CIRCLE ONE: Reese Section / Jones Section	MSU Net ID:_____

ECE 372 Test #3 – Spring 2006

Part I: (72 pts)

a. (5 pts) Write *C* code that configures PORTB for the IO shown in the figure for problem (a) on the Figure sheet. The internal weak pullup must be enabled. Do not assume any default bit values.

b. (15 pts) Assuming the IO configuration of the previous problem, write a *while(1){}* loop that implements the LED/Switch IO state machine shown for problem (b) in the figures. Either use a *switch()* statement approach or a *if-then-else* approach. Assume you have available the *DelayMs()* function for blinking the LED; you do NOT have to include debounce delays for the switch input.

c. (20 pts) For the switch configuration shown in Figure (a), implement a program that determines if the switch bounces for a single press and release of the pushbutton. This can be done by determining if more than one rising edge occurs for a single press and release of the pushbutton. Divide your solution into two code segments -- an ISR, and *main()* code. In the ISR, your should increment a variable named *count* each time a rising edge occurs on RB2, and also set a semaphore variable that indicates the interrupt occurred. In the *while(1)* loop of *main()*, you should clear *count* and the semaphore variable, then wait for the semaphore variable to be set, then delay long enough using *DelayMS()* to debounce the switch. If the *count* variable is greater than 1, then switch bounce occurred (print out a message in this case).

1. (13 pts) ISR code

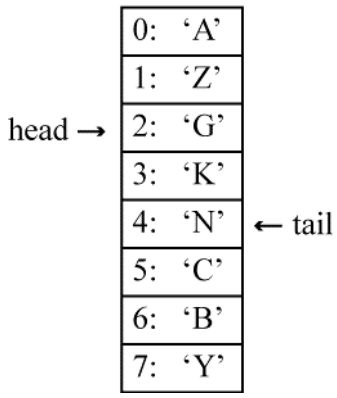
2. (7 pts) *main()* code,be sure to configure and enable your interrupt.

d. (7 pts) Assume an asynchronous serial channel with a data format of 1 start bit, 8 data bits, and 3 stop bits between characters. If I wanted to guarantee that ten characters would be transmitted in 5 ms, what is the MINIMUM baud rate I could use from the standard baud rates of 4800, 9600, 19200, 38400, 57600, 76800 or 115200? You must show your WORK in order to get any credit for this problem. Assume the receiver accepts data as fast as I transmit it.

e. (7 pts) Write *C* code that implements the *void putch(char c)* function (transmits one character to the serial port). *No interrupts are enabled*.

f. (9 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).

Problem (f)



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

_____

f2. From figure F, what character is returned if the buffer is read?

_____ _____

f3. From figure F, what location is modified if one character is written to the buffer? _____

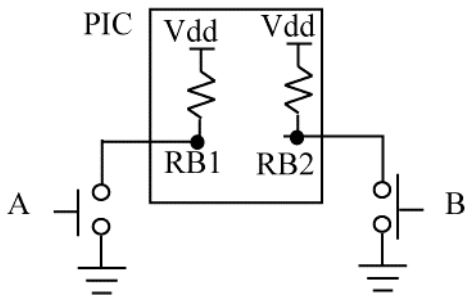
g. (9 pts) Given the code below and Figure (g), answer the questions – assume that the switches do NOT bounce.

g.1 Assume the PIC is powered on, and then Button A is pressed and *released* 3 times. What is the value of *count* after this?

g.2 Assume the PIC is powered on, and then Button A is pressed and *released*, then Button B is pressed and *released*, then Button A is pressed and *held down* – at this point, what is the value of *count*?

g.3 Assume the PIC is then powered on, and Button B is pressed and *released* 3 times. What is the value of *count* after this?

Problem (g)



Part II: (28 pts) Answer 7 out of the next 9 questions. Cross out the 2 questions that you do not want graded. Each question is worth 4 pts.

1. By theory, if the frequency of a CMOS microcontroller like the PIC18 is lowered from 40 MHz to 20 MHz, give the value of the new current draw I_{new} in terms of the old current draw I_{old} if the Voltage is kept constant, and only dynamic power is considered.

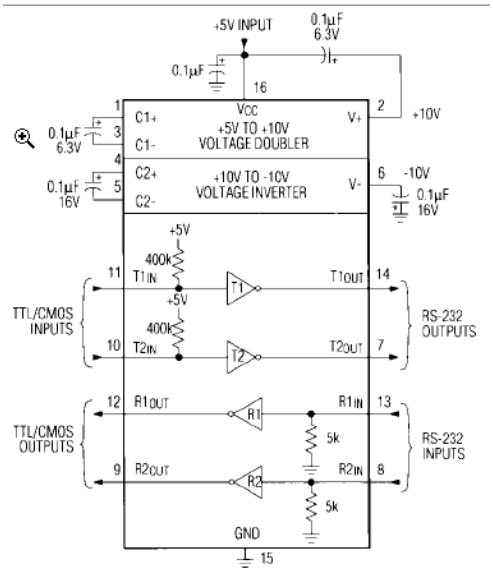
2. Draw a picture that shows how an open-drain output differs from a normal CMOS output.

3. Assume a low-true pushbutton is connected to RB0 and that the INT0 interrupt is initially configured as a falling edge triggered interrupt and is enabled. Fill in the ISR below such that EACH edge of a push and release is detected, and that after 10 edge detections the interrupt is disabled.

4. How many COMPLETE characters be received into the PIC USART before the OERR bit is set? Give the exact definition of when a framing error is detected.

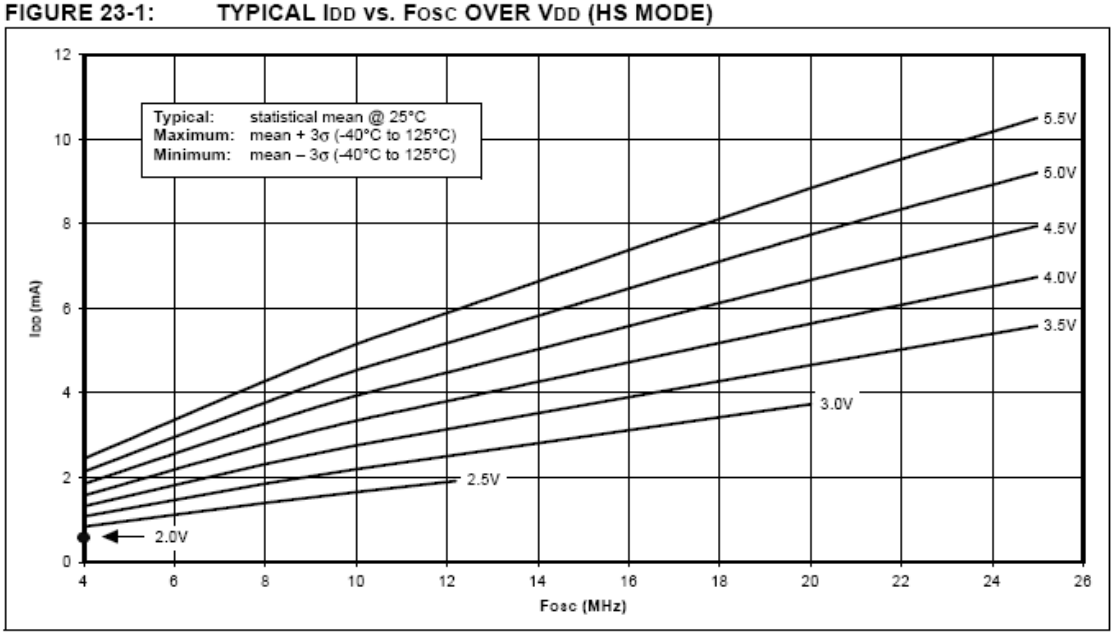
5. In the code below, what will happen assuming the standard PIC18 setup that you have been using in lab?

6. Asssume that the PIC18 TX pin is connected to the *T1in* pin of the RS232 transceiver on the right. What voltage transition would you expect to see on the *T1out* pin (from "??V" to "??V") when a start bit is sent? What voltage transition would you expect to see on the *T1in* pin (from "??V" to "??V") when a start bit is sent?



7. Assume the FOSC of your PIC18 system is 18 MHz and the VDD is 5.0V. Assume that you are measuring the total current draw of your breadboard with an RS232 interface and a power-on LED and it is 35 mA when the PIC is operating normally. What should you expect the new total current draw to be if you place the PIC in sleep mode. Use the graph below to explain your answer.

FIGURE 23-1: TYPICAL IDD vs. FOSC OVER VDD (HS MODE)

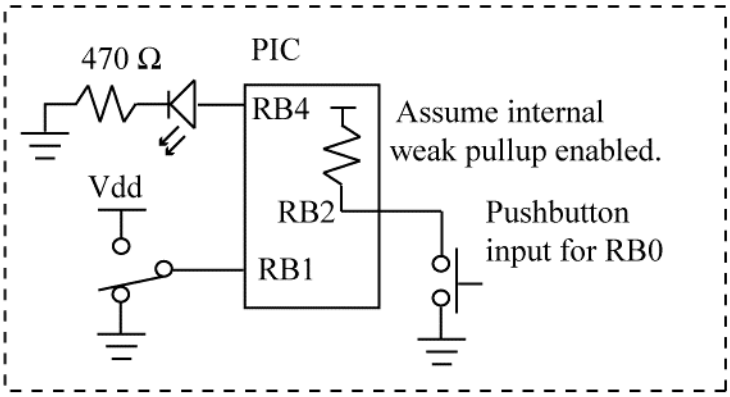


8. Assume that the MSB of the 8-bit value 0x81 is a parity bit for the remaining seven bits. Is this even or odd parity? If the lower seven bits (0x01) were actually received as "0x04", would the parity bit detect this transmission error? (Explain your answer!)

9. When a PIC18 interrupt occurs, what registers are automatically saved by the PIC18 before the ISR is entered? For a high priority interrupt, the PIC also clears the GIE bit to '0' before the ISR is entered; why is this done?

Figures

Problem (a)



Problem (b)