EE361 Microcomputer-Based Digital Design

Quiz 1 Solution

Open book, open notes.

You may use the computer as long as you do not use the MPLAB program.

Name: _____
Section: _____
Date: 5 September 2008

There is a program listing on the back of this sheet.

1. List all the PIC16F884 instructions that provide 11 bits of data. What is the correct interpretation of these 11 bits?

   Solution

   The instructions are `call` and `goto`, as shown in Table 15-2 on page 232 of the data sheet. These are the only instructions in the list of 14-bit opcodes that contain 11 `k`-bits. These 11 bits constitute the 11 least significant bits of the 13-bit program counter PC. The higher-order bits come from the PCLATH register. In the case of the PIC16F884, the most significant of all the 13 bits (that is, bit 12) is a 0 because only $2^{11} = 2048$ program words have been implemented.

2. Use hexadecimal to express the smallest and the largest legitimate program addresses in the PIC16F884.

   Solution

   Figure 2-2 on page 21 of the datasheet shows that the smallest legal address is 0000_{16} and the largest legal address is 0fff_{16}.

Listing 1: Frament of a program for a PIC16F884 microcontroller.

```
1  X_data  equ     D'15'    ; Initial value for X
2  Y_data  equ     -D'25'   ; Initial value for Y
3  ZZZ     equ     ???      ; Storage location for Z
4  RP1     equ     ???     ; RP1 bit of STATUS register
5  RP0     equ     ???     ; RP0 bit of STATUS register
6  Status  equ     ???     ; Address of STATUS register
7      ...
8      bcf     Status,RP1   ; Designate Bank 1
9      bsf     Status,RP0
10     movlw   X_data       ; Retrieve X
11     addlw   Y_data       ; Add in Y
12     movwf   ZZZ          ; Store result in ZZZ
```

3. Consider the program fragment in Listing 1 on the preceding page. Its purpose is to sum two values, `X_data` and `Y_data`, and place the sum in a general purpose register `ZZZ`.

   a. The program has several incomplete instances of the assembler directive `EQU`. These are shown by the presence of three question marks (???) in place of numbers. Provide suitable values for the missing numbers, using hexadecimal to express them.

      SOLUTION

      Suitable definitions of the `EQU` directives would be

      ```
      ZZZ equ H'20'
      ```

      Any address within the 16-byte ranges H'20'–H'7F', H'A0'–H'FF', or H'120'–H'17F' would do. Note that references within the three ranges H'F0'–H'FF', H'170'–H'17F', and H'1F0'–H'1F' are synonyms for corresponding addresses within the range H'70'–H'7F'.

      ```
      RP1 equ 0x6
      ```

      Table 2-1 on page 26 shows that the RP1 bit in the STATUS register is in bit position 6. This time we coulduse an alternative means of specifying a hexadecimal constant

      ```
      RP0 equ H'5'
      ```

      Table 2-1 on page 26 shows that the RP0 bit in the STATUS register is in bit position 5.

      ```
      Status equ 0x3.
      ```

      Figure 2-5 on page 24 shows that the STATUS register is register number 3. However, values of H'83', H'103', and H'183' are perfectly permissible—albeit more obscure—synonyms for this value.

   b. Use hexadecimal to explain what the final result will be and what the values of the Z, C, and DC flags will be.

      SOLUTION

      The final result will be

      $$\begin{aligned} -10_{10} &= -00001010_2 \\ &= (11110101 + 1)_{\text{2's complement}} \\ &= 11110110_{\text{2's complement}} \\ &= \text{F6}_{\text{16's complement}}. \end{aligned}$$

      or, bearing in mind that $15_{10} = 0000\,1111$ and $-25_{10} = -0001\,1001 = 1110\,0111$

      $$\begin{array}{rr} & 0000\,1111 \\ + & 1110\,0111 \\ \hline & 1111\,0110 \end{array}$$

      which has the advantage of showing that DC= 1 and C= 0. Also, Z= 0.

   c. Use hexadecimal to show the address in data memory at which the result will be stored.

      SOLUTION

      The result will be stored at location $\langle \text{RP1} \rangle \langle \text{RP0} \rangle [\text{ZZZ}] = 0, 1, 010\,0000_2 = 0\text{A}0_{16}$. Note that ZZZ is only a seven-bit address; the other two bits needed come from the RP1 and RP0 bits in the STATUS register. Bit RP1 was cleared (made a 0) in line 8 and bit RP0 was set (made a 1) in line 9.

      That there are only seven bits is clearly seen by examining the entry for the `movwf` instruction shown in Table 15-2 on page 232: it only contains seven `f`-bits to hold file-address information.

      If ZZZ had been defined to have needed more than seven bits, only the *least* signficant bits would have been used. The two most significant bits of the nine-bit address still would have come from RP1 and RP0. This kind of error can be hard to notice.