ECE 376 - Final Exam: Name _____

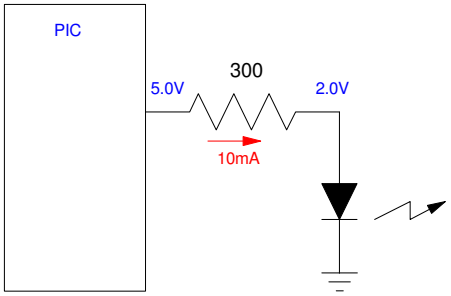
Open-Book, Open Note, Calculators and Matlab permitted. Individual Effort.

1) Binary Inputs: Assume a 6411 NPN transistor (if needed)

- Vbe = 0.7V
- Vce(sat) = 0.2V
- β = 300
- max(Ic) = 6A

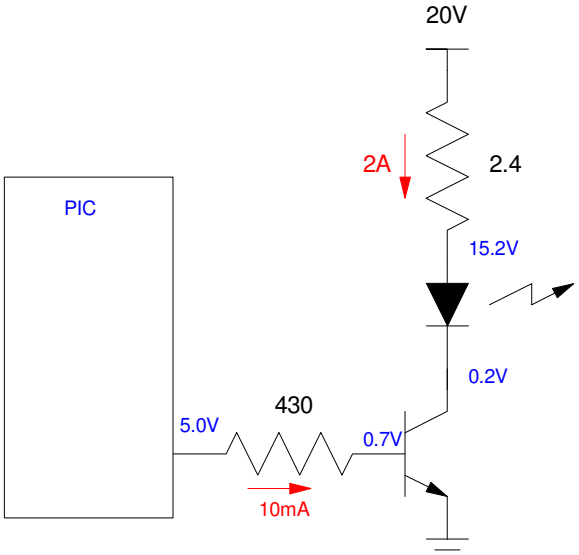
1a) Give a circuit which allows a PIC to turn on and off a 20mW LED at 10mA

- Id = 10mA
- Vd = 2.0V



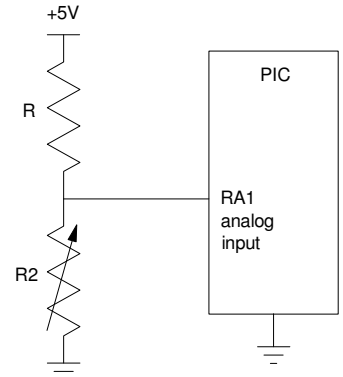
1b) Give a circuit which allows a PIC to turn on and off a 30W LED

- Vd = 15.0V
- Id = 2.0A



2) Analog Inputs: A CdS light sensor has the following resistance - lux (light intensity) relationship

$$R_2 = 1000 \cdot \left(\frac{10}{Lux}\right)^{0.6} \text{ Ohms}$$



If the room is 100 Lux, determine

- The resistance,
- The voltage,
- The A/D reading, and
- The resolution (the smallest change in Lux you can detect)

with the following circuit. Assume

- R = 900 + 100*(your birth month) + (your birth date)

R 900 + 100*mo + day	A/D reading 0..1023	Voltage votlage at 100 Lux	R resistance at 100 Lux	Resolution smallest change in Lux you can measure
1414	**154**	**0.7542**	**251.19**	**1.261**

$$R_2 = 1000 \cdot \left(\frac{10}{Lux}\right)^{0.6} = 251.19\Omega$$

$$V = \left(\frac{R_2}{R_2+R}\right)5V = 0.7542V$$

$$A2D = \left(\frac{V}{5}\right)1023 = 154.31$$

Resolution: If the A/D changes by 1.000 (smallest change you can detect)

V changes by 4.88mV

$$V = 0.7542 + \left(\frac{5V}{1023}\right) = 0.7591V$$

R becomes

$$R_2 = \left(\frac{0.7591V}{5-0.7591}\right)1414 = 253.091\Omega$$

Lux becomes

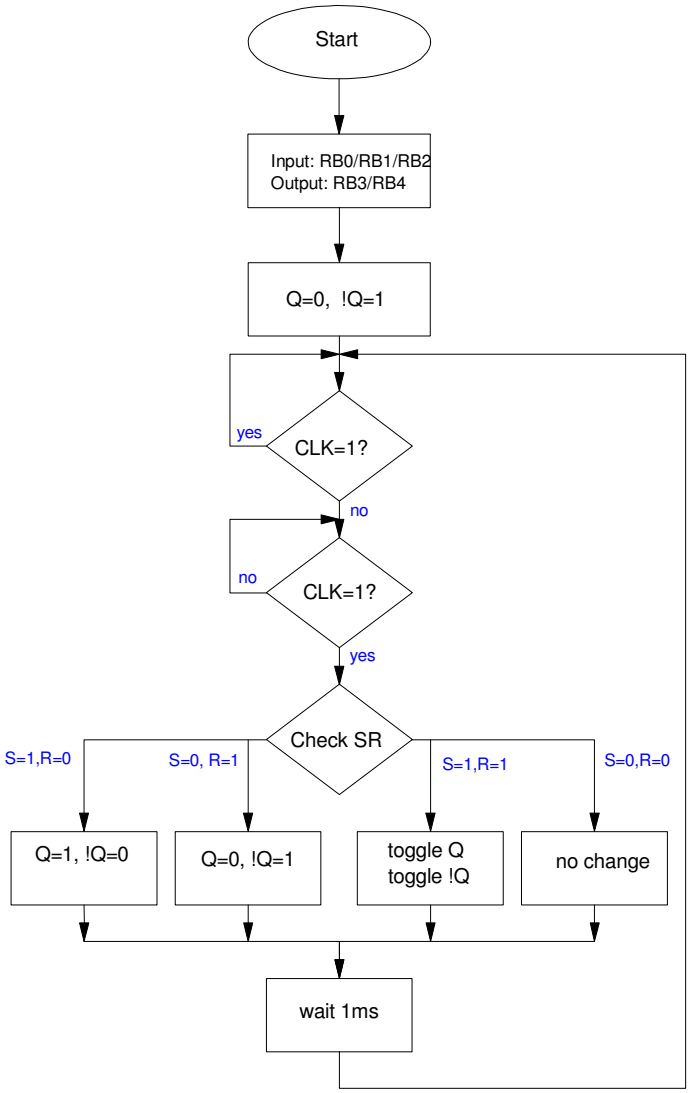
$$Lux = 93.786 \text{ lux}$$

or a change of 1.261 Lux

3) C-Coding: Write a C progra to turn your PIC into a $65 SR flip flop. Assume the following pin assignments

PORTB							
RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
-	-	-	Q output	!Q output	S input	R input	CLK input

```
void main(void) {

   ADCON1 = 0x0F;

   TRISB = 0x07;


   PORTB = 0x10;


   while(1) {

      while(RB0);

      while(!RB0);

      if( RB2 && !RB1 ) {
          RB4 = 1;
          RB3 = 0;
          }
      if( !RB2 && RB1) {
           RB4 = 0;
           RB3 = 1;
           }
      if( RB2 && RB1 ) {
           RB4 = !RB4;
           RB3 = !RB3;
            }

      Wait_ms(1);

   }
```



4) C Coding with Analog Inputs: Assume a temperature sensor is connected to a PIC so that the A/D reading is 10x the temperature in degrees F.

Write a C program which turns on and off a fan connected to RC0 based upon the temperature

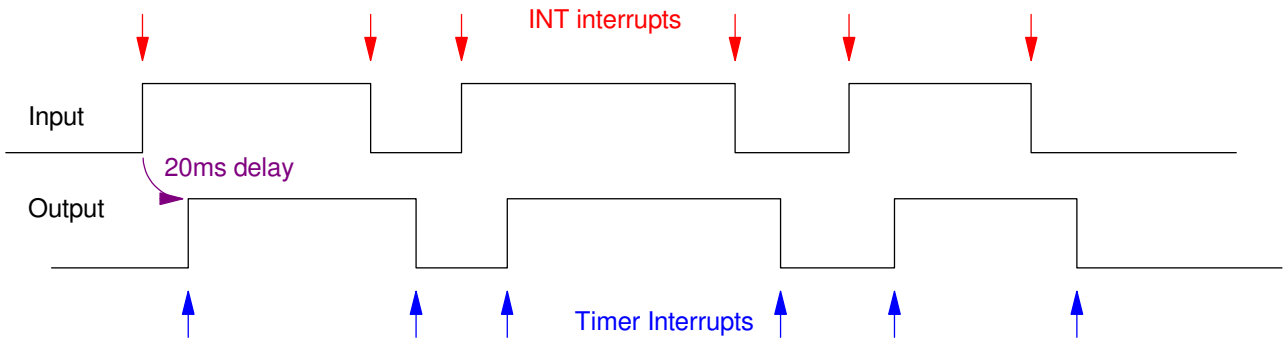
Temperature	A/D reading	% On	On-Time (RC0)	Off-Time (RC0)
T > 85F	850 - 1023	100%	always on	
80F < T < 85F	800 - 849	75%	45 seconds	15 seconds
75 < T <80	750 - 799	50%	30 seconds	30 seconds
70 < T < 75	700 - 749	25%	15 seconds	45 seconds
T < 70	0 - 699	0%	-	always off

```
void main(void)
{
   unsigned char A2D;
   unsigned char SECONDS;

// Initialize the A/D port
   TRISA = 0xFF;
   TRISE = 0x0F;
   ADCON2 = 0x85;
   ADCON1 = 0x07;
   ADCON0 = 0x01;

   while(1) {
      A2D = A2D_Read(0);
      SECONDS = (SECONDS + 1)% 60;

      if(A2D >= 850) PORTC = 0xFF;
      elseif(A2D >= 800) {
         if(SECONDS < 45) PORTC = 0xFF;
         else PORTC = 0;
         }
      elseif(A2D >= 750) {
         if(SECONDS < 30) PORTC = 0xFF;
         else PORTC = 0;
         }
      elseif(A2D >= 700) {
         if(SECONDS < 15) PORTC = 0xFF;
         else PORTC = 0;
         }
      else PORTC = 0;

      Wait_ms(1000);

      }
```

5) 20ms Delay (take 1): Using one or more Timer and/or INT interrupts, write the interrupt service routine for a C program which

- Reads in X, a TTL signal (0V/5V), and
- Outputs Y, the same waveform with a 20ms delay.

Assume each edge is more than 20ms apart



Input (pins used, INT nterrups used)	Output Timer interrupts. Assume pre-scalar = 8
RB0: INT0 (rising) RB1: INT1 (falling)	Timer0: PS=8, Set Timer1: PS=8, Clear

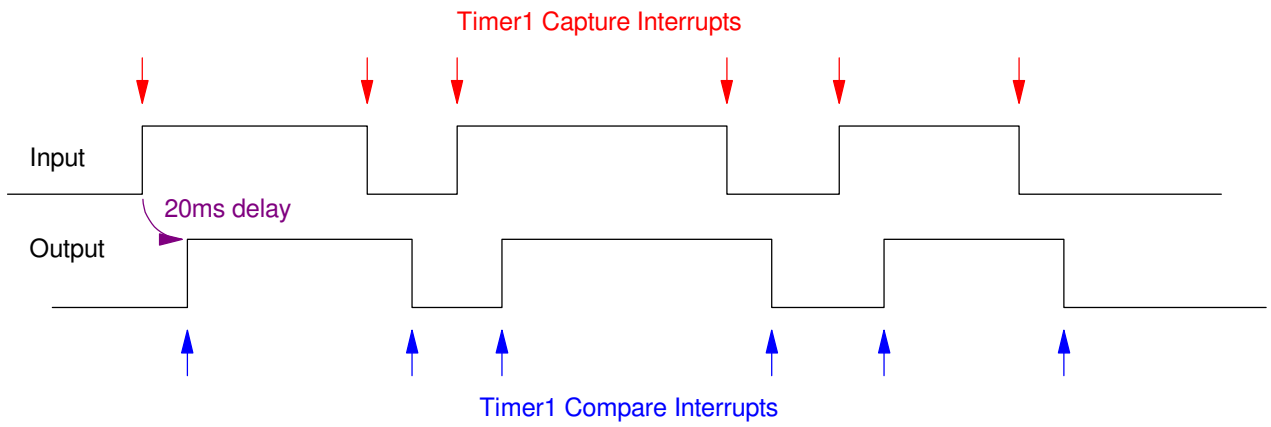
INT interrupt service routine(s)	Timer interrut service routines

`if(INT0IF) { // rising edge`
`  TMR0 = -25000;`
`  TMR0ON = 1;`
`  TMR0IF = 0;`
`  }`
 	
`if(INT1IF) { // falling edge`
`  TMR1 = -25000;`
`  TMR1ON = 1;`
`  TMR1IF = 0;`
`  }`
	`if(TMR0IF) { // rising edge`
	`  RC0 = 1;`
	`  TMR0ON = 0;`
	`  TMR0IF = 0;`
	`  }`
	
	`if(TMR1IF) { // falling edge`
	`  RC0 = 0;`
	`  TMR1ON = 0;`
	`  TMR1IF = 0;`
	`  }`

6) 20ms Delay (take 2 - Capture / Compare Interrupts): Write a program which uses Capture & Compare interrupts to

- Output a waveform (Y) which is identical to X, only
- Y is delayed by 20ms

Assume each edge is more than 20ms apart



Timer1 Interrupt assume prescalar = 8	Capture1 (Input)	Compare2 (Output)

`if(TMR1IF) {`
`  TIME += 0x10000;`
`  TMR1IF = 0;`
`  }`
	`if (CCP1IF) {`
	`  CCPR2 = CCPR1 + 25000;`
	`  if(CCP1CON == 0x05) {`
	`  CCP1CON = 0x04;`
	`  ccP2CON = 0x08;`
	`  }`
	`  else`
	`  CCP1CON = 0x05;`
	`  CCCP2CON = 0x09;`
	`  }`
	`  CCP1IF = 0;`
	`  }`
		`if(CCP2IF) {`
		`  CCP2IF = 0;`
		`  }`

20ms is 200,000 clocks

- 25000 intrrupts with a pre-scalar if 8

Capture 1 sets up Compare2

- Capture2 happens 20ms later (CCPR1 + 25000)
- If Capture1 captured the time of a rising edge (CCP1CON = 0x05)
  - Compare2 is set to set RC2 (CCP2CON = 0x08)
- If Capture1 captured the time of a falling edge (CCP1CON = 0x04)
- Compare2 is clear to set RC2 (CCP2CON = 0x09)