(Problem 2) Given that Fosc = 1 MHz, calculate the delay generated by the following subroutine code, show your work: (4 /4 points)

```
SUB1        MOVLW   D'250'
            MOVWF   D1
            MOVLW   1
            MOVWF   D2
            MOVWF   D3

L0          DECFSZ  D1, F
            GOTO    L1
            DECFSZ  D2, F
L1          GOTO    L2
            DECFSZ  D3, F
L2          GOTO    L0
            RETURN
```

$$2 * \underbrace{(1+1+1+1+1)}_{\text{sub 1}} + \underbrace{(1+2+2+2)*249}_{D1}$$

$$\underset{\text{when D1 is zero}}{+2} + \underset{D_2}{2} + \underset{D_3}{2} + \underset{\text{return}}{2} \Rightarrow$$

$$\underset{\text{call}}{2} + \underset{\text{sub1}}{5} + \underset{D1}{1743} + 8 \Rightarrow 1758 \#$$

$$\text{Delay} = 1758 * 4/1\times10^6$$

$$\Rightarrow 7.032 * 10^{-3}$$

$$\Rightarrow 7.032 \text{ ms}$$

(Problem 3) Given the code below, answer the following questions: (1.5 /7 points)

```
1)    STATUS     EQU      3
2)    Z          EQU      2

3)    COUNTER    EQU      0x22
4)               ORG      0x05
5)               MOVLW    0X03
6)               MOVWF    COUNTER
7)    B          EQU      1
8)    A          EQU      0
9)    LOOP       ADDWF    COUNTER, A
10)              DECF     COUNTER, B
                 BTFSS    STATUS, Z
                 GOTO     LOOP
                 DECF     COUNTER, B
                 END
```

$$\begin{array}{r} 011 \\ 011 + \\ \hline 110 \end{array} \quad \begin{array}{r} 101 \\ 010 + \\ \hline 111 \\ 001 + \\ \hline 000 \end{array}$$

A) What is the size of this program in **Bytes**? (1 point)

$$\frac{14 * 14}{8} \Rightarrow 24.5 \text{ Bytes}$$

B) How many instruction cycles does it take to execute the program? (1 point)

# 7

C) What are the final values of location 0x22 and W-register? (2 points)

i. [0x22] = 0X00

ii. [W] = ~~1000~~ 1000

$$\begin{array}{r} 111 \\ 001 + \\ \hline 1000 \end{array}$$

9

First Exam
30/3/2014

Embedded Systems
Computer Engineering Department

Dr. RamziSaifan
60 minutes

Name: [redacted] Student ID: [redacted] Section: STH MW

(Problem 1) (3 /9)

a) List the configurations required to keep the PIC in reset mode on power up for longest period of time, and without using external circuits. Do not calculate the delay. (2 points)

* the Pcounter is set to zero
* the SFR, are set such that peripherals are in safe and disabled
* applying logic zero in master clear input (MCLR)

0 ✗

b) Replace the following code by one instruction which should has the same effect (1 point)

```
MOVLW        0xFF
XORWF        0x32, 0
```

$\boxed{\text{Xorlw B'11111111'}}$ ✗

FF

c) Replace the following code by one instruction which should has the same effect (2 point)

```
MOVF         STATUS, 0
MOVWF        0x23
MOVF         0x22, 0
ADDWF        0x22, 1
BTFSC        0x23, 0
INCF         0x22, 1
MOVF         0x22, 0
```

$\boxed{\text{MOVF 0x22,0}}$ ✗

d) Is the following statement true or false and why? (2 point)

False ← Harvard architecture computers are usually faster and simpler:

Harvard is Faster because it use piplineing
but it's not simpler (it has diffrent sizes of busses and more than 1 bus

2

e) Write a piece of code that checks data memory location 0x25 and increments it if it is odd. (2 points)

```
btfsc        ?
incF         0x25, 1
goto         done
end
```

1

D) How many bits each of the following reserves in the program memory? (3 points)

I. A in line 8: zero ✓
II. B in line 10: 1 bit ✓
III. Loop in line 12: zero ✗
IV. STATUS in line 11: zero ✗
V. 0x03 in line 5: 8 bits ✓
VI. What is the address of the instruction in line 13? OX12 ✗

1.5

(Problem 4) Write a complete and running program which multiplies each data memory location in the range 0x20 through 0x35 by 4. Also, complement data memory location 0xA2. (1 /5 points)

```
Status      equ   0x03
FSR         equ   0x04
INDF        equ   0x00
Reset       equ   0x20
X           equ   D'4'
Z           eq    D'16'
Counter     equ   0x21

            org   0x000
            goto  start
            org   0x005

~~start     movlw~~
start       movlw   z
            movwf   counter
            movlw   0x20
            movwf   FSR

~~loop~~    call    multiply

~~multiply~~
```

```
~~multiply~~
~~repeat~~
~~multiply~~   ~~addwf 0x20,0~~
[illegible]
multiply repeat   ~~[illegible]~~ 0x20,0
                  Decfsz  X
                  goto    repeat
                  incfsr
```

```
multiply
     reapet    incwf    FSR
               Decfsz   X
               goto     repeat
               return
```

d) In a PIC16F84A, given that there are 5 LEDs connected to portA such that: the LEDs connected to RA0-RA2 are in current sink mode, while the LEDs connected to RA3-RA4 are connected in current source mode. Write below only two instructions which are required to make the five LEDs to be on. Assume that PORT A is already configured as output. **Explain briefly the answer. (2 points)**

X

Problem 3) (3 **/7 points)**

a) In the boxes below, write the required code which is required to complement memory location: 0x195 using both direct addressing and indirect addressing modes. **(2 points)**

temp equ 0x195

Using direct addressing

```
BCF PCLATH,4
BCF PCLATH,3
org 0x195
COMPF temp,F
```

X

Using indirect addressing

```
BCF PCLATH,4
BCF PCLATH,3
MOVLW 0x195
MOVWF FSR
COMPF INDF,F
```

X

0x195 = 0001 1001 0101

b) In the box below, write the required code to make the code execution goes to the instruction in location 0x15AA (i.e., goto 0x15AA). **(1 point)**

```
bsf PCLATH,4
bcf PCLATH,3
~~goto 0x15AA~~
~~goto 0x AA~~
goto 0x15AA
```

1

0x15AA = 1 0101 1010 1010

c) Assume microcontroller A wants to send one byte to microcontroller B serially:

Write a piece of code that will make A sends the data in 9 bits such that the ninth bit is 0. **(2 points)**

```
BCF TRISC,6
BCF TRISC,7
MOVLW D'6'
MOVWF SPBRG
BSF TXSTA,TXEN
BSF TXSTA,TX9D
```

```
     MOVLW B'1001 0000'   ; RX9 = 0
     MOVWF RCSTA
     MOVWF TXReg
     TXSTA
wait BTFSS TXSTA,TRMT
     goto wait
```

1

Assume B detected OERR, write the code to solve the problem and starts receiving again. **(2 points)**

```
     bcf RCSTA,CREN
     bcf RCSTA,OERR
wait BTFSS PIR1,RCIF
     goto wait
     goto save
save MOVF RCREG,W
     MOVWF,INDF
     incf FSR,F
     goto wait
```

1

14.5

30/4/2014 Embedded Systems Second exam
Computer Engineering Department 70 minutes

Name: [redacted] Student ID: [redacted] Section: ح ث ن

Problem 1) Given that the following code continuously adds the content of memory location 0x0A until an external interrupt is observed on RB0. In this case the result is stored in location 0x10 and the W-register is cleared. (8.5 **/11 points)**

```
#INCLUDE    P16F84A.INC
ORG         0X0000
GOTO        START
ORG         0X0004
            GOTO        ISR
START       BSF         STATUS , RP0
            BSF         INTCON , INTE
            BSF         OPTION_REG , 6 →
            BSF         INTCON , GIE
            BCF         STATUS , RP0
ADD         ADDWF       0X0A , 0
            GOTO        ADD

ISR         MOVWF       0X10
            CLRW
            BCF         INTCON , INTF
            RETFIE
            END
```

a) Modify the code such that the interrupt happens when a **falling edge** arrives on RB0. Do the modification beside the same code. **(1 point)** 1 BCF OPTION_REG , 6

b) What are the values to be stored in the option register and in timer0 register which are required to make timer0 overflows after 25.6ms given that F_{osc} = 1 MHz. Initialize any non-required bit to zero. Show your work. **(3 points)**

$*\text{delay} = (256 - \overset{TMR_0}{\cancel{TMR_0}}) * \frac{4}{F_{osc}} * \text{Prescaler}$

$\Rightarrow 25.6 \times 10^{-3} = \overset{100}{\cancel{[illegible]}} * 4 \times 10^{-6} * Y$

$\Rightarrow \boxed{Y = 64} \Rightarrow \boxed{256 - 100 = 156 = TMR_0}$

$\Rightarrow$ Pre-scaller = 64 = 101

256 − TMR0 = 100 ⇒ TMR0 = 156

Timer 0 = D'156'

Option = ~~0~~XD5

↳ 5 ≡ 101 ≡ 64 scale

2.5

c) In box1 below, modify the code above such that **when either timer0 interrupt or external interrupt on RB0 observed**, the result is stored in location 0x10 and the working register is cleared. For this part, use the values to be stored in option register and in timer0 register according to what has been calculated in **b above. (3 points)** ① ①

كل فلاغ لحاله

d) In box2 below, modify **only** the ISR from the above code such that every 5 rising edges arrive on RB0, the result is stored in location 0x10 and the working register is cleared. Register **COUNTER** is initialized to zero if you need it. **(4 points)**

Ⓒ Box(1)

```
#include P16F84A.INC
 org 0x00
 goto Start
  org 0x04
  goto ISR
Start
   bsf   Status, Rp0
   bsf  INTCON,INTE
   bsf  INTCON, GIE
   bcf  INTCON, T0IF
   bsf  INTCON, T0IE
   Banksel  option_reg
   Movlw  0XD6
   Movwf  option_reg.
  Banksel  TMR0
  Movlw  D'156'
  Movwf  TMR0
  Bcf   Status, RP0
ADD
        ADDWF  0x0A,0
        goto   ADD
ISR

 Movwf  0x10
 clrwf
 Bcf  INTCON, INTF
 Bcf  INTCON, T0IF
 ~~RETF~~RETFIE
    End
```