

# **16.317: Microprocessor Systems Design I**

Spring 2014

Lecture 27: Key Questions  
April 11, 2014

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

2. 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

3. 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
```

4. 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
```