Jonathan W. Valvano First Name: _______________ Last Name:_____________________
May 10, 2001, 7-10pm

This is an open book, open notes exam. You may put answers on the backs of the pages, but please don't turn in any extra sheets. You will write the software in C. You have 3 hours, so please allocate your time accordingly.

(20) Question 1. Count the number of falling edges of an input signal connected to PH2 using interrupts.

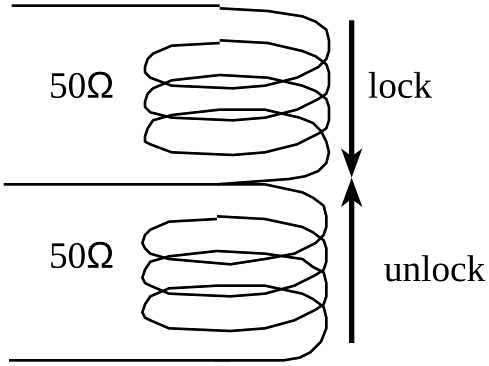
```
unsigned short counter;
```

Part a) Show the C code to initialize Port H bit 2 (PH2) to interrupt on the fall of this input signal. Make the ritual friendly. You do not have to write the main program, which will call this ritual, then perform other unrelated tasks. Use at least one `#define` to make your code more readable.

Part b) Show the ISR that increments the `counter` on every fall of PH2.

Part c) Is the *read-modify-write* sequence in the above ISR a critical section? Explain.

(15) Question 2. Interface a door lock/unlock double solenoid to the microcomputer. The two coild have three wires. To lock the door, place a voltage between the top and middle wires to generate a downward current through the top solenoid coil. To unlock the door, place a voltage between the bottom and middle wires to generate an upward current through the bottom solenoid coil. The impedance of each solenoid coil is 50 Ω, and the inductance is huge. A coil voltage between 3 and 6 volts for 1 second is required for activation. Normally, there are no currents through either coil.
Part a) Show the hardware interface. Label all chip numbers and component values. NOTICE THE DIRECTION OF THE CURRENTS.



Part b) Show the ritual that initializes the interface.
Part c) Show a `lock()` function that energizes the lock solenoid coil, waits exactly 1 second, then turns off all current. You must use the TCNT timer system and gadfly synchronization. Develop helper functions as needed.
Part d) Show an `unlock()` function that energizes the unlock solenoid coil, waits exactly 1 second, then turns off all current. Again, you must use the TCNT timer system and gadfly synchronization.

(15) Question 3. The SPI port of one 6812 is connected to the SPI port of another 6812 in a master/slave configuration. In particular, PS7, PS6, PS5, PS4 of one 6812 are connected to PS7, PS6, PS5, PS4 of the other.
Part a) Characterize the type of data communication possible (circle your choice):

1) data can only flow from master to slave

2) data can only flow from slave to master

3) data can only flow in both directions between the master and slave

Part b) You wish to maximize bandwidth. Interrupts will not be used. There are many possible configurations, but the hardware connections can not be changed. Give the initialization values *in hexadecimal* for each SPI.

	Master	Slave
DDRS		
SP0CR1		
SP0CR2		
SP0BR		

(10) Question 4. Assume an expanded mode 6812 is initialized to have 3 stretches (the total cycle time is 500 ns).
Part a) Determine *read data required* (actual numbers from the beginning of the cycle)

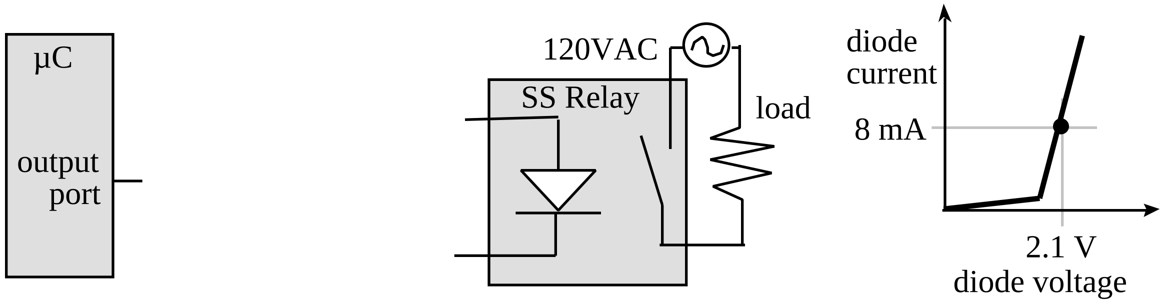
Part b) Determine *write data available* (actual numbers from the beginning of the cycle)

(5) Question 5. Why don't you disable interrupts at the start of an ISR by executing `sei`, and re-enable interrupts at the end of the ISR by executing `cli`?

(15) Question 6. Write three C functions that implement a FIFO queue with the following specifications. The FIFO size is 256 16-bit entries. You can optimize the code assuming the size will always remain at 256 entries. You must use an *indexed* scheme with exactly these global definitions and prototypes. No other globals are allowed.

```
static unsigned char PutI;   // Index of where to put next
static unsigned char GetI;   // Index of where to get next
static unsigned short Size;  // Number of elements currently in the FIFO
                             // FIFO is empty if Size is 0.  FIFO is full if Size is 256
static short Fifo[256];      // The statically allocated fifo data
void InitFifo(void);         // initialize
int PutFifo (short data);    // return -1 if successful, 0 if couldn't put because it was full
int GetFifo (short *datapt); // return -1 if successful, 0 if couldn't get because it was empty
```

(10) Question 7. A solid state relay can be used to switch 120 VAC power to a load. For example, the software can turn on/off lights. To activate the relay (apply power to the load), you must deliver about 2.1 V at 8 mA to the control diode. To deactivate the relay, the diode current should be zero. Build the hardware interface between the microcomputer and the relay. Label all chips and component values.



(10) Question 8. You can answer in sentence form or as C code.
Part a) How do you acknowledge an SCI RDRF interrupt? (i.e., clear RDRF)

Part b) How do you acknowledge an SCI TDRE interrupt? (i.e., clear TDRE)

Part c) How do you acknowledge an output compare channel 7 interrupt? (i.e., clear C7F)

Part d) How do you acknowledge a TOF interrupt? (i.e., clear TOF)

Part e) How do you acknowledge an RTI interrupt? (i.e., clear RTIF)