

# ECE 376 - Test #1: Name \_\_\_\_\_

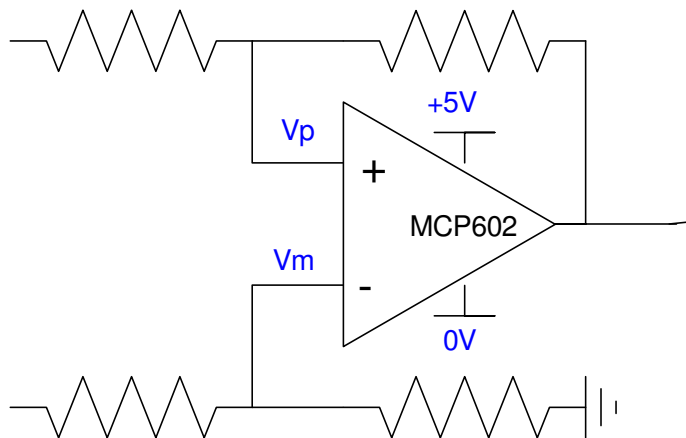
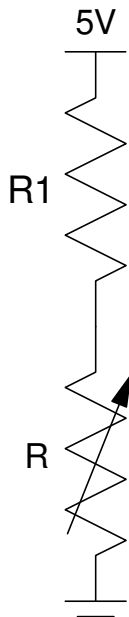
Open book, open notes. Calculators and Matlab permitted. Individual effort (help from other people or web sites where other people help you solve the problems not permitted). February 4, 2022

1) **Digital Inputs.** Design a circuit which outputs

- 0V when  $R < 1200$  Ohms
- 5V when  $R > 1400$  Ohms

Assume

- $R1 = 900 + 100 \cdot (\text{your birth month}) + (\text{your birth date})$ .
- May 14th, for example, gives  $R1 = 1414$  Ohms



2) Digital Outputs: Design a circuit which allows your PIC to drive a 10W LED at N mcd

- $N = 900 + 100 * (\text{your birth month}) + (\text{your birth date})$
- $N = 1414$  mcd for May 14th, for example

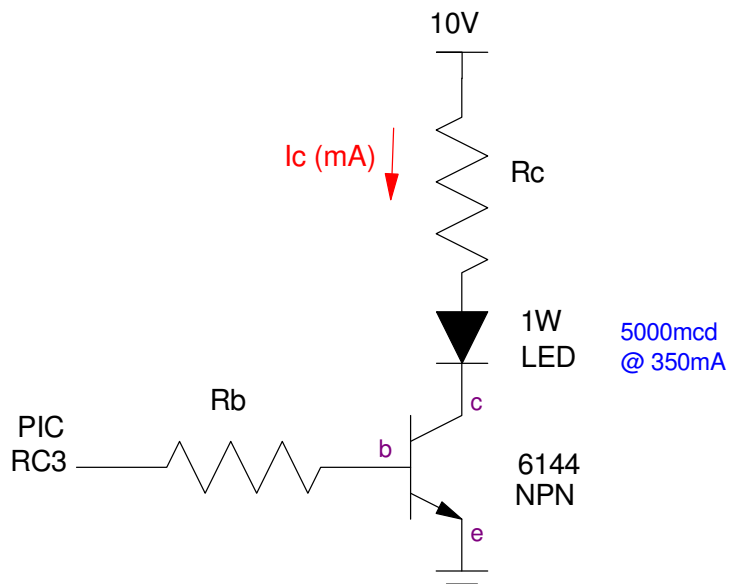
Assume a 1W LED has the following characteristics

- $V_f = 3.2V$  @  $350mA$
- $5,000mcd$  @  $350mA$

Assume a 6144 NPN transistor

- $V_{be} = 700mV$
- $V_{ce(sat)} = 360mV$
- Current gain =  $\beta = 200$

N mcd $900 + 100 * \text{Month} + \text{Day}$	Ic (mA) current needed to produce N mcd	Rb	Rc



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:	0	Birth Month (1..12)	Birth Date (1..31)
incf PORTB,W			
decf PORTC,F			
movlw 7			
addwf PORTB,F			
subwf PORTC,W			
movlw 7			
andwf PORTB,F			
iorwf PORTC,W			
negf PORTB,F			
comf PORTC,F			

#### 4) Assembler & Timing:

a) Determine the number of clocks the following assembler subroutine takes to execute.

- Assume MONTH and DAY be your birth month and day.

b) Modify this routine (change A, B, and C) so that it takes 35,000,000 clocks (3.5 seconds) to execute

- +/- 50,000 clocks

Month birth month 1..12	Day birth date 1..31	C	N number of clocks Wait takes
		200	
A	B	C	N
			35,000,000 +/- 50,000

Wait:

```
    movlw    MONTH    (A)
    movwf    CNT2
```

W2:

```
    movlw    DAY      (B)
    movwf    CNT1
    nop
    nop
    nop
    nop
```

W1:

```
    movlw    200 (C)
    movwf    CNT0
    nop
    nop
    nop
```

W0:

```
    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 a data encryption device with the encryption method depending upon which button is pressed:

- RB0 = clock      change PORTD on a rising edge
- RB1 = 1          PORTD = PORTC + 17
- RB2 = 1          PORTD = PORTC - 17
- otherwise        PORTD = PORTC

