



c. ( 20 pts) Implement the flowchart and LED/switch configuration of Figure(c) using an interrupt to handle the button on the RB1 input. Use semaphores controlled by the ISR to tell the *while(1)* loop in the main() program to blink, turn on or turn off the LED. Declare all semaphores you use.

1. (14 pts) ISR code

2. (6 pts) *main()* code, be sure to configure and enable your interrupt. Do not assume any default values for variables or control bits.

- d. (6 pts) How many bytes can be sent in 3 seconds assuming a baud rate of 4800, and an asynchronous data format of 1 start bit, 8 data bits, and 2 stop bits assuming the bytes are sent as fast as possible?
- e. (6 pts) Write *C* code that implements the *void putchar(char c)* function (write one character to the serial port). No interrupts are enabled.
- f. (12 pts) Assume the definitions of a circular buffer that we have used in lab (i.e, the head pointer is used to place data into the buffer, the tail pointer is used to take data out of the buffer, the buffer is empty when head is equal to tail, and that pointers are incremented and wrapped before used to access the buffer).

f1.(3) From figure F, how many characters are currently *available* in the buffer? (this is not the total number of locations in the buffer) \_\_\_\_\_

f2. (3) From figure F, what character is returned if the buffer is read?  
\_\_\_\_\_

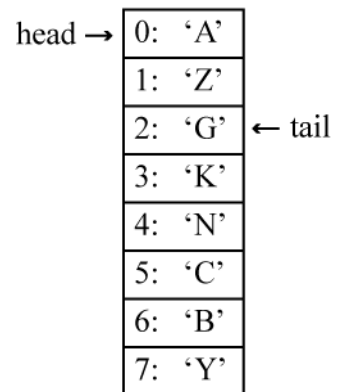
f3. (6) If a softwareFIFO as shown in figure F is used for interrupt driven asynchronous receive on the PIC, circle one (buffer read/buffer write/neither) for each of the following:

f3a. The Buffer Read/Buffer Write/neither is placed in the *putch()* function.

f3b. The Buffer Read/ Buffer Write/neither is placed in the *getch()* function.

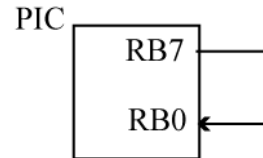
f3c. The Buffer Read/ Buffer Write/neither is placed in the interrupt service routine.

Problem (f)



- g. (9 pts) For each of the code segments below, draw the waveform that results on output RB7. If the waveform is repeating, then say so and draw at least two cycles. Also, clearly indicate logic '0' and logic '1' on your waveforms. For all code segments, pin RB7 is connected to RB0 as shown. Be sure to clearly indicate the initial state of the RB7 output.

Problem (g)



```

g.1
interrupt isr() {
if (INT0IF) {
    INT0IF = 0; RB7 = 1;
}
}
main() {
    TRISB = 0x7F; RB7 = 1;
    IPEN = 0; INTEDG0 = 0; INT0IF = 0;
    INT0IE = 1; PEIE = 1; GIE = 1;
    while (1){
        RB7 = 0;
        while(!RB0);
    }
}

```

```

g.2
interrupt isr() {
if (INT0IF) {
    INT0IF = 0; RB7 = 1; INT0IE = 0;
}
}
main() {
    TRISB = 0x7F; RB7 = 1;
    IPEN = 0; INTEDG0 = 0; INT0IF = 0;
    INT0IE = 1; PEIE = 1; GIE = 1;
    while (1){
        RB7 = 0;
        while(!RB0);
    }
}

```

```

g.3

interrupt isr() {
if (INT0IF) {
    INT0IF = 0; RB7 = 1;
}
}
main() {
    TRISB = 0x7F; RB7 = 1;
    IPEN = 0; INTEDG0 = 1; INT0IF = 0;
    INT0IE = 1; PEIE = 1; GIE = 1;
    while (1){
        RB7 = 0;
        while(!RB0);
    }
}

```

- h. (8 pts) Explain what happens in the code below by specifying what appears on the console after the PIC is powered on, and justify your answer by explaining the sequence of events. You must clearly specify if output continually appears on the console or if at some point it stops.

```
main() {
  char c;
  serial_init(95,1); // 19200 in HSPLL mode, crystal = 7.3728 MHz
  SWDTEN = 0;
  if (!POR) {
    POR = 1;
    SWDTEN = 1;
    printf("Hello!");pcrlf();
  }
  if(!SWDTEN) {
    asm("sleep");
  }
  while (1) {
    printf("Looping");pcrlf();
  } //end main()
}
```

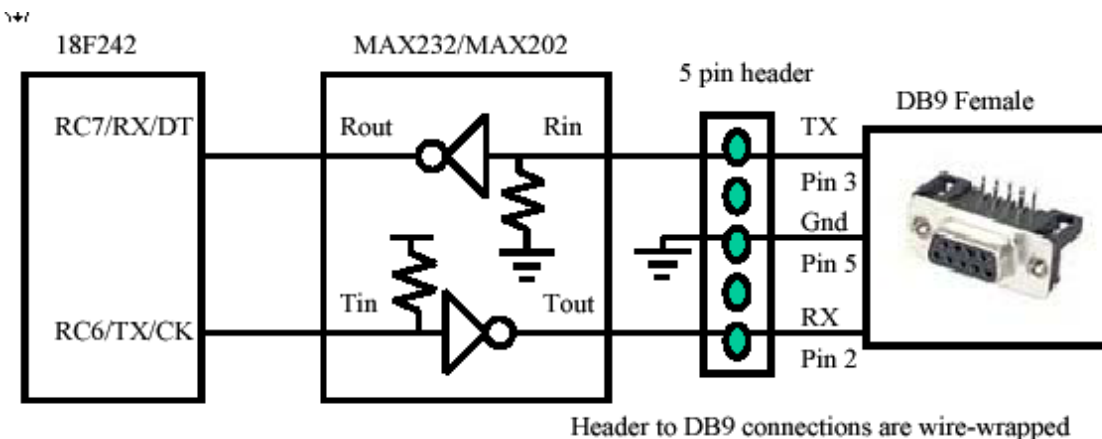
Part II: (20 pts) Answer 5 out of the next 7 questions. Cross out the 2 questions that you do not want graded. Each question is worth 4 pts.

1. In the table below, put a checkmark in each entry if an output port of the specified type can be driven to the particular state listed by the port itself (and not by any extra or external components)

	High?	Low?	High Impedance (aka, floating)
Normal CMOS output			
Tri-State output			
Open-drain output			

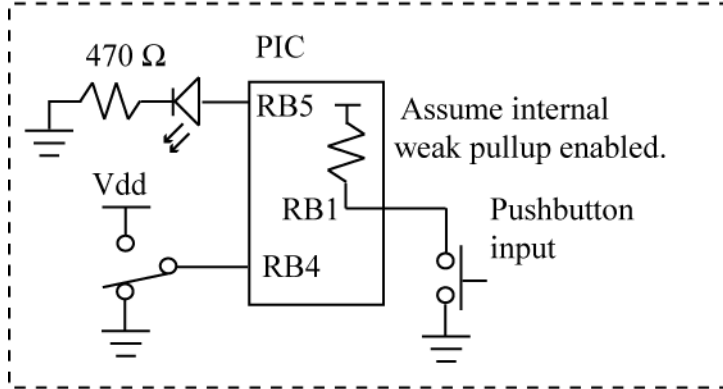
2. Draw a pushbutton connected to a PIC18 port that will provide a logic '1' when the switch is pressed, and a logic '0' when the switch is released. Your connection must be electrically viable, i.e. you cannot produce a short between VDD and GND or any other undesirable condition.

- If the PIC is drawing 3 mA using the HS oscillator option, and this is changed to the HSPLL option, what would you expect the new current draw to be? (give a number) Why? Justify your answer.
- Draw the waveform for the value 0x4A using 1 start bit, 1 stop bit, and 8-data bits as transmitted by the PIC from its TX pin using asynchronous serial IO.
- In the diagram below, assume the DB9 is connected to the RS232 port of a PC. If you measured the voltage at the TX pin of the DB9 when the port is idle (no characters being transmitted), about what voltage would you expect to see? If you measured the voltage at the PIC18 TX pin when the port is idle (no characters being transmitted), about what voltage would you expect to see?

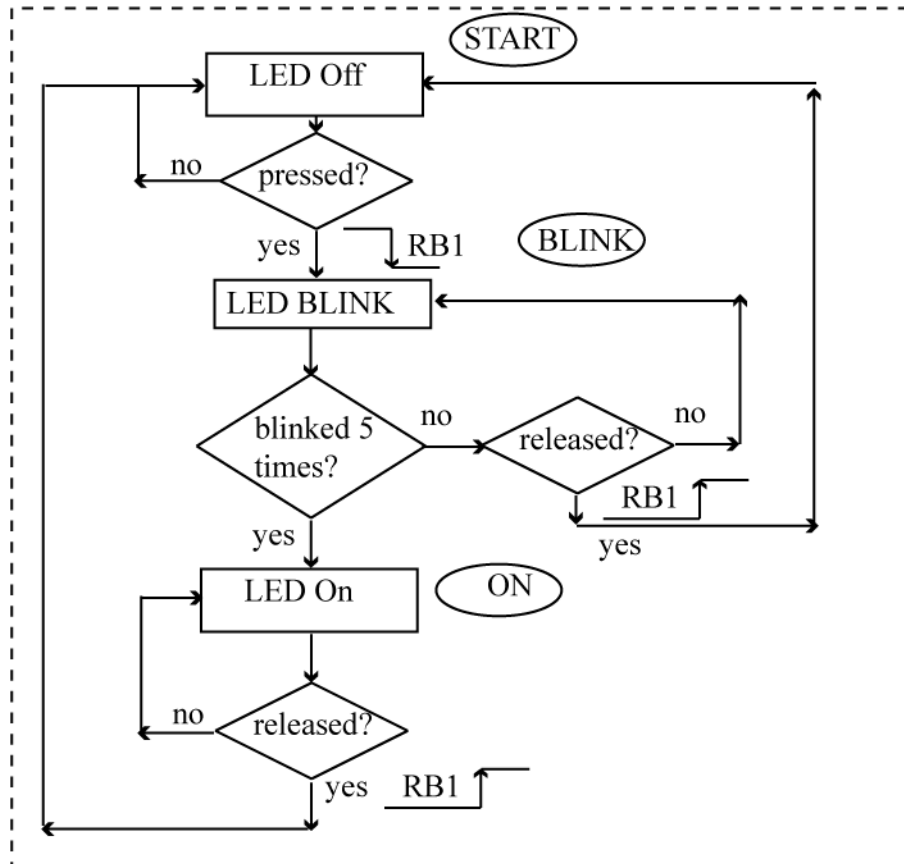




Problem (a)



Problem (b)



Problem (c)

