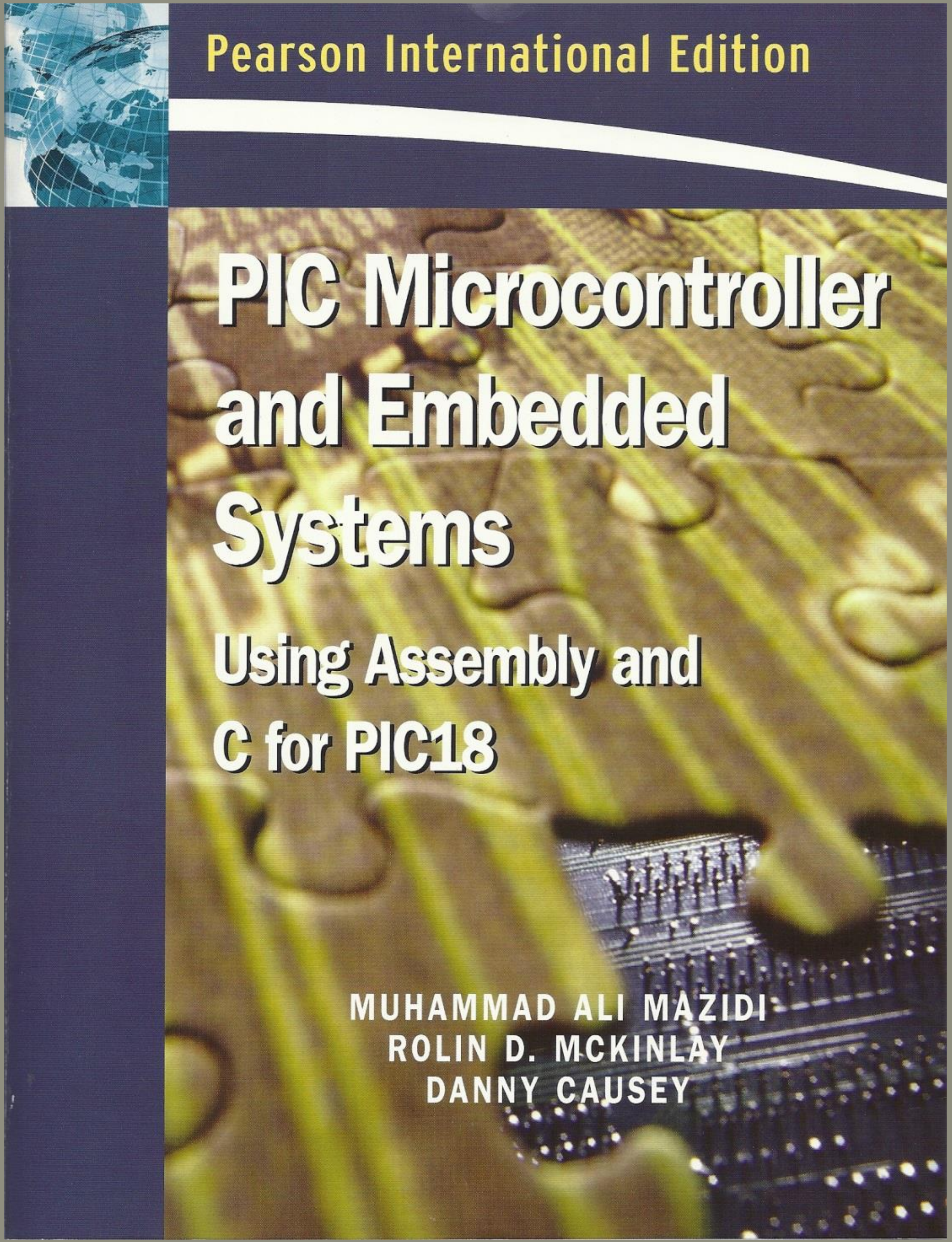
Chapter 4

PIC I/O Port Programming


Pearson International Edition
PIC Microcontroller and Embedded Systems
Using Assembly and C for PIC18
MUHAMMAD ALI MAZIDI
ROLIN D. MCKINLAY
DANNY CAUSEY

I/O Port Pins

- PIC18 has five ports; PORTA, PORTB, PORTC, PORTD, and PORTB.
- Each port has three SFRs associated with it; PORTx, TRISx (Tristate), and LATx (Latch).

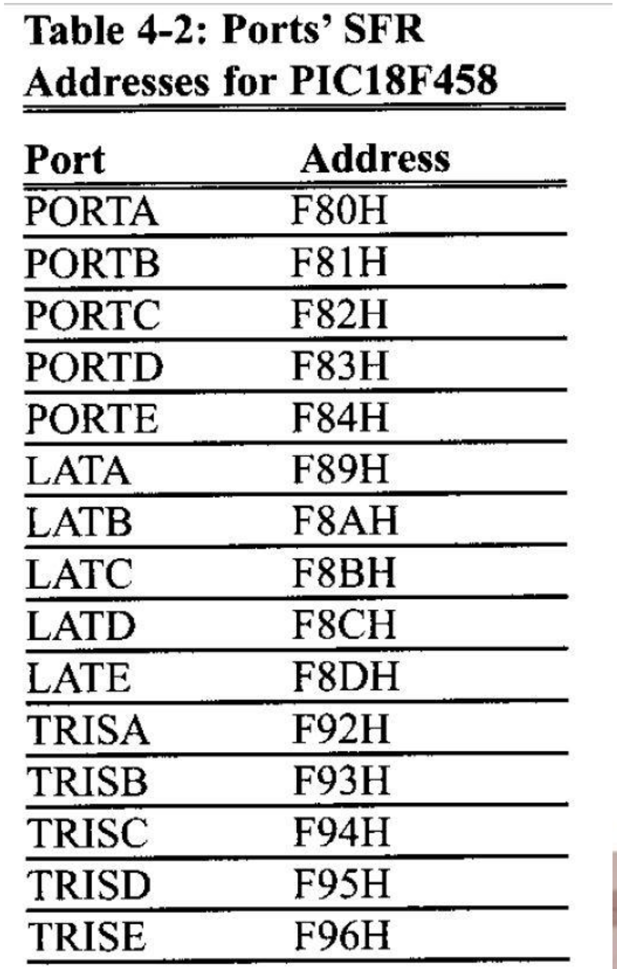
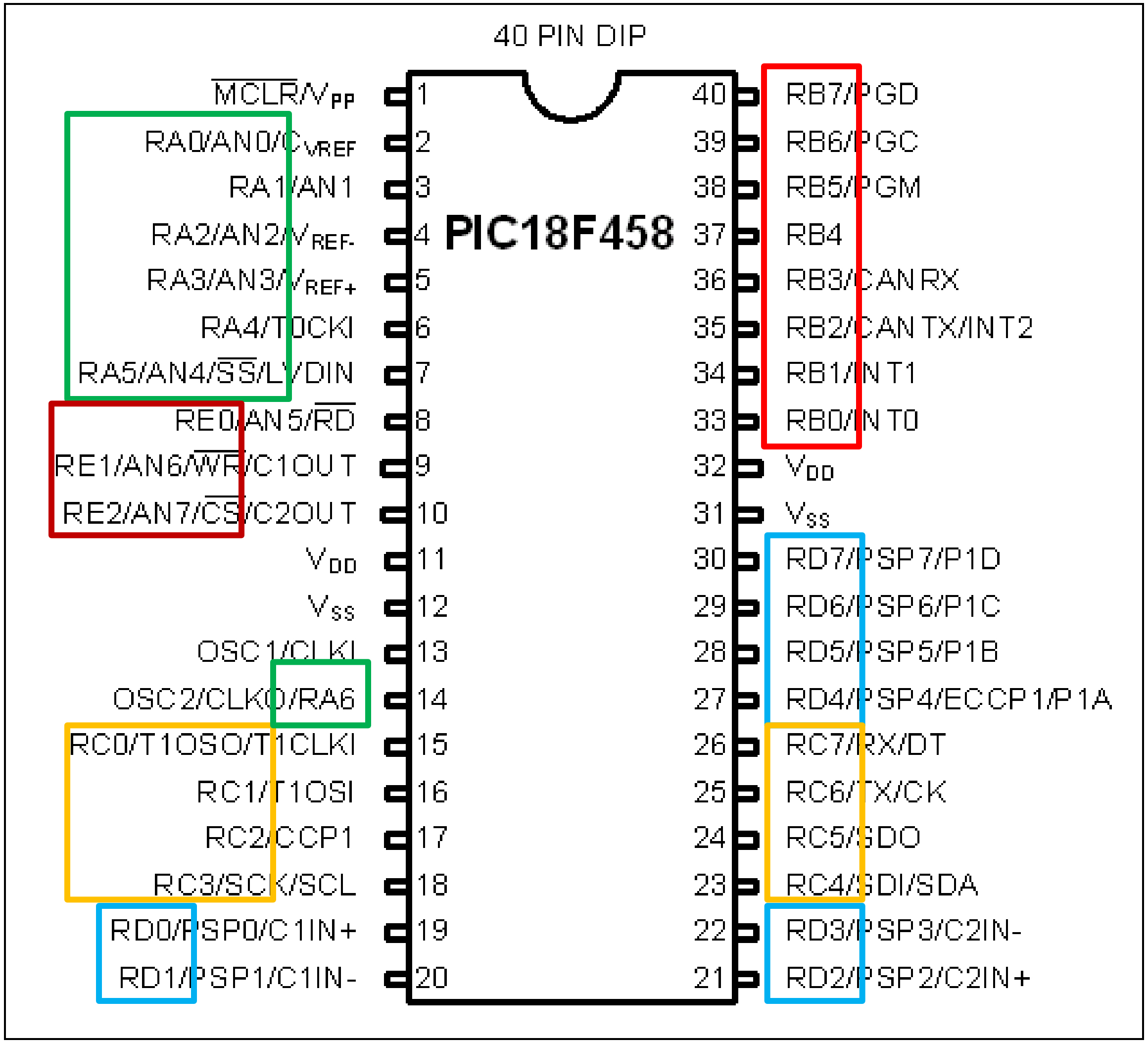


Table 4-2: Ports' SFR Addresses for PIC18F458

Port	Address
PORTA	F80H
PORTB	F81H
PORTC	F82H
PORTD	F83H
PORTE	F84H
LATA	F89H
LATB	F8AH
LATC	F8BH
LATD	F8CH
LATE	F8DH
TRISA	F92H
TRISB	F93H
TRISC	F94H
TRISD	F95H
TRISE	F96H

Figure 4-1. PICF458 Pin Diagram



TRIS Register Role

- The TRISx SFR is used solely for the purpose of making a given port an input or output port.
- The data will not go from the port register to the pins of the PIC unless we activate the TRIS bit (set it to zero)

TRIS Register Role

The following code will toggle all 8 bits of Port B.

```
        MOVLW   0X0
        MOVWF   TRISB
L1      MOVLW   0X55
        MOVWF   PORTB
        CALL    DELAY
        MOVLW   0X55
        MOVWF   PORTB
        CALL    DELAY
        GOTO    L1
```

Figure 4-2. CMOS States for P and N Transistors

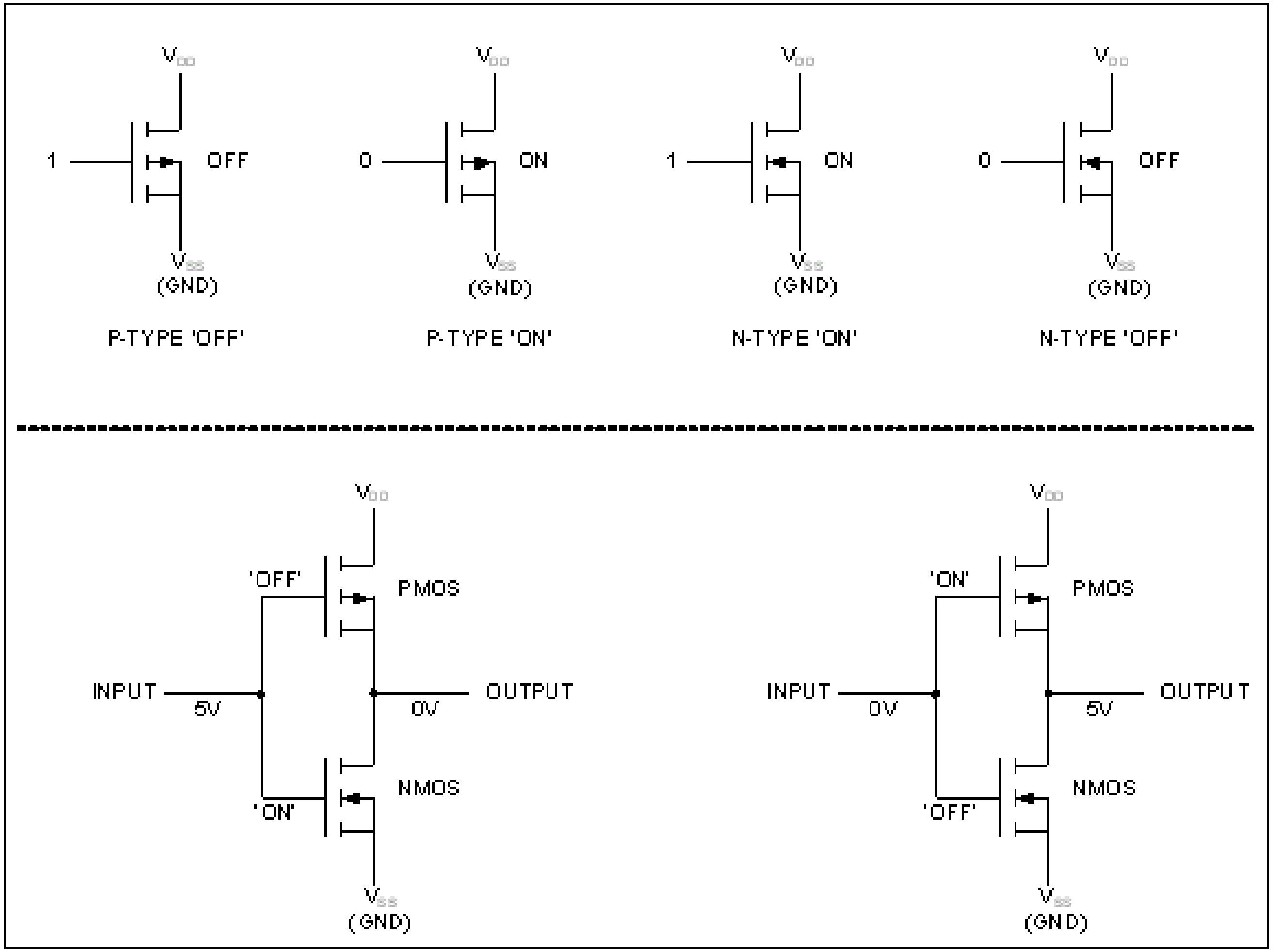


Figure 4-3. Outputting (Writing) 0 to a Pin in the PIC18

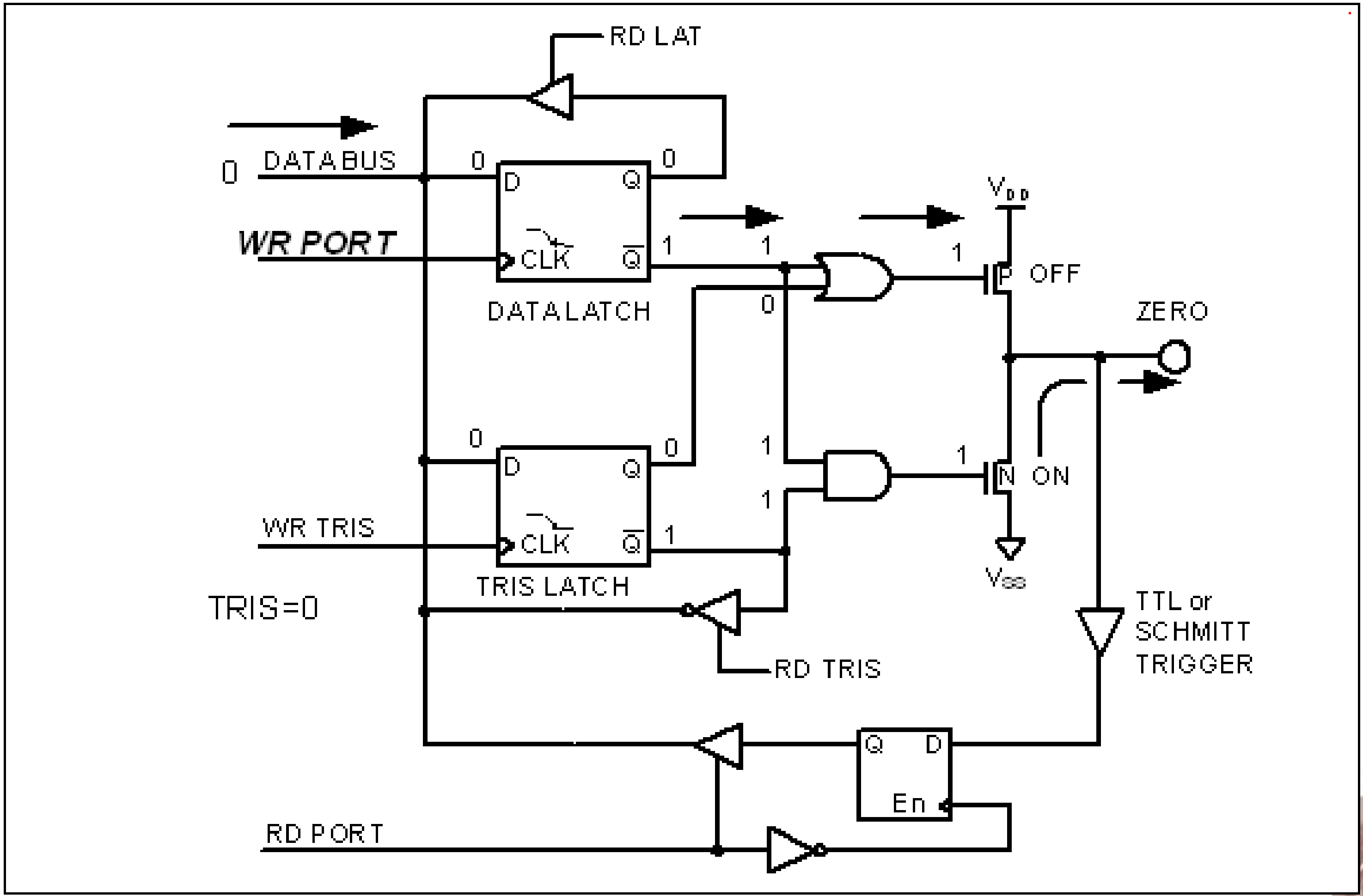
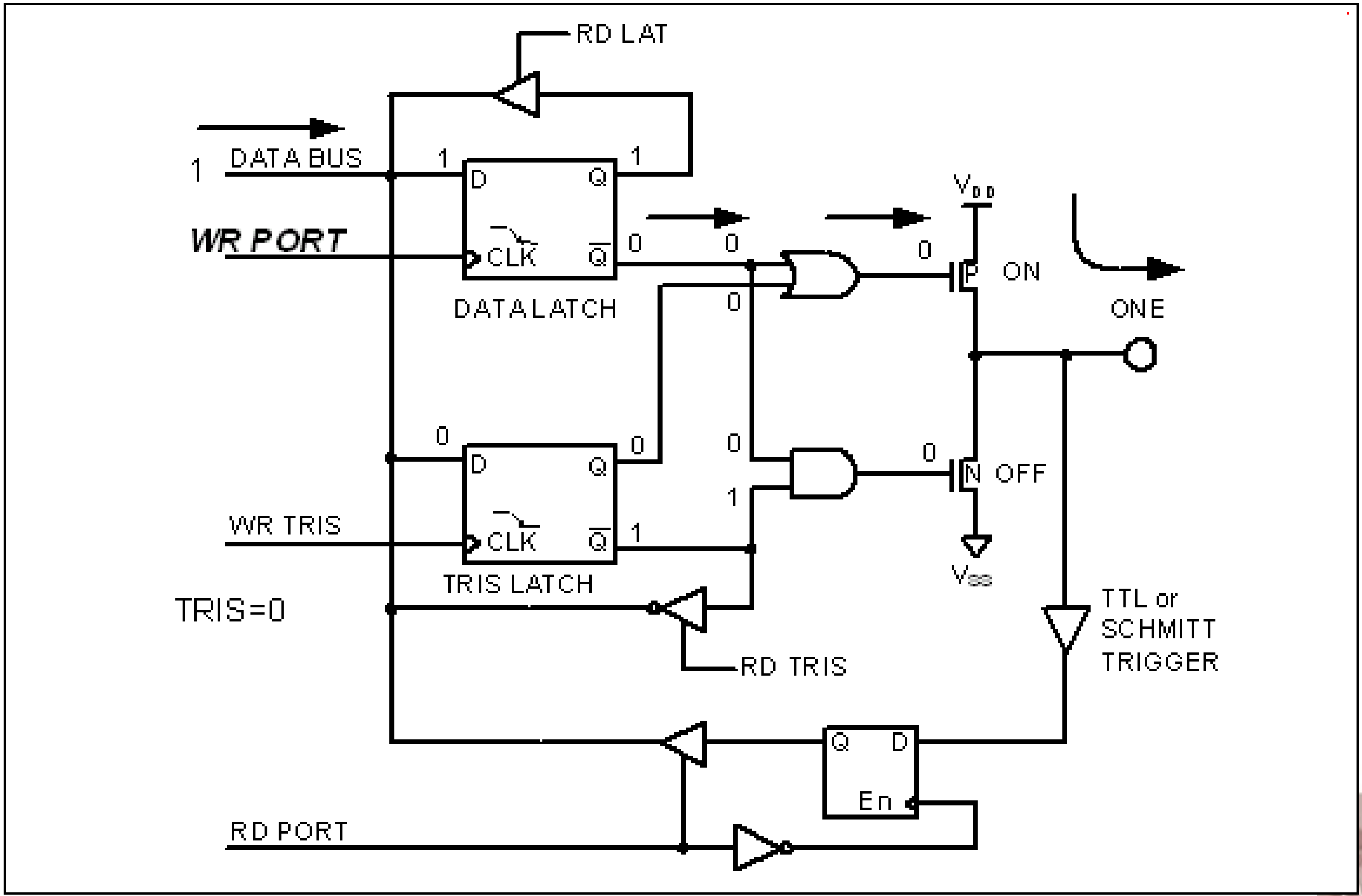


Figure 4-4. Outputting (Writing) 1 to a Pin in the PIC18



TRIS Register Role

- To make a port an input port, we must first put 1s into the TRISx register.
- Notice that 0 stands for out and 1 for in.

```
        CLRF     TRISB
        SETF     TRISC
L2      MOVF     PORTC, W
        ADDLW    5
        MOVWF    PORTB
        BRA      L2
```

Figure 4-5. Inputting (Reading) 0 from a Pin in the PIC18

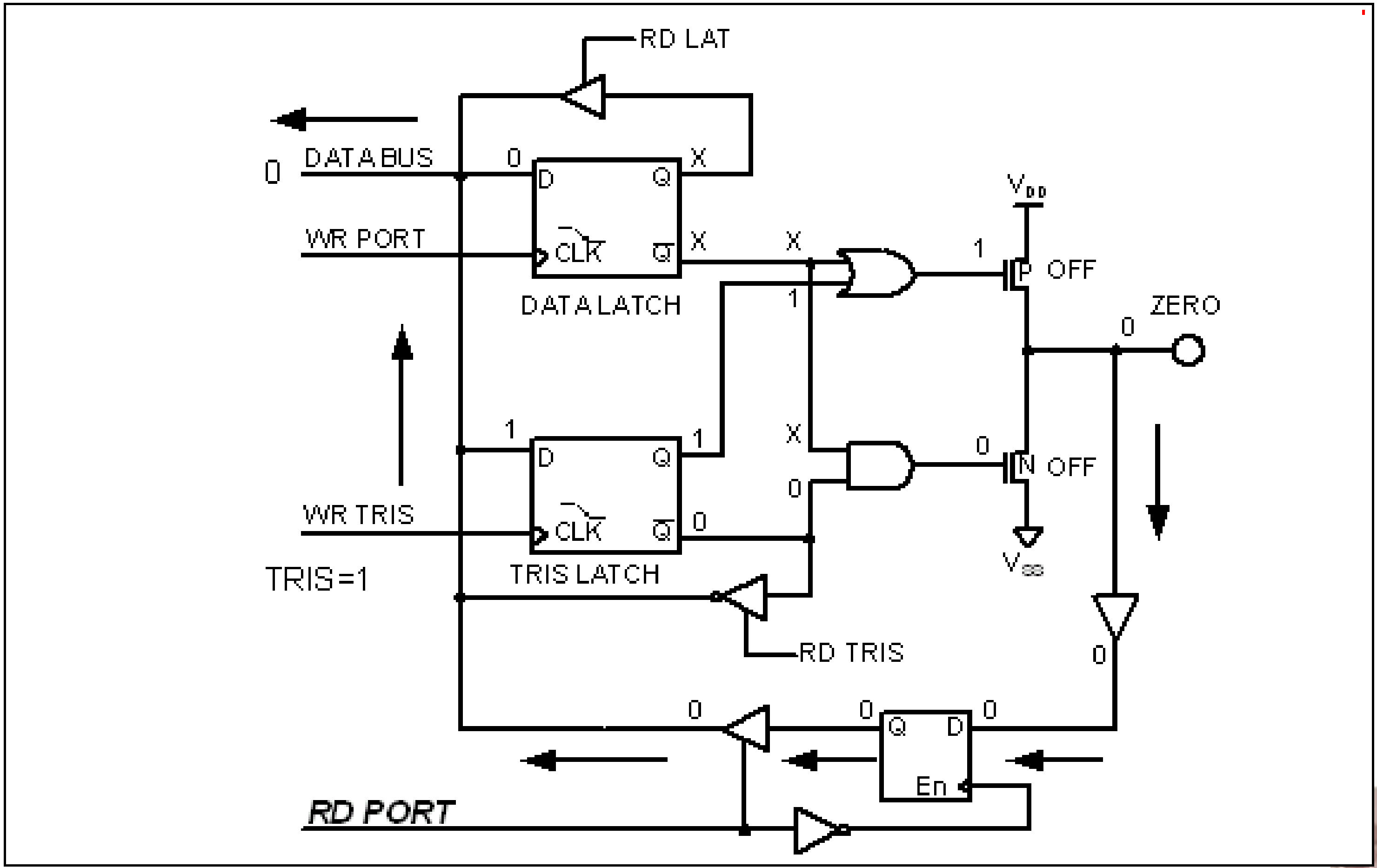
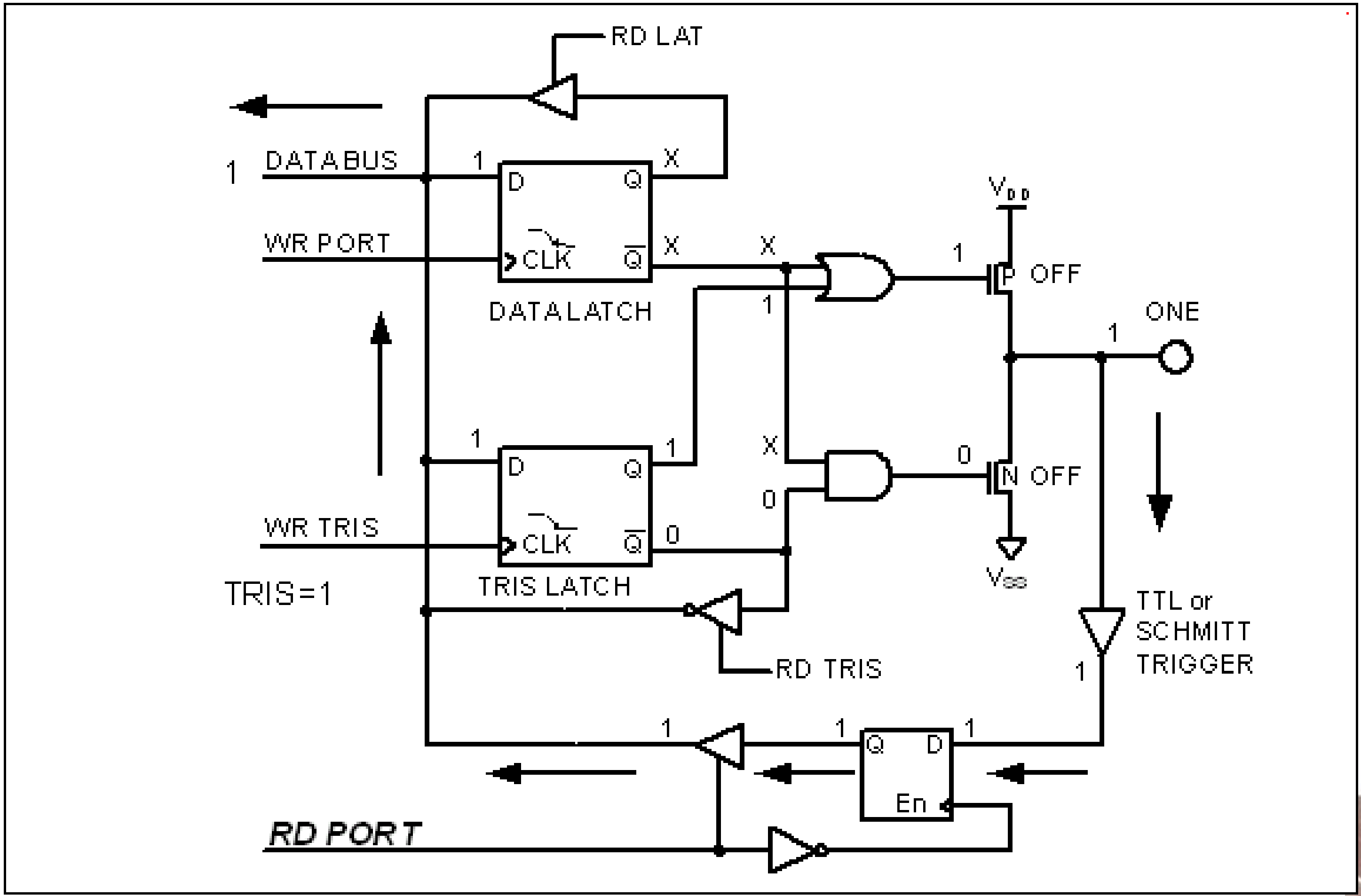


Figure 4-6. Inputting (Reading) 1 from a Pin in the PIC18



Dual Role of Ports A and B

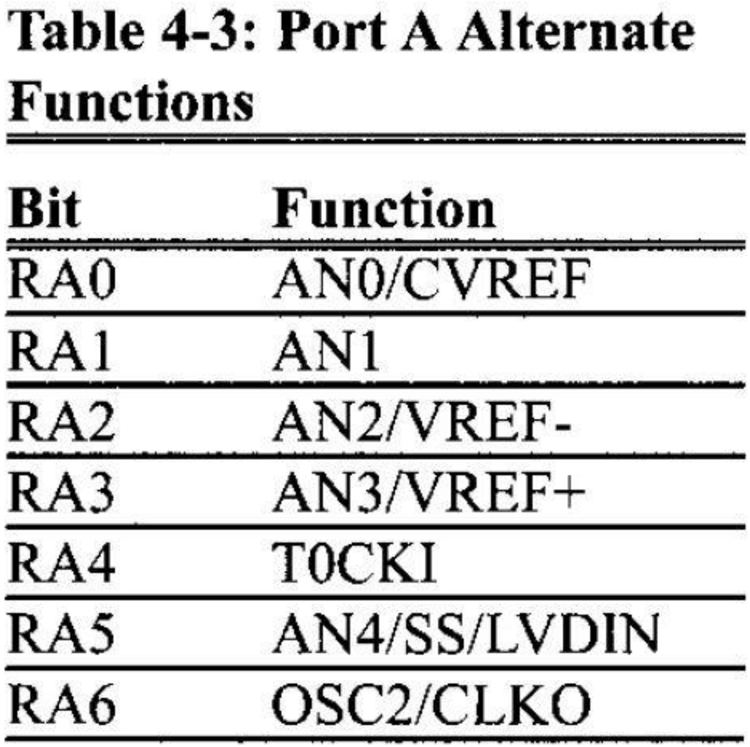


Table 4-3: Port A Alternate Functions

Bit	Function
RA0	AN0/CVREF
RA1	AN1
RA2	AN2/VREF-
RA3	AN3/VREF+
RA4	T0CKI
RA5	AN4/SS/LVDIN
RA6	OSC2/CLKO

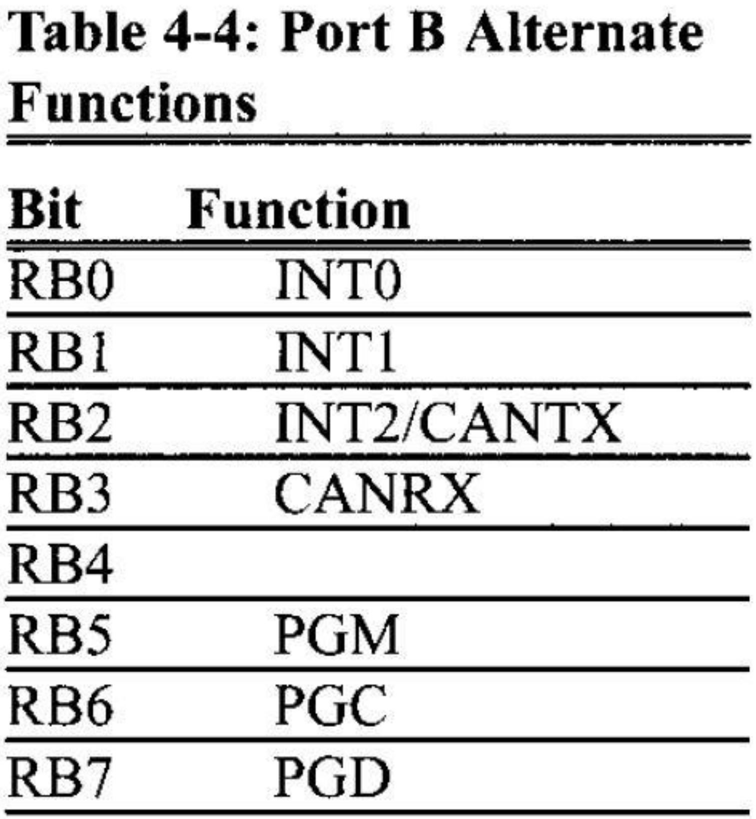


Table 4-4: Port B Alternate Functions

Bit	Function
RB0	INT0
RB1	INT1
RB2	INT2/CANTX
RB3	CANRX
RB4	
RB5	PGM
RB6	PGC
RB7	PGD

Dual Role of Ports C and D

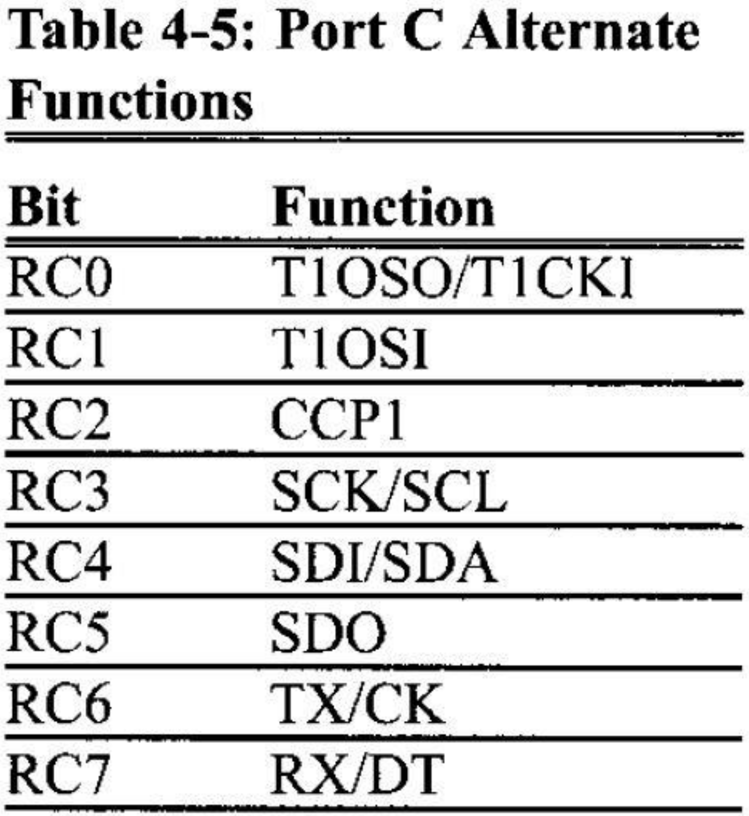


Table 4-5: Port C Alternate Functions

Bit	Function
RC0	T1OSO/T1CKI
RC1	T1OSI
RC2	CCP1
RC3	SCK/SCL
RC4	SDI/SDA
RC5	SDO
RC6	TX/CK
RC7	RX/DT

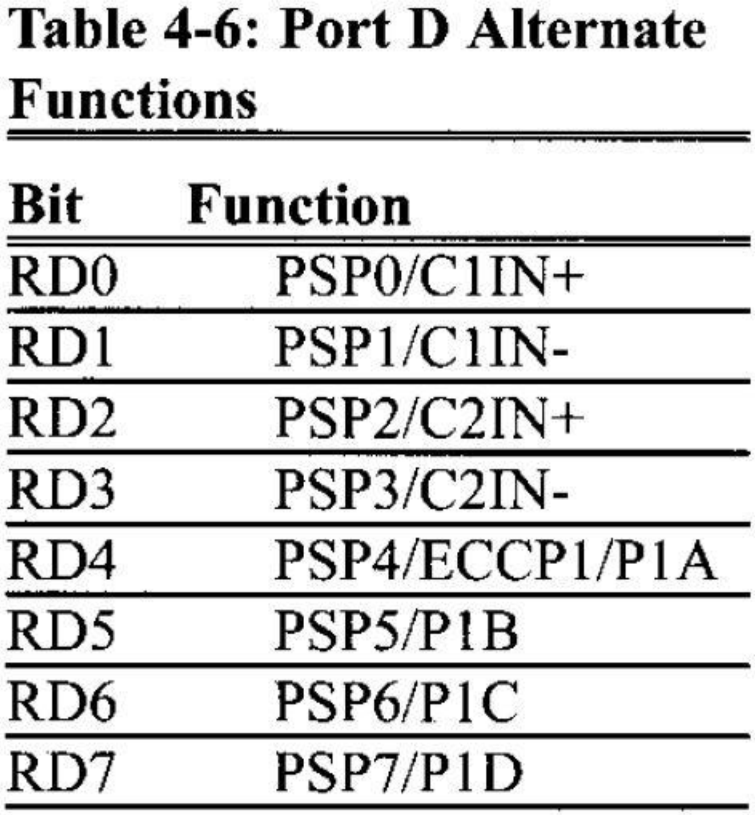


Table 4-6: Port D Alternate Functions

Bit	Function
RD0	PSP0/C1IN+
RD1	PSP1/C1IN-
RD2	PSP2/C2IN+
RD3	PSP3/C2IN-
RD4	PSP4/ECCP1/P1A
RD5	PSP5/P1B
RD6	PSP6/P1C
RD7	PSP7/P1D

Table 4-9: Single-Bit Addressability of Ports for PIC18F458/4580

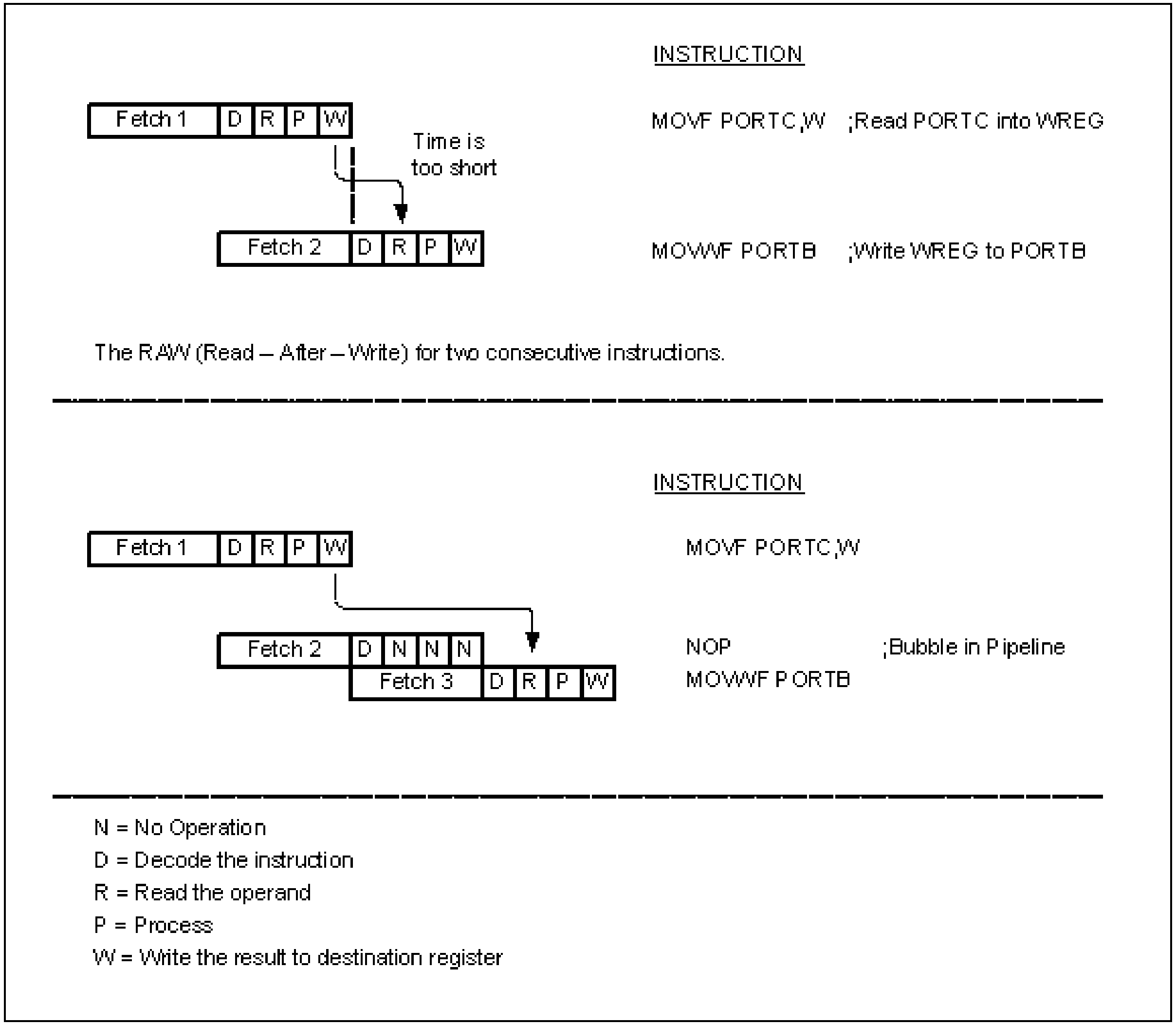
PORT	PORTB	PORTC	PORTD	PORTE	Port Bit
RA0	RB0	RC0	RD0	RE0	D0
RA1	RB1	RC1	RD1	RE1	D1
RA2	RB2	RC2	RD2	RE2	D2
RA3	RB3	RC3	RD3		D3
RA4	RB4	RC4	RD4		D4
RA5	RB5	RC5	RD5		D5
	RB6	RC6	RD6		D6
	RB7	RC7	RD7		D7

Read-After-Write (RAW) Dependency

- We need a NOP (or some other instruction) to make sure that the data is written into WREG before it is read for outputting to port B.

```
        CLRF      TRISB
        SETF      TRISC
L4      MOVF      PORTC, W
        NOP
        MOVWF     PORTB
        BRA       L4
```

Pipeline for Read Followed by Write I/O



INSTRUCTION

MOVF PORTC,W ;Read PORTC into WREG

MOVWF PORTB ;Write WREG to PORTB

The RAW (Read – After – Write) for two consecutive instructions.

INSTRUCTION

MOVF PORTC,W

NOP ;Bubble in Pipeline

MOVWF PORTB

N = No Operation
D = Decode the instruction
R = Read the operand
P = Process
W = Write the result to destination register

Example 4-1

Write a test program for the PIC18 chip to toggle all the bits of PORTB, PORTC, and PORTD every 1/4 of a second. Assume a crystal frequency of 4 MHz.

Solution:

```
;tested with MPLAB for the PIC18F458 and XTAL = 4 MHz

      list P=PIC18F458
#include P18F458.INC

R1 equ 0x07
R2 equ 0x08

ORG 0
      CLRF   TRISB          ;make Port B an output port
      CLRF   TRISC          ;make Port C an output port
      CLRF   TRISD          ;make Port D an output port
      MOVLW  0x55           ;WREG = 55h
      MOVWF  PORTB          ;put 55h on Port B pins
      MOVWF  PORTC          ;put 55h on Port C pins
      MOVWF  PORTD          ;put 55h on Port D pins
L3    COMF   PORTB,F        ;toggle bits of Port B
      COMF   PORTC,F        ;toggle bits of Port C
      COMF   PORTD,F        ;toggle bits of Port D
      CALL   QDELAY         ;quarter of a second delay
      BRA    L3
```

```
;-----------1/4 SECOND DELAY
QDELAY
        MOVLW D'200'
        MOVWF R1
D1      MOVLW D'250'
        MOVWF R2
D2      NOP
        NOP
        DECF R2, F
        BNZ D2
        DECF R1, F
        BNZ D1
        RETURN
        END
```

Calculations:

4 MHz / 4 = 1 MHz
1 / 1 MHz = 1 μs
Delay = 250 × 200 × 5 MC × 1 μs = 250,000 μs (if we include the overhead, we will have 250,800. See Example 3-17 in the previous chapter.)

Use the MPLAB simulator to verify the delay size.

I/O Ports and Bit-Addressability

- To access individual bits of the port without altering the rest of the bits in the port.

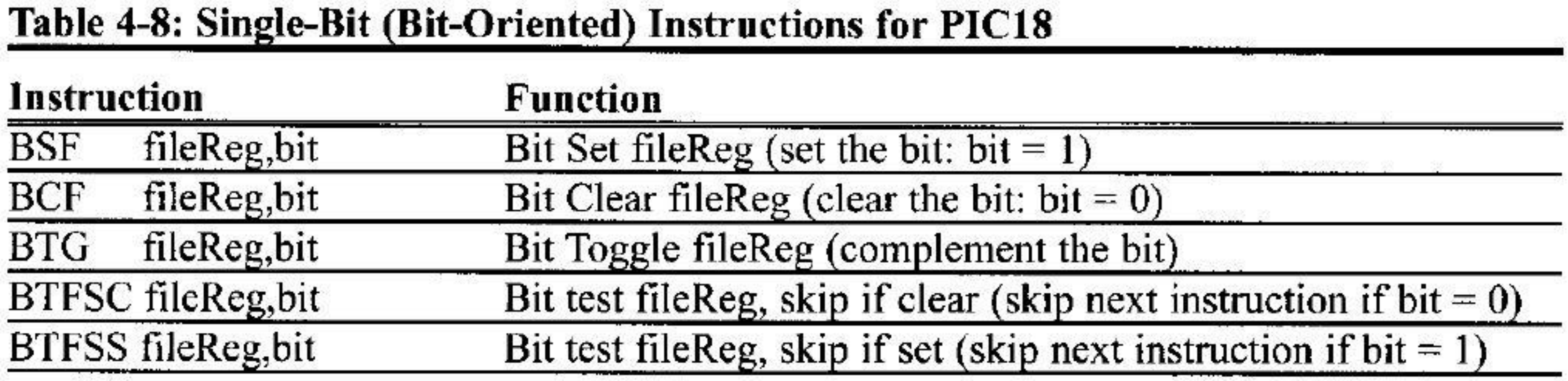


Table 4-8: Single-Bit (Bit-Oriented) Instructions for PIC18

Instruction	Function
BSF fileReg,bit	Bit Set fileReg (set the bit: bit = 1)
BCF fileReg,bit	Bit Clear fileReg (clear the bit: bit = 0)
BTG fileReg,bit	Bit Toggle fileReg (complement the bit)
BTFSC fileReg,bit	Bit test fileReg, skip if clear (skip next instruction if bit = 0)
BTFSS fileReg,bit	Bit test fileReg, skip if set (skip next instruction if bit = 1)

BSF and BCF instructions

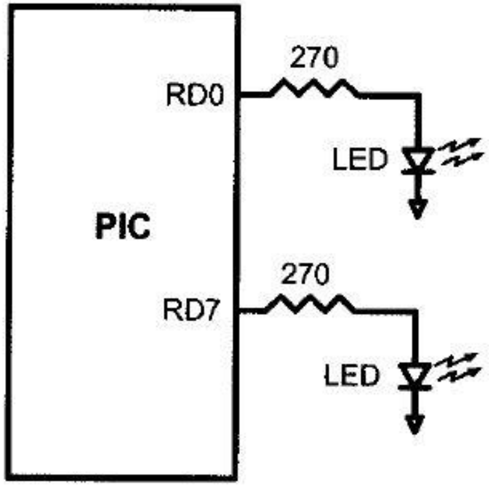
- BSF   fileReg, bit_num – to set high a single bit of a given fileReg.
- BCF   fileReg, bit_num – to clear a single bit of a given fileReg.

Example 4-2

An LED is connected to each pin of Port D. Write a program to turn on each LED from pin D0 to pin D7. Call a delay module before turning on the next LED.

Solution:

```
CLRF   TRISD          ;make PORTD an output port
BSF    PORTD,0        ;bit set turns on RD0
CALL   DELAY          ;delay before next one
BSF    PORTD,1        ;turn on RD1
CALL   DELAY          ;delay before next one
BSF    PORTD,2
CALL   DELAY
BSF    PORTD,3
CALL   DELAY
BSF    PORTD,4
CALL   DELAY
BSF    PORTD,5
CALL   DELAY
BSF    PORTD,6
CALL   DELAY
BSF    PORTD,7
CALL   DELAY
```



Example 4-3

Write the following programs:

(a) Create a square wave of 50% duty cycle on bit 0 of Port C.

(b) Create a square wave of 66% duty cycle on bit 3 of Port C.

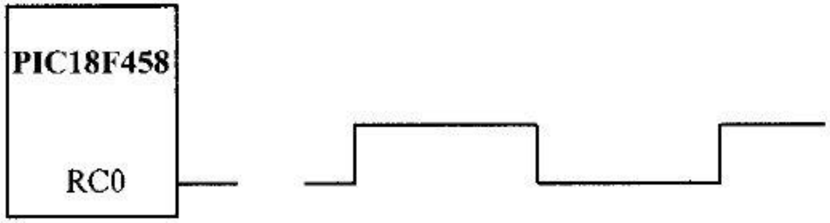
Solution:

(a) The 50% duty cycle means that the "on" and "off" states (or the high and low portions of the pulse) have the same length. Therefore, we toggle RC0 with a time delay between each state.

```
      BCF   TRISC,0      ;clear TRIS bit for RC0 = out
HERE  BSF   PORTC,0      ;set to HIGH RC0 (RC0 = 1)
      CALL  DELAY        ;call the delay subroutine
      BCF   PORTC,0      ;RC0 = 0
      CALL  DELAY
      BRA   HERE         ;keep doing it
```

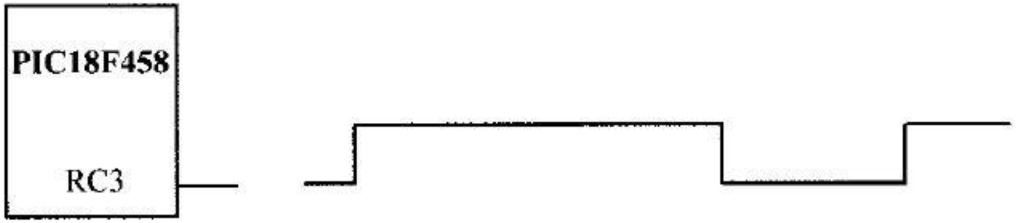
Another way to write the above program is:

```
      BCF   TRISC,0      ;make RC0 = out
HERE  BTG   PORTC,0      ;complement bit 0 of PORTC
      CALL  DELAY        ;call the delay subroutine
      BRA   HERE         ;keep doing it
```



(b) A 66% duty cycle means that the “on” state is twice the “off” state.

```
      BCF   TRISC,3      ;clear TRISC3 bit for output
BACK  BSF   PORTC,3      ;RC3 = 1
      CALL  DELAY        ;call the delay subroutine
      CALL  DELAY        ;twice for 66%
      BCF   PORTC,3      ;RC3 = 0
      CALL  DELAY        ;call delay once for 33%
      BRA   BACK         ;keep doing it
```



BTFSS and BTFSC instructions

- BTFSS fileReg, bit_num – to test the bit and skips the next instruction if it is HIGH.
- BTFSC fileReg, bit_num – to test the bit and skips the next instruction if it is LOW.

Example 4-4

Write a program to perform the following:

(a) Keep monitoring the RB2 bit until it becomes HIGH;

(b) When RB2 becomes HIGH, write value 45H to Port C, and also send a HIGH-to-LOW pulse to RD3.

Solution:

```
      BSF   TRISB,2      ;make RB2 an input
      CLRF  TRISC        ;make PORTC an output port
      BCF   PORTD,3      ;make RD3 an output
      MOVLW 0x45         ;WREG = 45h
AGAIN BTFSS PORTB,2      ;bit test RB2 for HIGH
      BRA   AGAIN        ;keep checking if LOW
      MOVWF PORTC        ;issue WREG to Port C
      BSF   PORTD,3      ;bit set fileReg RD3 (H-to-L)
      BCF   PORTD,3      ;bit clear fileReg RD3 (L)
```

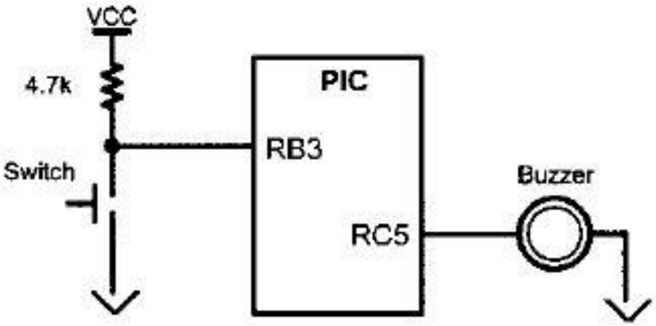
In this program, instruction "`BTFSS  PORTB,2`" stays in the loop as long as RB2 is LOW. When RB2 becomes HIGH, it skips the branch instruction to get out of the loop, and writes the value 45H to Port C. It also sends a HIGH-to-LOW pulse to RD3.

Example 4-5

Assume that bit RB3 is an input and represents the condition of a door alarm. If it goes LOW, it means that the door is open. Monitor the bit continuously. Whenever it goes LOW, send a HIGH-to-LOW pulse to port RC5 to turn on a buzzer.

Solution:

```
      BSF   TRISB,3      ;make RB3 an input
      BCF   TRISC,5      ;make RC5 an output
HERE  BTFSC PORTB,3      ;keep monitoring RB3 for HIGH
      BRA   HERE         ;stay in the loop
      BSF   PORTC,5      ;make RC5 HIGH
      BCF   PORTC,5      ;make RC5 LOW for H-to-L
      BRA   HERE
```



Example 4-6

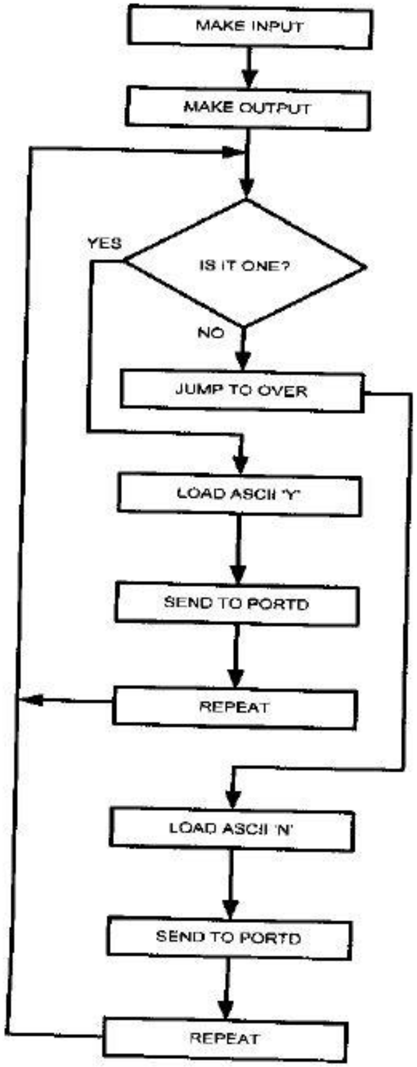
A switch is connected to pin RB2. Write a program to check the status of SW and perform the following:

(a) If SW = 0, send the letter 'N' to PORTD.

(b) If SW = 1, send the letter 'Y' to PORTD.

Solution:

```
      BSF   TRISB,2    ;make RB2 an input
      CLRF  TRISD      ;make PORTD an output port
AGAIN BTFSS PORTB,2    ;bit test RB2 for HIGH
      BRA   OVER       ;it must be LOW
      MOVLW A'Y'       ;WREG = 'Y' ASCII letter Y
      MOVWF PORTD      ;issue WREG to PORTD
      GOTO  AGAIN      ;we can use BRA too
OVER  MOVLW A'N'       ;WREG = 'N' ASCII  letter N
      MOVWF PORTD      ;issue WREG to PORTD
      GOTO  AGAIN      ;we can use BRA too
```



INSTRUCTIONS

```
          BSF TRISB, 2
          CLRF TRISD
AGAIN     BTFSS PORTB, 2
          BRA OVER
          MOVLW A'Y'
          MOVWF PORTD
          GOTO AGAIN
OVER      MOVLW A'N'
          MOVWF PORTD
          GOTO AGAIN
```

LATx Port

- Reading the status of the input.
- Reading the internal latch of the LAT register.

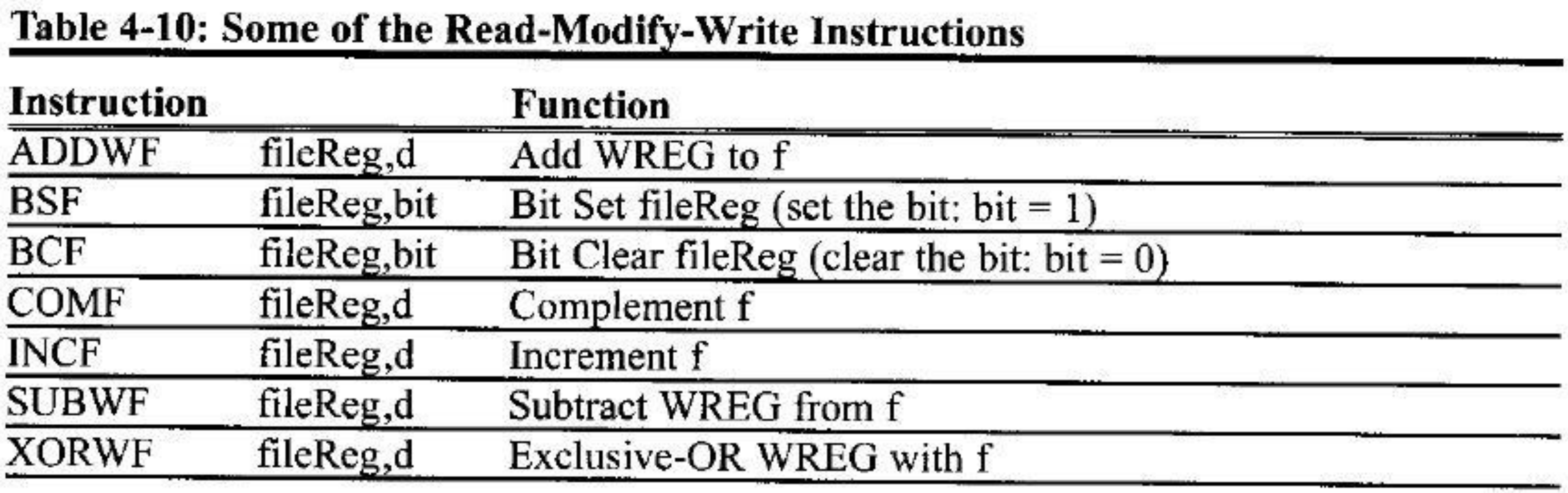
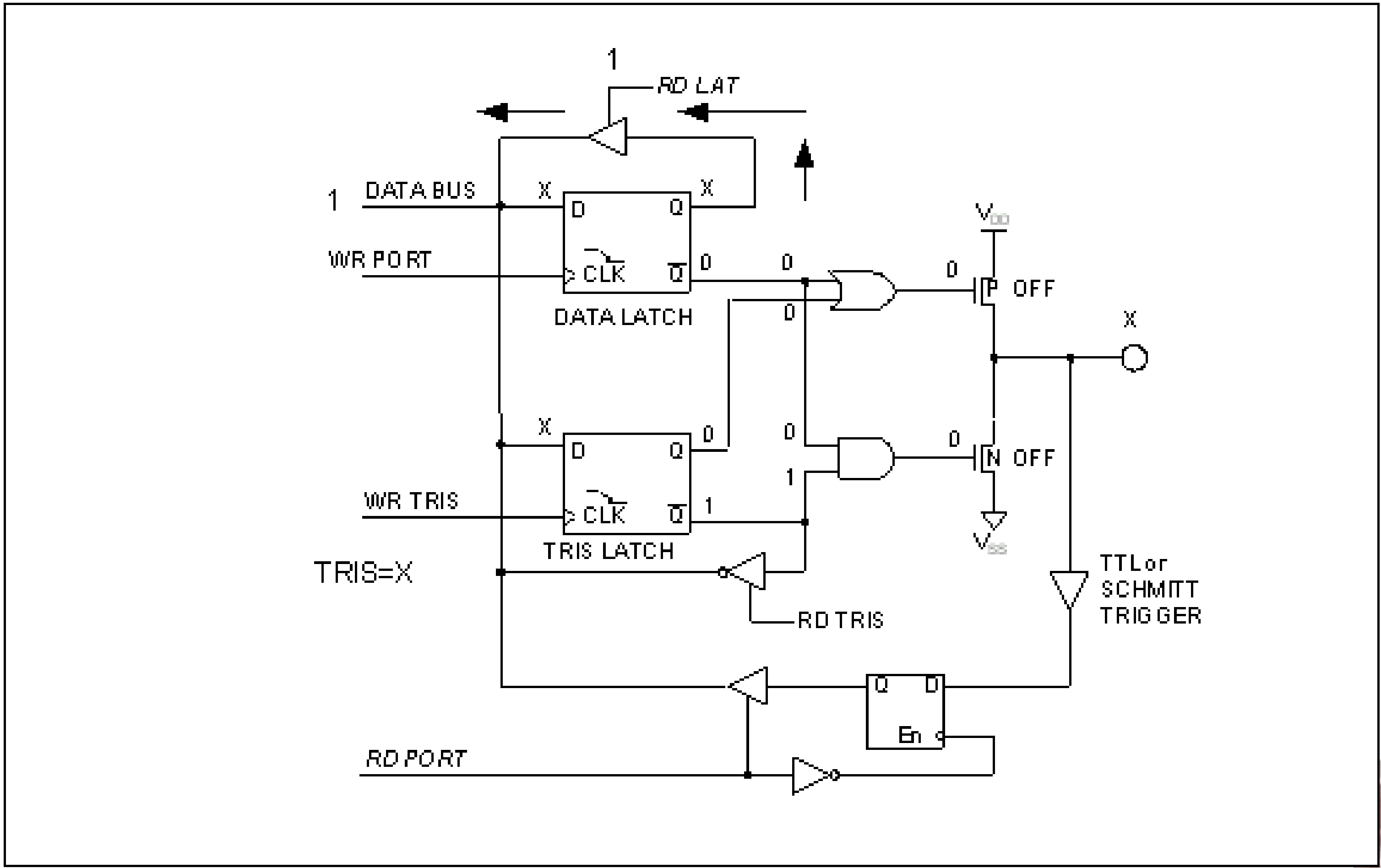


Table 4-10: Some of the Read-Modify-Write Instructions

Instruction		Function
ADDWF	fileReg,d	Add WREG to f
BSF	fileReg,bit	Bit Set fileReg (set the bit: bit = 1)
BCF	fileReg,bit	Bit Clear fileReg (clear the bit: bit = 0)
COMF	fileReg,d	Complement f
INCF	fileReg,d	Increment f
SUBWF	fileReg,d	Subtract WREG from f
XORWF	fileReg,d	Exclusive-OR WREG with f

Figure 4-8. LATx Register Role in Reading a Port or Latch


1
RD LAT
1
DATA BUS
X
D
Q
X
WR PORT
CLK
Q
0
0
0
P OFF
DATA LATCH
0
VDD
X
X
D
Q
0
0
0
N OFF
1
WR TRIS
CLK
Q
1
VSS
TRIS=X
TRIS LATCH
TTL or
SCHMITT
TRIGGER
RD TRIS
Q
D
En
RD PORT