0907333 Embedded Systems (Fall 2011)
Midterm Exam

الاسم: Key رقم التسجيل: رقم الشعبة:

==========

Instructions: Time **90** minutes. Closed books and notes. No calculators or mobile phones. No questions are allowed. Show your work clearly. All numbers are in hexadecimal unless otherwise specified.

==========

Q1. *(6 marks)*

a) If you are given two PIC microcontrollers; namely, 16F84A and 16F877A, then what can you tell about the general similarities and differences between these two chips?

They have the same core (CPU) 0.5 mark

They differ in memory size and peripherals 0.5 mark

b) Explain in few words the purpose of the address 0x0004 in the program memory of the PIC 16 series microcontrollers.

This address is the interrupt vector. It is the address where interrupt service routines should start.

c) What do you understand when we say that the PIC 16 series microcontrollers are 8-bit microcontrollers?

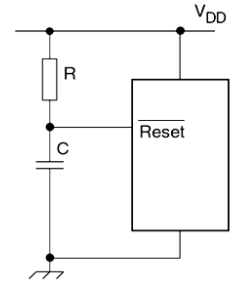
The ALU is 8 bits.

d) In PIC16 series, explain why there is no single instruction that can perform addition between two memory locations in one step.

Since the working register is hard-wired as one of the ALU inputs.

e) For the following circuit, what is the purpose of RC circuit that is connected to the RESET input of the microcontroller?

This configuration for the Reset input is required with slow power supplies (< 0.05 V/ms) to extend the reset period to assure proper operating conditions before program execution starts.



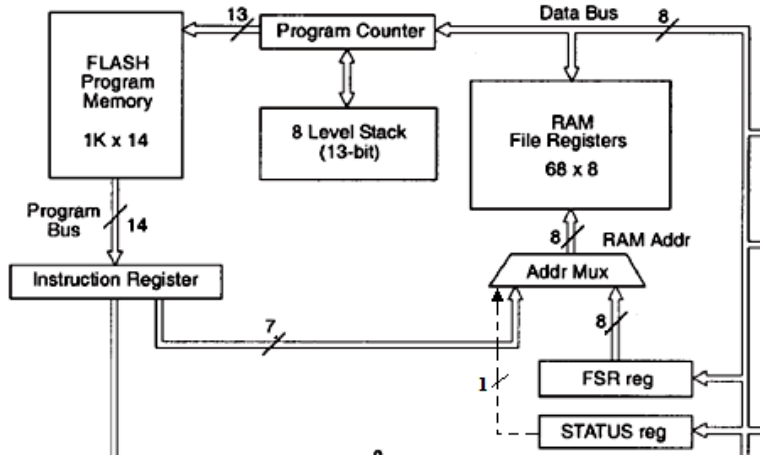
f) What is the effect of executing the following instructions?

```
movlw     b'11110000'
movwf     trisb
```

These instructions configure the lower 4 pins of PORTB as outputs and the remaining pins as inputs

Q2. The following figure shows part of the PIC1684A internal architecture. Study the figure and answer the following questions.

(4 marks)



a) What is the purpose of the Stack block?

The stack memory is used to store the program counter (address of next instruction) when interrupts occur or subroutines are called.

b) What is the size of each memory location in the Program Memory? Can you tell why?

14 bits, since the PIC 16 instructions are 14 bits.

c) What is the purpose of the 1-bit dashed wire?

It is the RP0 bit which is used for bank selection in direct addressing.

d) Explain why the Program Counter is connected to the Data Bus?

In some cases the ALU may execute instructions that modify the program counter, such ADDWF PCL, F

Q3. What happens inside the microcontroller hardware when the instruction **`call 34`** is executed?

(1 mark)

The program counter content is pushed to stack and then it is loaded with 34 to start the execution of the subroutine.

Q4. Assume that the following code has just been executed.

(2 marks)

```
movlw  2f
addlw  55
```

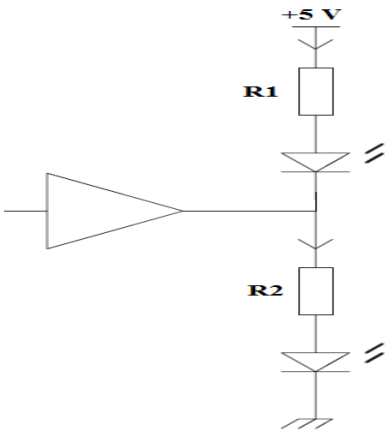
a) Specify the condition of the following three status flags: 0.5 mark each

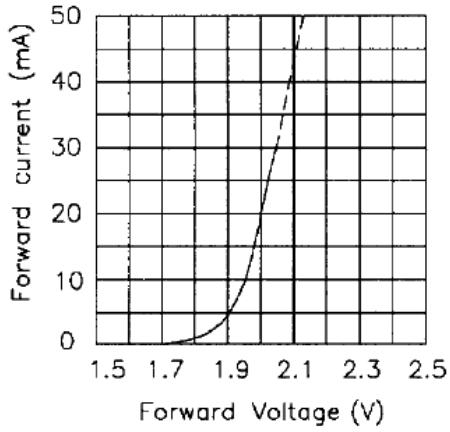
C:	0	DC:	1	Z:	0

b) The binary content of the working register is ____10000100_____

Q5. For the following circuit and with the aid of the LED voltage-current characteristics graph shown, determine the values for the current limiting resistors R1 and R2. Assume that the LED operates on a 40-mA current. Assume the output port characteristics as given in the table below.

(5 marks)





VOH	5 V
ROH	150 Ohm
VOL	0 V
ROL	40 Ohms

@ 40mA the LED voltage is 2.1 volts

R1 (2.5 marks)

KVL → 5 – 40mA*(R1+ROL) - 2.1 - VOL = 0

R1 = 32.5 Ohms

R2 (2.5 marks)

KVL → VOH – 40mA*(R2+ROH) - 2.1 = 0

R2 = -77.5 Ohms

The port can't drive the LED at the specified settings

Q6. How long does it take to execute the following instructions on a PIC 16F84A running at a clock of 4 MHz?

(1 mark)

```
        goto  L2
L1      movwf var1          TIME = #cycles * 4 / Fosc
        btfss var1,0             = (2+1) * 1 usec = 3 usec
L2      sublw 10
```

The answer is __3___ µsec

Q7. What is the content of the working register after executing the following sequence of instructions?

(1 mark)

```
movlw  08
movwf  20
subwf  20, w
```

The answer is _00____

Q8. Write the initialization code needed for the PIC 16F84A to set Timer TMR0 (at address 01h) to generate an interrupt after 20 external pulses. The Registers INTCON (at address 0bh) and OPTION_REG (at address 81h) are shown below. You need to write the code needed to initialize the three registers properly.

(5 marks)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	EEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
bit 7							bit 0

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7							bit 0

bit 7 **$\overline{\text{RBPU}}$:** PORTB Pull-up Enable bit
1 = PORTB pull-ups are disabled
0 = PORTB pull-ups are enabled by individual port latch values

bit 6 **INTEDG**: Interrupt Edge Select bit
1 = Interrupt on rising edge of RB0/INT pin
0 = Interrupt on falling edge of RB0/INT pin

bit 5 **T0CS**: TMR0 Clock Source Select bit
1 = Transition on RA4/T0CKI pin
0 = Internal instruction cycle clock (CLKOUT)

bit 4 **T0SE**: TMR0 Source Edge Select bit
1 = Increment on high-to-low transition on RA4/T0CKI pin
0 = Increment on low-to-high transition on RA4/T0CKI pin

bit 3 **PSA**: Prescaler Assignment bit
1 = Prescaler is assigned to the WDT
0 = Prescaler is assigned to the Timer0 module

bit 2-0 **PS2:PS0**: Prescaler Rate Select bits

Bit Value	TMR0 Rate	WDT Rate
000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64
111	1 : 256	1 : 128

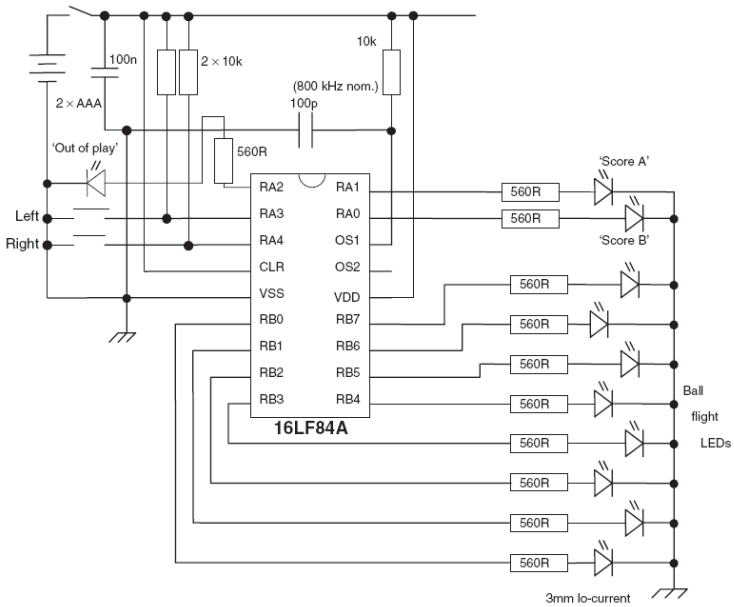
One possible soultion

TMR0 = 256 – 20 = 236 (to overflow after 20 pulses) 1 mark
OPTION_REG = B'xx1x 1xxx' (select external clock and assign prescaler to WDT) 1 mark
INTCON = B'1x1x00xx' (GIE =1, T0IE = 1, T0IF = 0) (1 mark)

```
movlw  D'236'
movwf  TMR0              0.5 mark
bsf    STATUS, RP0       0.5 mark
movlw  0x28
movwf  OPTION_REG        0.5 mark
bsf    INTCON, T0IE
bsf    INTCON, GIE       0.5 mark
```

Q9. For the following ping-pong circuit, complete the program below to achieve the function summarized in the table shown to the right.

(5 marks)



Right button	Left button	LEDs turned on
Released	Released	All off
Released	Pushed	Lower 4
Pushed	Released	LEDs connected to RB7:RB4
Pushed	Pushed	Lower 8

```
   status      equ 03
   porta       equ 05
   trisa       equ 05
   portb       equ 06
   trisb       equ 06
   temp        equ 20
   ;
               org 00
   ;
   start       bsf status,5
               movlw B'00011000'
               movwf trisa
               movlw 00
               movwf trisb
               bcf status,5
repeat         movlw B'00011000'
               andwf porta,w     ; mask RA4 and RA3
               movwf temp

               movlw B'00011000' ; case 1
               subwf temp, w
               btfsc status, z   ;
               goto  case1

               movlw B'00010000' ; case 2
               subwf temp, w
               btfsc status, z   ;
               goto  case2

               movlw B'00001000' ; case 3
               subwf temp, w
               btfsc status, z   ;
               goto  case3
```

- Other solutions possible
- Grading
  1) Checking each case is 0.75 mark
  2) Output correct values is 0.5 mark

```
                movlw 0xff       ; case 4
                movwf portb
                clrf  porta
                goto  repeat

case1           clrf portb
                clrf porta
                goto repeat

case2           movlw 0x0f
                movwf portb
                goto  repeat

case3           movlw 0xf0
                movwf portb
                clrf  porta
                goto  repeat
                end
```

TABLE 7-2: PIC16CXXX INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	14-Bit Opcode MSb			LSb	Status Affected	Notes
BYTE-ORIENTED FILE REGISTER OPERATIONS									
ADDWF	f, d	Add W and f	1	00	0111	dfff	ffff	C,DC,Z	1,2
ANDWF	f, d	AND W with f	1	00	0101	dfff	ffff	Z	1,2
CLRF	f	Clear f	1	00	0001	lfff	ffff	Z	2
CLRW	-	Clear W	1	00	0001	0xxx	xxxx	Z	
COMF	f, d	Complement f	1	00	1001	dfff	ffff	Z	1,2
DECF	f, d	Decrement f	1	00	0011	dfff	ffff	Z	1,2
DECFSZ	f, d	Decrement f, Skip if 0	1 (2)	00	1011	dfff	ffff		1,2,3
INCF	f, d	Increment f	1	00	1010	dfff	ffff	Z	1,2
INCFSZ	f, d	Increment f, Skip if 0	1 (2)	00	1111	dfff	ffff		1,2,3
IORWF	f, d	Inclusive OR W with f	1	00	0100	dfff	ffff	Z	1,2
MOVF	f, d	Move f	1	00	1000	dfff	ffff	Z	1,2
MOVWF	f	Move W to f	1	00	0000	lfff	ffff		
NOP	-	No Operation	1	00	0000	0xx0	0000		
RLF	f, d	Rotate Left f through Carry	1	00	1101	dfff	ffff	C	1,2
RRF	f, d	Rotate Right f through Carry	1	00	1100	dfff	ffff	C	1,2
SUBWF	f, d	Subtract W from f	1	00	0010	dfff	ffff	C,DC,Z	1,2
SWAPF	f, d	Swap nibbles in f	1	00	1110	dfff	ffff		1,2
XORWF	f, d	Exclusive OR W with f	1	00	0110	dfff	ffff	Z	1,2
BIT-ORIENTED FILE REGISTER OPERATIONS									
BCF	f, b	Bit Clear f	1	01	00bb	bfff	ffff		1,2
BSF	f, b	Bit Set f	1	01	01bb	bfff	ffff		1,2
BTFSC	f, b	Bit Test f, Skip if Clear	1 (2)	01	10bb	bfff	ffff		3
BTFSS	f, b	Bit Test f, Skip if Set	1 (2)	01	11bb	bfff	ffff		3
LITERAL AND CONTROL OPERATIONS									
ADDLW	k	Add literal and W	1	11	111x	kkkk	kkkk	C,DC,Z	
ANDLW	k	AND literal with W	1	11	1001	kkkk	kkkk	Z	
CALL	k	Call subroutine	2	10	0kkk	kkkk	kkkk		
CLRWDT	-	Clear Watchdog Timer	1	00	0000	0110	0100	$\overline{TO}$,$\overline{PD}$	
GOTO	k	Go to address	2	10	1kkk	kkkk	kkkk		
IORLW	k	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z	
MOVLW	k	Move literal to W	1	11	00xx	kkkk	kkkk		
RETFIE	-	Return from interrupt	2	00	0000	0000	1001		
RETLW	k	Return with literal in W	2	11	01xx	kkkk	kkkk		
RETURN	-	Return from Subroutine	2	00	0000	0000	1000		
SLEEP	-	Go into standby mode	1	00	0000	0110	0011	$\overline{TO}$,$\overline{PD}$	
SUBLW	k	Subtract W from literal	1	11	110x	kkkk	kkkk	C,DC,Z	
XORLW	k	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z	

<Good Luck>