Jonathan W. Valvano First Name: _____ Last Name:_____
May 10, 2002, 9am 12 noon

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. Assume there are 8 digital signals connected to PORTJ. All PORTJ pins will be inputs and will be used to detect rising and falling edges of these signals. A key wakeup interrupt should be requested on the rising edge of PJ7, PJ6, PJ5 or PJ4. An interrupt should also be requested on the falling edge of PJ3, PJ2, PJ1 or PJ0. The ISR will simply count the number of these edges.

```
unsigned short Count;
```

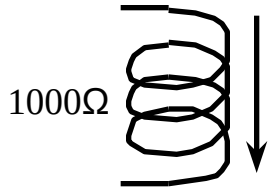
Part a) Show the ritual (arm, initialize `Count`, and enable).

Part b) Show the PORTJ key wakeup ISR. Pay careful attention to consider the case where two or more edges occur at approximately the same time. For example, if PJ0 falls and PJ7 rises, then `Count` should be incremented twice.

```
#pragma interrupt_handler KeyWakupJhandler()
void KeyWakupJhandler(void){
```

```
#pragma abs_address:0xffd0
void (*the_vector[])() = { KeyWakupJhandler}
```

(10) Question 2. Design the hardware interface for one coil of a stepper motor. The motor will require four identical circuits, but you have to show only one interface. Label all chip numbers and component values. The coil resistance is 1000 ohms. The operating point for the coil is 24V and 24mA. No software is required for this problem, just hardware.



(10) Question 3. Design a finite state machine that will be used to control the speed and direction of a stepper motor. The rotation speed is fixed at 1 rps. There is a single binary input, which will control the direction, where 1 means CW and 0 means CCW. There are four binary outputs to the stepper motor. The half-stepping technique will be used to control the stepper. In particular, for a clockwise (CW) rotation, the output sequence should be 5,4,6,2,10,8,9,1. Each new output in the sequence causes one half-step to occur. For a counterclockwise (CCW) rotation, the output sequence should be reversed: 1,9,8,10,2,6,4,5. There are 400 half-steps per rotation. Draw the finite state machine graph. Each state has a 1) state name, 2) a 4-bit output, 3) time to wait, and 4) two next state arrows (one arrow for input 0 and the other arrow for input 1.) No software is required, just draw the FSM graph.

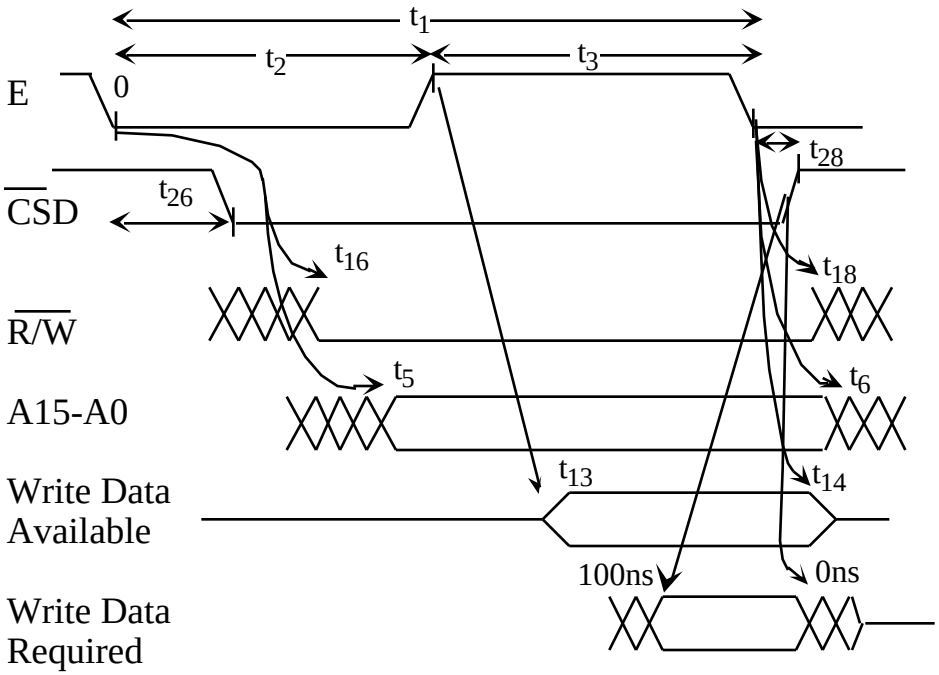
(10) Question 4. Consider the SCI interrupting system shown as assembly Programs 7.10 through 7.12 in the book. Part a) Translate the 6812 assembly `OutChar`, show in Program 7.11, into C code. You don't have to write the Fifo routine, just call the function. Hint: this is very similar to `SCI_OutChar` within the SCI12A.C file.

Part b) What horrible crash situation might occur if this function is called from the background. In particular, explain the sequence of events that cause a software crash.

(5) Question 5. A 10-bit ADC has an input range of –5V to +5V. What is the ADC resolution?

(5) Question 6. A real-time DAS samples its ADC 100 times per second. What range of frequencies is reliably represented in the digital samples?

(10) Question 7. There is a RAM chip interfaced to the 6812, running in expanded narrow mode. CSD is connected to the RAM chip enable. R/W is connected to the RAM write enable, and the RAM output enable is grounded. The following timing diagram shows the write cycle interface between the 6812 and a memory. The memory setup time is 100 ns and its hold time is 0.



Part a) What edge writes data into the RAM? Be as specific as possible.

Part b) Just considering the write cycle, how many cycle stretches are required?

(10) Question 8. Consider a system with RTI periodic interrupts. Once initialized the RTI interrupts will be requested periodically without stopping or pausing. **Circle** the following operations that occur as a typical RTI interrupt is processed. Don't worry about the sequence, just specify which events occur. Include only those operations that occur for each RTI interrupt, and not those operations that occur once in the ritual. Some of these operations may not occur.

A) The RTIF flag is armed as a result of executing a specific instruction (e.g., `RTICTL=0x86;`);

B) The RTIF flag is disarmed as a result of executing a specific instruction (e.g., `RTICTL=0x00;`);

C) The RTIF flag is set as a result of executing a specific instruction (e.g., `RTIFLG=0x80;`);

D) The RTIF flag is set as a result of the RTI hardware;

E) The `rti` instruction is executed;

F) The `cli` instruction is executed;

G) The `sei` instruction is executed;

H) The I bit is automatically set (I=1) by the hardware (not the result of executing a specific instruction);

I) The I bit is automatically cleared (I=0) by the hardware (not the result of executing a specific instruction);

J) The PC, Y, X, A, B, CCR registers are pushed on the stack;

K) The RTIF flag is cleared as a result of executing a specific instruction (e.g., `RTIFLG=0x80;`);

L) The RTIF flag is cleared as a result of the RTI hardware.

(10) Question 9. Consider the case of the serial port interrupt-driven input interface as implemented with SCI12A.C, which is similar to Figure 4.4 in the book. The **RxFifo** queue is used to buffer the data between the background (hardware producer which is the RDRF interrupt service routine) and the foreground (software consumer which is the function `SCI_InChar`). The average serial port input rate is 10 bytes/sec. I.e., the human operator can type 10 characters/sec. The average time for the foreground software to process each character is 1 second. I.e., the average rate at which the foreground calls `SCI_InChar` is 1 Hz. This system does not work properly, and the purpose of this question is to evaluate the problem, and suggest a correction.

Part a) Is the system I/O bound or CPU bound?

Part b) Are any data lost? If so, explain what happens to cause data to be lost. If data is not lost, explain what the problem is.

Part c) Can the problem be fixed by increasing the RxFifo size? If not, suggest an effective way to correct the problem.

(10) Question 10. Place the letter code in the empty box for the best definition for the following terms

	asynchronous
	atomic
	critical
	noninstrusive
	open collector
	private
	promotion
	real-time
	stabilize
	synchronous

(A) A debugging technique that specifies all its inputs, so, the output results are consistently repeatable.
(B) A digital logic output that has two states low and HiZ.
(C) A protocol where the two devices have separate and distinct clocks
(D) A protocol where the two devices share the same clock.
(E) An object that can be accessed only by software modules in that group.
(F) Increasing the precision of a number for convenience or to avoid overflow errors during calculations.
(G) Locations within a software module, which if an interrupt were to occur at one of these locations, then an error could occur
(H) Software execution that can't be divided or interrupted.
(I) The collection of information itself does not affect the parameters being measured.
(J) There is an upper bound on the delay between when software is supposed to be run and when it is actually run.