

ECE 376 - Test #1: Name _____

1) **Digital Inputs.** A thermistor has the following resistance vs lumens:

$$R = 2000 \cdot \exp\left(\frac{4000}{T+273} - \frac{4000}{298}\right) \Omega$$

where T is the temperature in degrees C. Design a circuit which outputs:

- +5V when $T > 5C$
- +0V when $T < 0C$
- No change for $0C < T < 5C$

Let R1 be determined by your birth date

- $R1 = 900 + 100 \cdot \text{birth month} + \text{Birth Date}$

$R1 = 900 + 100(\text{Birth Month}) + \text{Birth Date}$ ex: May 14 = 1414 Ohms	$R1 = 1414 \text{ Ohms}$
--	--------------------------

@ 5C: (on)

$$R = 5253.1 \Omega$$

$$V_{on} = 3.9396V$$

@ 0C (off)

$$R = 6837.0 \Omega$$

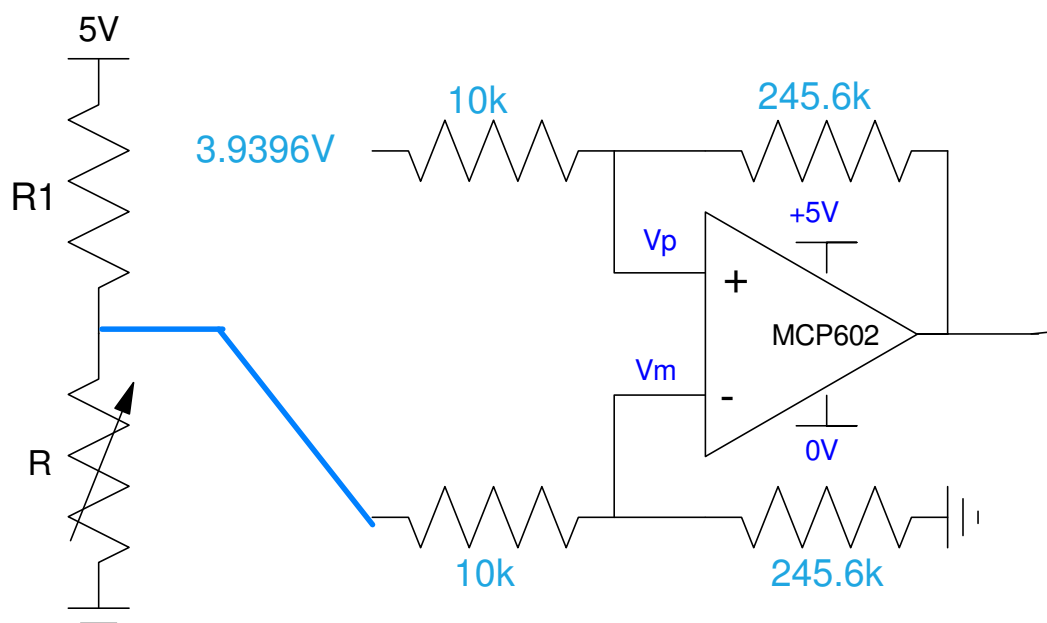
$$V_{off} = 4.1431V$$

Since $V_{on} < V_{off}$, connect to the minus input

Set the offset equal to V_{on}

Gain is

$$\text{gain} = \left(\frac{\text{change in output}}{\text{change in input}} \right) = \left(\frac{5V - 0V}{4.1431V - 3.9396V} \right) = 24.56$$



2) Digital Outputs: A 5W green LED has the following specifications

- 520 - 525nm wavelength
- Forward Voltage: 7V @ 700mA
- Luminous Flux: 600 Lumens @ 700mA

Determine Rc, and Rb so that a PIC microcontroller drives this LED at N mA where

- $N = 900 + 100 * (\text{birth month}) + (\text{birth date})$

Assume a 6144 NPN transistor

- $V_{be} = 700\text{mV}$
- $V_{ce}(\text{sat}) = 0.2\text{V}$
- Current gain = $\beta = 200$

N mA 900 + 100*(Birth Month) + Birth Date ex: May 14th = 1414mA	Lumens	Rb	Rc
1414mA	1212 lumens	430 Ohms	2.1975 Ohms

Rc

$$R_c = \left(\frac{10\text{V} - 7\text{V} - 0.2\text{V}}{1414\text{mA}} \right) = 2.1075\Omega:$$

Lumens

$$LM = \left(\frac{1414\text{mA}}{700\text{mA}} \right) 600 = 1212 \text{ lumens}$$

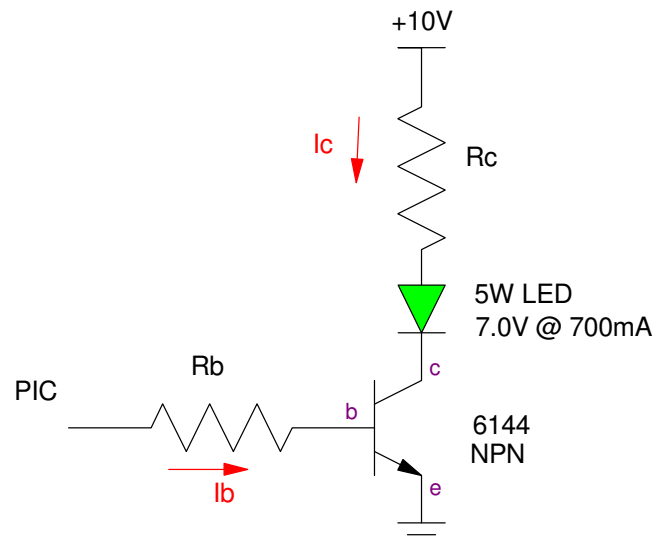
To saturate the transistor

$$\beta I_b > I_c$$

$$I_b > \left(\frac{I_c}{\beta} \right) = \left(\frac{1414\text{mA}}{200} \right) = 7.07\text{mA}$$

Let $I_b = 10\text{mA}$

$$R_b = \left(\frac{5\text{V} - 0.7\text{V}}{10\text{mA}} \right) = 430\Omega$$



3) **Assembler:** Determine the contents of the W, PORTB, and PORTC registers after each operation.
Assume

- PORTB and PORTC are output.
- Default is decimal

	W	PORTB	PORTC
Start:	0x13 = 19	5 birth month	14 Birth Date (1-31)
addwf PORTC, F	19	5	33
subwf PORTB, F	19	-14 or +242	33
andwf PORTC, W	18	-14	33
negf PORTB, F	18	14	33
movf PORTC, W	33	14	33
movff PORTB, PORTC	33	14	14
movwf PORTB	33	33	14
incf PORTB, W	34	33	14
decf PORTC, F	34	33	13

4) Assembler & Timing: Determine the number of clocks the following assembler subroutine takes to execute. Assume MONTH and DAY be your birth month and day.

MONTH (birth month: 1..12)	DAY (birth day: 1..31)	N Number of clocks Wait routine takes
5	14	108,196
Find A / B / C for N = 31,414,000 +/- 500,000 (3.414 seconds +/- 0.05 second)		
A = 255	B = 252	C = 60 N = 31,425,186 off by 11,186

$$N = 8ABC + 9AB + 8A + 6$$

Wait:

```

movlw    MONTH (A)           6
movwf    CNT2
nop
nop

```

W2:

```

    movlw    DAY (B)          8 * A
    movwf    CNT1
    nop
    nop
    nop

```

W1:

```

    movlw    192 (C)          9 * A * B
    movwf    CNT0
    nop
    nop
    nop
    nop

```

W0:

```

    nop                      N = 8 * A * B * B
    nop
    nop
    nop
    nop
    decfsz   CNT0,F
    goto     W0

```

```

    decfsz   CNT1,F
    goto     W1

```

```

    decfsz   CNT2,F
    goto     W2

```

return

5) Assembler & Flow Charts. Write an assembler program to turn your PIC processor into flashlight with auto shut off

- Press RB0 to turn off the lights (PORTC = PORTD = 0)
- Press RB1 to turn on the lights (PORTC = PORTD = 0xFF)
- Every 3.14 seconds, the number in PORTC and PORTD decrements to zero
- When you reach zero, the light stays off

Write the corresponding assembler code

note: Assume a Wait routine exists that waits 3.14 seconds

- (a different problem on this test)

```
#include <pic18f4620.inc>
```

```

    org    0x800
    movlw  0x0F
    movwf  ADCON1

L1:   movlw  0xFF
    movwf  TRISB
    clrf   TRISC
    clrf   TRISD

L2:   clrf   PORTC
    clrf   PORTD

L3:   movlw  0
    cpfseq PORTB
    goto   L4
    goto   L3

L4:   btfss  PORTB, 0
    goto   L6

L5:   clrf   PORTC
    clrf   PORTD

L6:   btfss  PORTB, 1
    goto   L8

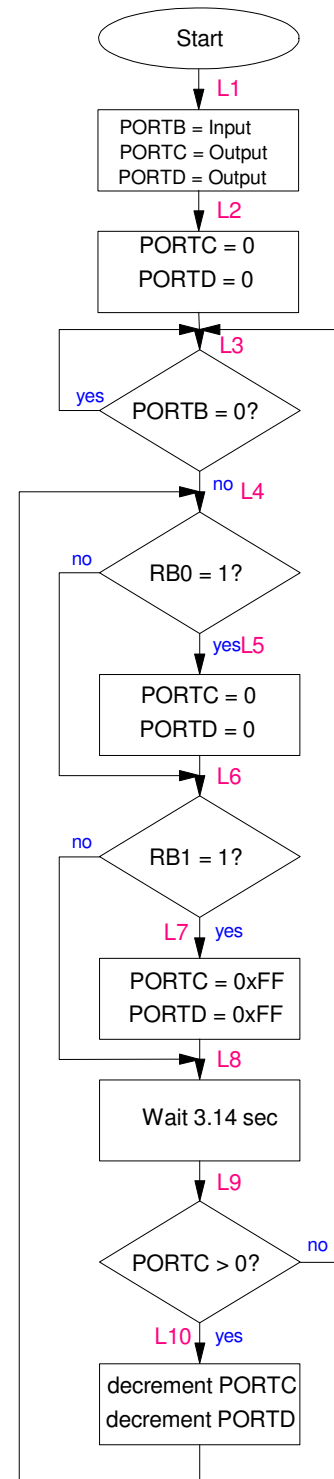
L7:   movlw  0xFF
    movwf  PORTC
    movwf  PORTD

L8:   call   Wait

L9:   movlw  0
    cpfsgt PORTC
    goto   L3

L10:  decf   PORTC, F
    decf   PORTD, F
    goto   L4

```



Memory Read & Write			
MOVWF	PORTA	memory write	w → PORTA
MOVFF	PORTA PORTB	copy	PORTA → PORTB
MOVF	PORTA,W	memory read	PORTA → W
MOVLW	234	Move Literal to WREG	123 → W
Memory Clear, Negation			
CLRF	PORTA	clear memory	0x00 → PORTA
COMF	PORTA, W	toggle bits	!PORTA → W (bit toggle)
NEGF	PORTA, W	negate	-PORTA → W (2's compliment)
Addition & Subtraction			
INCF	PORTA,F	increment	PORTA + 1 → PORTA
ADDWF	PORTA, F	add	PORTA + W → PORTA
ADDWFC	PORTA, W	add with carry	PORTA + W + carry → W
ADDLW		Add Literal and WREG	
DECF	PORTA,F	decrement	PORTA - 1 → PORTA
SUBFWB	PORTA,F	subtract with borrow	PORTA - W - c → PORTA
SUBWF	PORTA,F	subtract no borrow	PORTA - W → PORTA
SUBWFB	PORTA,F	subtract with borrow	PORTA - W - c → PORTA
SUBLW	223	Subtract WREG from #	223 - W → W
Shift left (*2), shift right (/2)			
RLCF	PORTA,F	rotate left through carry (9-bit rotate)	
RLNCF	PORTA,F	rotate left no carry	
RRCF	PORTA,F	rotate right through carry	
RRNCF	PORTA,F	rotate right no carry	
Bit Operations			
BCF	PORTA, 3	Bit Clear f	clear bit 3 of PORTA
BSF	PORTA, 4	Bit Set f	set bit 4 of PORTA
BTG	PORTA, 2	Bit Toggle f	toggle bit 2 of PORTA
Logical Operations			
ANDWF	PORTA, F	logical and	PORTA = PORTA and W
ANDLW	0x23	AND Literal with WREG	W = W and 0x23
IORWF	PORTA,F	logical or	PORTA = PORTA or W
IORLW	0x23	Inclusive OR Literal	W = W or 0x23
XORWF	PORTA,F	logical exclusive or	PORTA = PORTA xor W
XORLW	0x23	Exclusive OR Literal	W = W xor 0x23
Tests (skip the next instruction if...)			
CPFSEQ	PORTA	Compare PORTA to W, skip if PORTA = W	
CPFSGT	PORTA	Compare PORTA to W, Skip if PORTA > W	
CPFSLT	PORTA	Compare PORTA to W, Skip if PORTA < W	
DECFSZ	PORTA,F	decrement, skip if zero	
DCFSNZ	PORTA,F	decrement, skip if not zero	
INCFSZ	PORTA,F	increment, skip if zero	
INFSNZ	PORTA,F	increment, skip if not zero	
BTFSZ	PORTA, 5	Bit Test f, Skip if Clear	
BTSS	PORTA, 1	Bit Test f, Skip if Set	
Flow Control			
GOTO	Label	Go to Address 1st word	
CALL	Label	Call Subroutine 1st word	
RETURN		Return from Subroutine	
RETLW	0x23	Return with 0x23 in WREG	