

EECE.3170: Microprocessor Systems Design I

Summer 2016

Lecture 11: Key Questions

June 15, 2016

1. Describe how to write PIC code to implement operations that deal with two registers (e.g. moving the contents of one register to another; adding two registers).
2. Describe how to implement conditional jumps.

3. Describe how to implement simple rotate operations and multi-bit shift/rotate operations.

4. Translate these x86 operations to PIC code. Assume that there are registers defined for each x86 register (e.g. AL, AH, BL, BH, etc.)

- OR AL, BL

- SUB BL, AL

- JNZ label

- JB label (B = below = unsigned less than)

- ROL AL, 5

5. Describe how to work with multi-byte data.

6. Translate these x86 operations to PIC code. Assume that there are registers defined for each x86 register (e.g. AL, AH, BL, BH, etc.). 16-bit values (e.g., AX) must be dealt with as individual bytes

- `MOVZX AX, BL`

- `MOVSX AX, BL`

- `INC AX`

- `SUB BX, AX`

- `RCL AX, 5`

Describe the operation of the given subroutine, which implements a 10 ms delay loop.

```
.*****  
;  
; TenMs subroutine and its call inserts a delay of exactly ten milliseconds  
; into the execution of code.  
; It assumes a 4 MHz crystal clock. One instruction cycle = 4 * Tosc.  
; TenMsH equ 13 ; Initial value of TenMs Subroutine's counter  
; TenMsL equ 250  
; COUNTH and COUNTL are two variables  
TenMs  
    nop ; one cycle  
    movlw TenMsH ; Initialize COUNT  
    movwf COUNTH  
    movlw TenMsL  
    movwf COUNTL  
Ten_1  
    decfsz COUNTL,F ; Inner loop  
    goto Ten_1  
    decfsz COUNTH,F ; Outer loop  
    goto Ten_1  
    return
```

1. What factors determine amount of delay in loop?
2. What's downside of using loop for delay?
3. Under what conditions does function decrement COUNTH?
4. Under what conditions does function return?
5. How many times does each instruction in this function execute?

Describe the operation of the given subroutine, which toggles a series of 3 LEDs in sequence, assuming those LEDs are attached to bits 0-2 of Port D.

BlinkTable

```
movf    PORTD, W      ; Copy present state of LEDs into W
andlw   B'00000111'   ; and keep only LED bits
addwf   PCL,F         ; Change PC with PCLATH and offset in W
retlw   B'00000001'   ; (000 -> 001) reinitialize to green
retlw   B'00000011'   ; (001 -> 010) green to yellow
retlw   B'00000110'   ; (010 -> 100) yellow to red
retlw   B'00000010'   ; (011 -> 001) reinitialize to green
retlw   B'00000101'   ; (100 -> 001) red to green
retlw   B'00000100'   ; (101 -> 001) reinitialize to green
retlw   B'00000111'   ; (110 -> 001) reinitialize to green
retlw   B'00000110'   ; (111 -> 001) reinitialize to green
```

In calling program

```
call    BlinkTable ; get bits to change into W
xorwf   PORTD, F    ; toggle them into PORTD
```


6. What do the first two instructions in this function do?

7. What does the addwf instruction do?

8. Why do we need 8 retlw instructions?

9. How is each return value used?

10. Why are upper 5 bits of every return value equal to 0?