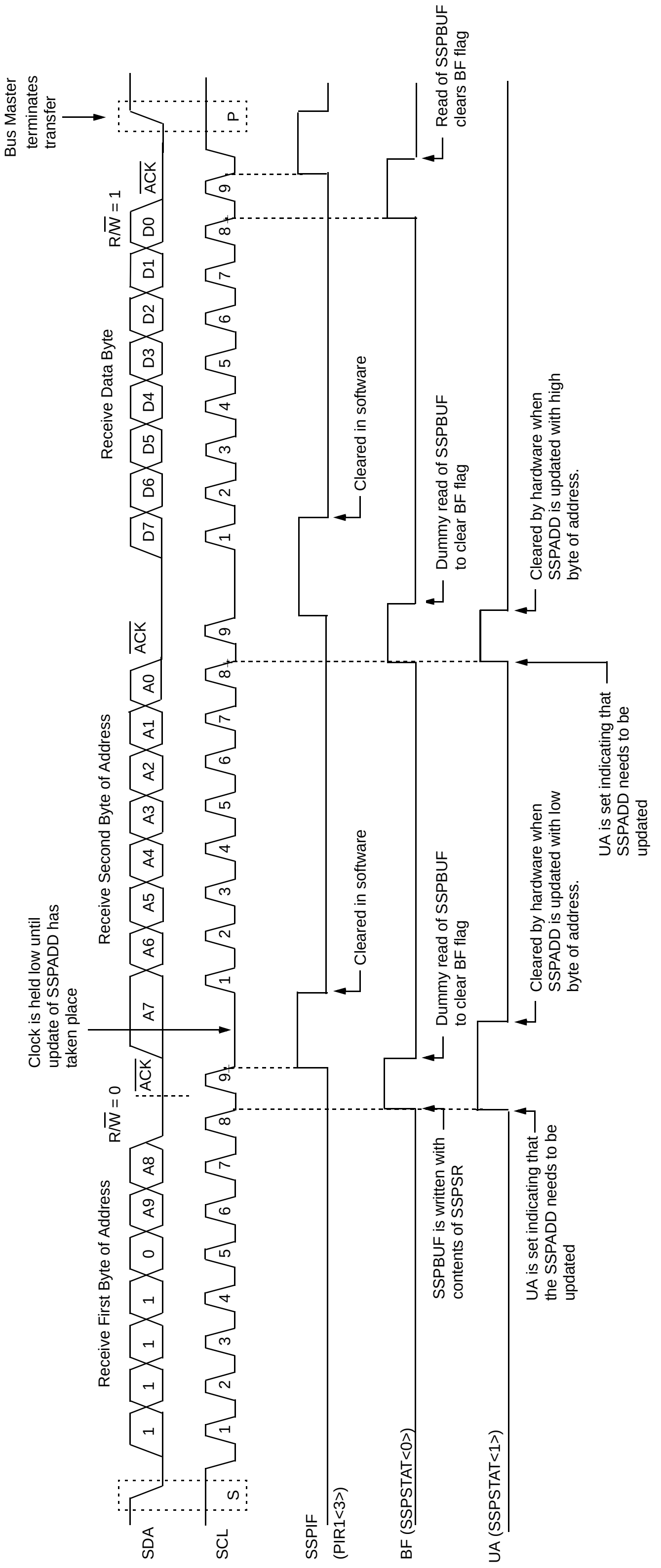


FIGURE 15-15: I^{2}C SLAVE-RECEIVER (10-BIT ADDRESS)


SDA
SCL
SSPIF
(PIR1<3>)
BF (SSPSTAT<0>)
UA (SSPSTAT<1>)
S
P
Receive First Byte of Address
1
1
1
1
0
A9
A8
R/$\overline{W}$ = 0
$\overline{ACK}$
1
2
3
4
5
6
7
8
9
Clock is held low until update of SSPADD has taken place
Receive Second Byte of Address
A7
A6
A5
A4
A3
A2
A1
A0
$\overline{ACK}$
Receive Data Byte
D7
D6
D5
D4
D3
D2
D1
D0
R/$\overline{W}$ = 1
$\overline{ACK}$
Bus Master terminates transfer
Cleared in software
Cleared in software
SSPBUF is written with contents of SSPSR
Dummy read of SSPBUF to clear BF flag
Dummy read of SSPBUF to clear BF flag
Read of SSPBUF clears BF flag
UA is set indicating that the SSPADD needs to be updated
Cleared by hardware when SSPADD is updated with low byte of address.
UA is set indicating that SSPADD needs to be updated
Cleared by hardware when SSPADD is updated with high byte of address.

15.2.5 MASTER MODE

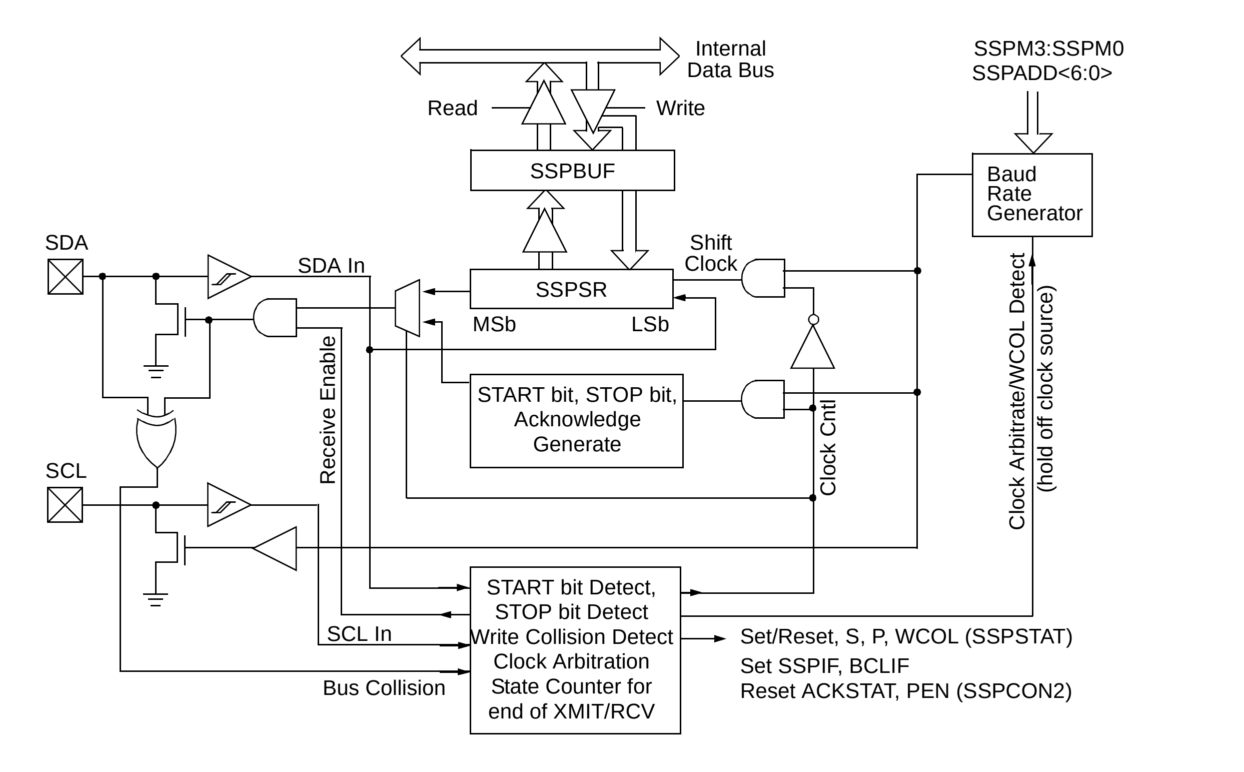
Master mode of operation is supported by interrupt generation on the detection of the START and STOP conditions. The STOP (P) and START (S) bits are cleared from a RESET, or when the MSSP module is disabled. Control of the I^{2}C bus may be taken when the P bit is set, or the bus is idle, with both the S and P bits clear.

In Master mode, the SCL and SDA lines are manipulated by the MSSP hardware.

The following events will cause SSP Interrupt Flag bit, SSPIF, to be set (SSP Interrupt if enabled):

- START condition
- STOP condition
- Data transfer byte transmitted/received
- Acknowledge transmit
- Repeated Start

FIGURE 15-17: SSP BLOCK DIAGRAM (I^{2}C MASTER MODE)



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:

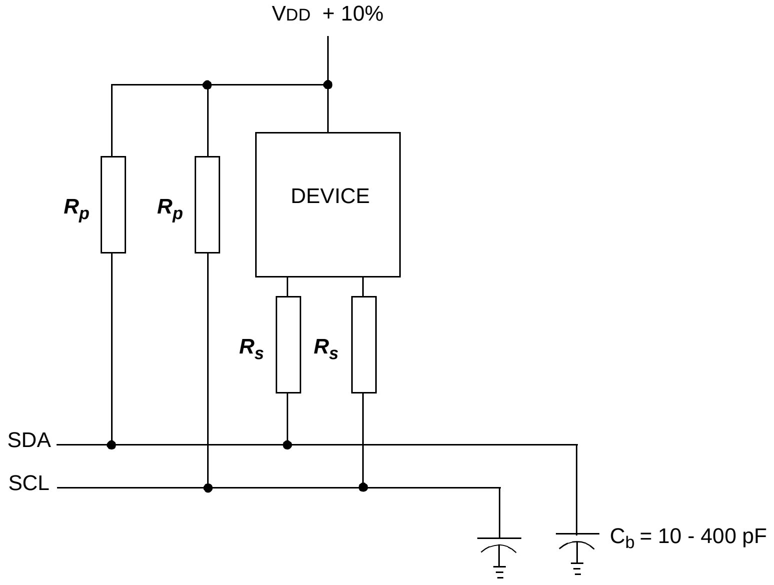
- Supply voltage
- Bus capacitance
- Number of connected devices (input current + leakage current)

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 $\underline{+}$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.

MOVPF Move p to f

Syntax: [*label*] MOVPF p,f

Operands: $0 \le f \le 255$
$0 \le p \le 31$

Operation: (p) → (f)

Status Affected: Z

Encoding:

`010p`	`pppp`	`ffff`	`ffff`

Description: Move data from data memory location 'p' to data memory location 'f'. Location 'f' can be anywhere in the 256 byte data space (00h to FFh), while 'p' can be 00h to 1Fh.

Either 'p' or 'f' can be WREG (a useful, special situation).

`MOVPF` is particularly useful for transferring a peripheral register (e.g. the timer or an I/O port) to a data memory location. Both 'f' and 'p' can be indirectly addressed.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example: `MOVPF REG1, REG2`

Before Instruction
REG1 = 0x11
REG2 = 0x33

After Instruction
REG1 = 0x11
REG2 = 0x11

MOVWF Move WREG to f

Syntax: [*label*] MOVWF f

Operands: $0 \le f \le 255$

Operation: (WREG) → (f)

Status Affected: None

Encoding:

`0000`	`0001`	`ffff`	`ffff`

Description: Move data from WREG to register 'f'. Location 'f' can be anywhere in the 256 byte data space.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example: `MOVWF REG`

Before Instruction
WREG = 0x4F
REG = 0xFF

After Instruction
WREG = 0x4F
REG = 0x4F

SWAPF Swap f

Syntax: [*label*] SWAPF f,d

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

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

Example: `SWAPF REG, 0`

Before Instruction
REG = 0x53

After Instruction
REG = 0x35

TABLRD Table Read

Syntax: [*label*] TABLRD t,i,f

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

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)

FIGURE 20-1: PIC17C7XX-33 VOLTAGE-FREQUENCY GRAPH

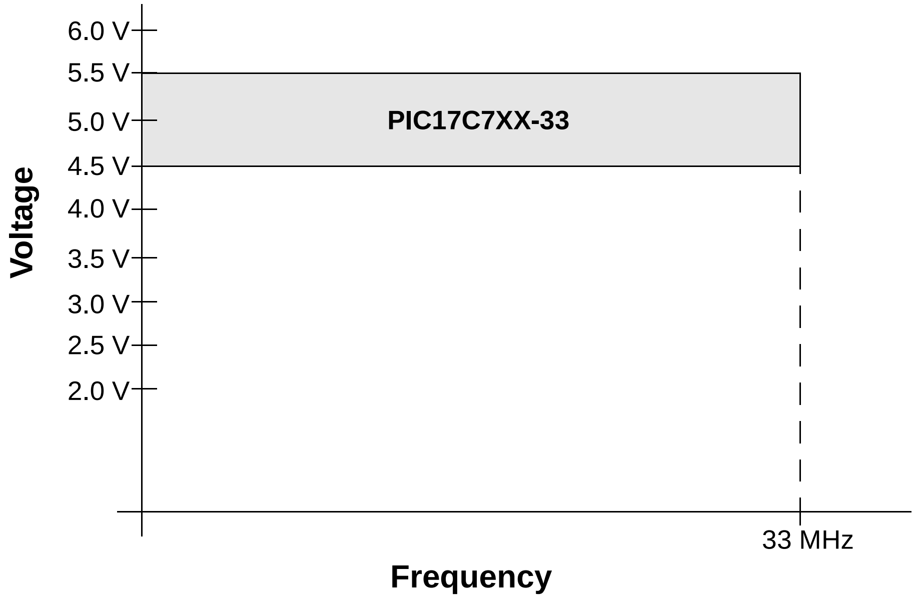


FIGURE 20-2: PIC17C7XX-16 VOLTAGE-FREQUENCY GRAPH

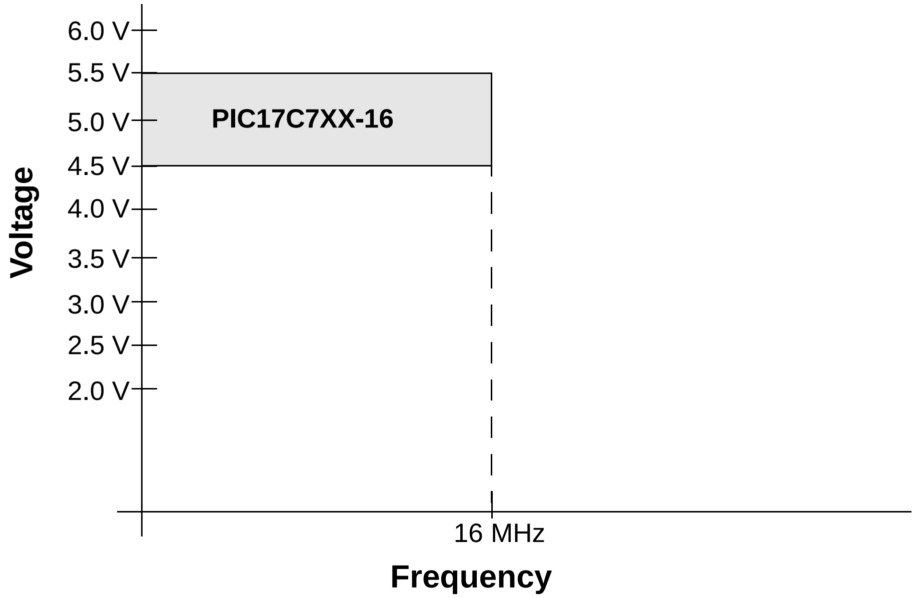
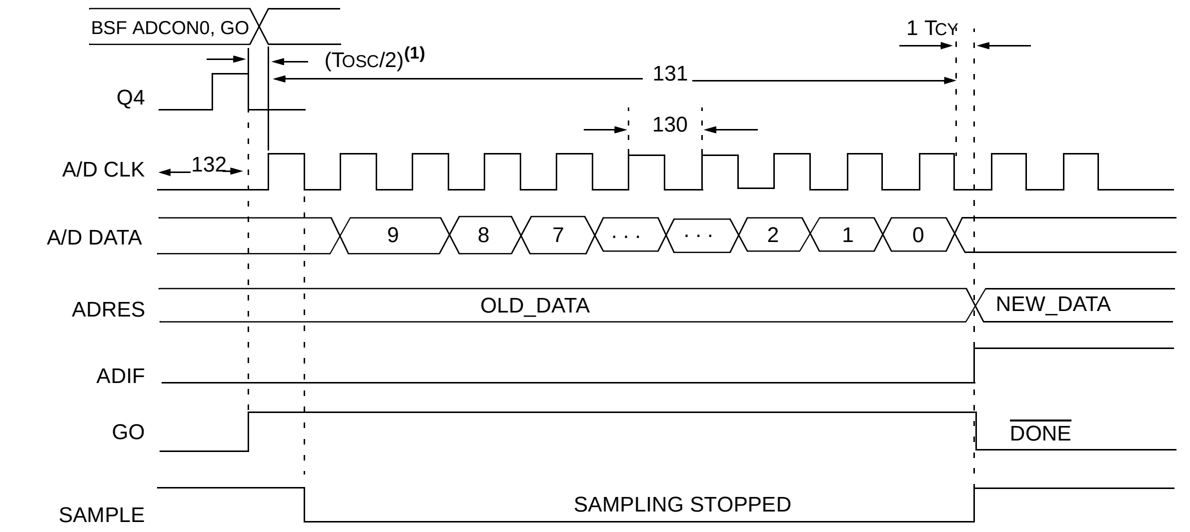


FIGURE 20-23: A/D CONVERSION TIMING



Note 1: If the A/D clock source is selected as RC, a time of TCY is added before the A/D clock starts. This allows the `SLEEP` instruction to be executed.

TABLE 20-19: A/D CONVERSION REQUIREMENTS

Param. No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
130	TAD	A/D clock period	PIC17**C**XXX	1.6	—	—	µs	TOSC based, VREF ≥ 3.0V
			PIC17**LC**XXX	3.0	—	—	µs	TOSC based, VREF full range
			PIC17**C**XXX	2.0	4.0	6.0	µs	A/D RC mode
			PIC17**LC**XXX	3.0	6.0	9.0	µs	A/D RC mode
131	TCNV	Conversion time (not including acquisition time) **(Note 1)**		11	—	12	Tad	
132	TACQ	Acquisition time		**(Note 2)** 10	20 —	— —	µs µs	 The minimum time is the amplifier settling time. This may be used if the “new” input voltage has not changed by more than 1LSb (i.e., 5 mV @ 5.12V) from the last sampled voltage (as stated on CHOLD).
134	TGO	Q4 to ADCLK start		—	Tosc/2	—	—	If the A/D clock source is selected as RC, a time of TCY is added before the A/D clock starts. This allows the `SLEEP` instruction to be executed.

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

Note 1: ADRES register may be read on the following TCY cycle.

2: See Section 16.1 for minimum conditions when input voltage has changed more than 1 LSb.

FIGURE 21-17: TYPICAL, MINIMUM AND MAXIMUM VOH vs. IOH (VDD = 5V, -40°C TO +125°C)

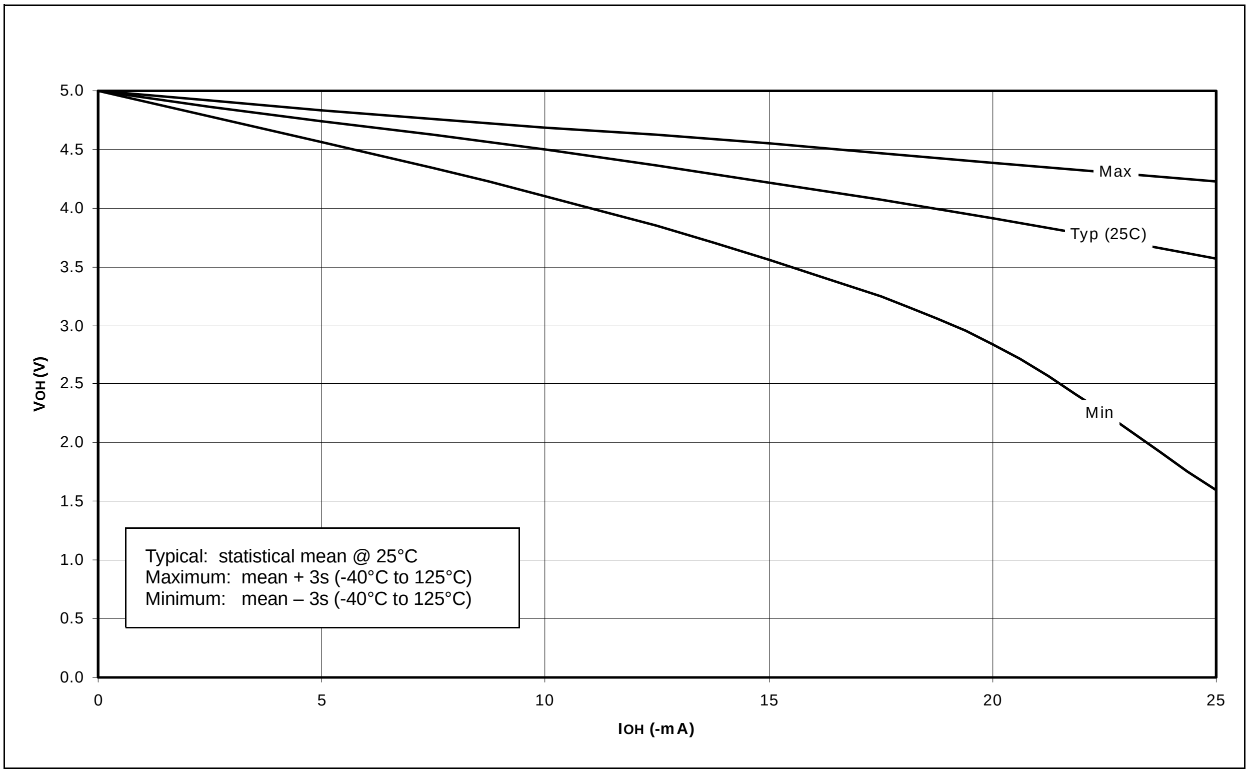
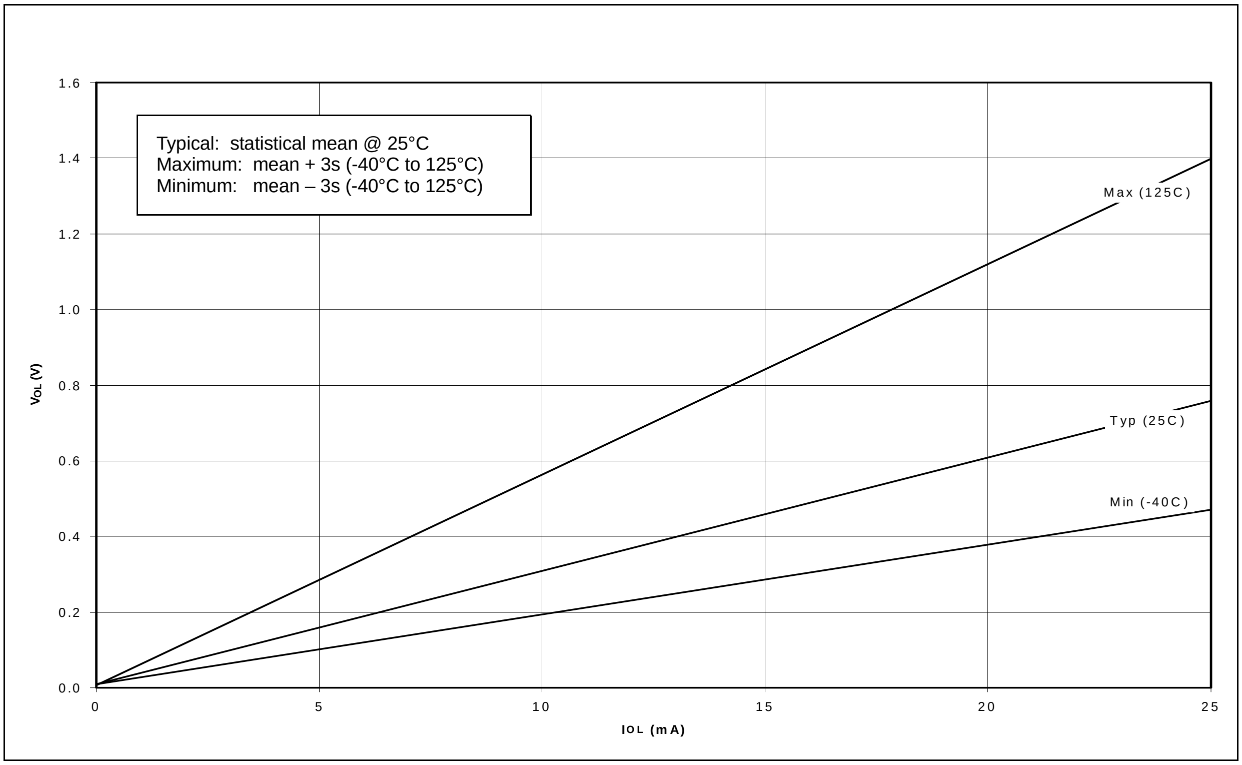


FIGURE 21-18: TYPICAL, MINIMUM AND MAXIMUM VOL vs. IOL (VDD = 5V, -40°C TO +125°C)


VOL (V)
IOL (mA)
Max (125C)
Typ (25C)
Min (-40C)
Typical: statistical mean @ 25°C
Maximum: mean + 3s (-40°C to 125°C)
Minimum: mean – 3s (-40°C to 125°C)