Jonathan W. Valvano First:_________________ Last:_____________________

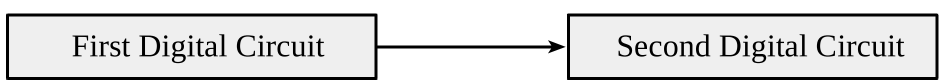
This is the closed book section. You must put your answers in the boxes on this answer page. When you are done, you turn in the closed-book part and can start the open book part.

Addr	Bit 7	6	5	4	3	2	1	Bit 0	Name
\$00C8	BTST	BSPL	BRLD	SBR12	SBR11	SBR10	SBR9	SBR8	SCIBD
\$00C9	SBR7	SBR6	SBR5	SBR4	SBR3	SBR2	SBR1	SBR0	
\$00CB	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	SCICR2
\$00CC	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF	SCISR1
\$00CF	R7T7	R6T6	R5T5	R4T4	R3T3	R2T2	R1T1	R0T0	SCIDRL

(4) Question 1. Write a C function that outputs (transmits) one byte using the SCI port. You may assume the SCI is already initialized. Use busy-wait synchronization.

(4) Question 2. Give an example C code that has a ***dropout*** error.

(4) Question 3. Consider the situation in which the output of one digital circuit is connected to the input of another digital circuit. There are no other connections on this signal, i.e., one output is tied to one input. The output specifications of the first circuit are V_{OH}, V_{OL}, I_{OH} and I_{OL}. The input specifications of the second circuit are V_{IH}, V_{IL}, I_{IH} and I_{IL}. These are the specifications, like you would find in a data sheet, not actual measurements of voltage and current like you would measure in lab with a DVM. Give the four **inequalities** relating these eight parameters that must be true in order for the interface to operate properly.



V_{OH} V_{IH}	I_{OH} I_{IH}
V_{OL} V_{IL}	I_{OL} I_{IL}

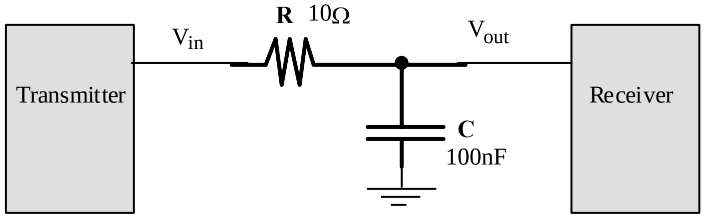
Address	Bit 7	6	5	4	3	2	1	Bit 0	Name
$00D8	SPIE	SPE	SPTIE	MSTR	CPOL	CPHA	SSOE	LSBF	SPICR1
$00D9	0	0	0	MODFEN	BIDIROE	0	SPISWAI	SPC0	SPICR2
$00DA	0	0	0	0	0	SPR2	SPR1	SPR0	SPIBR
$00DB	SPIF	0	SPTEF	MODF	0	0	0	0	SPISR
$00DD	Bit 7	6	5	4	3	2	1	Bit 0	SPIDR

(4) Question 4. Write a C function that outputs (transmits) one byte using the SPI port. You may assume the SPI is already initialized. Don't worry about SS. Use busy-wait synchronization.

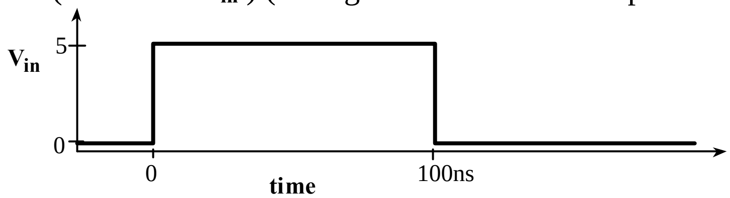
(4) Question 5. A software system needs to store a non-integer value. The parameter can be any value from -100.000 to +100.000 with a resolution of 0.001. What C data type do you need to define the integer portion of the fixed-point number?

(4) Question 6. Consider a real-time system that samples an ADC. The goal is to implement a sampling frequency of $\mathbf{f_s}$. Let $\mathbf{t_i}$ be the actual time sample **i** is taken. Give a mathematical equation, defining sampling jitter, $\boldsymbol{\delta t}$ in terms of $\mathbf{f_s}$ $\mathbf{t_{i-1}}$ and $\mathbf{t_i}$.

(8) Question 7. The SPI ports of two 9S12s are connected with a VERY long cable. We will model this cable as a single resistor in series with a capacitor, as shown in the figure below.



For this question, assume an ideal transmitter (output impedance of 0) and an ideal receiver (input impedance of infinity). Let **R**=10Ω, and **C**=100nF. Note that R*C is 1µs. Consider a 5V 100ns pulse on the output of the transmitter (labeled as $\mathbf{V_{in}}$) (as might occur with a 5 Mbps SPI transmission)



Derive an equation for $\mathbf{V_{out}}$ as a function of time for the first 100 ns. Show your work and plug in values for R*C is 1µs.

\

(4) Question 8. This is a *performance debugging* question. The debugging instrument needs to be *minimally intrusive*. Assume **x** is an unsigned 16-bit variable in a real-time system. The goal is to collect information about what the values of **x** are and at what times **x** was those values. The SCI, **PTT** and **PTAD** are unused by the system, and **PTT** and **PTAD** are digital outputs. The debugging code will be placed at strategic places in the system. **SCI_OutUDec** outputs a 16-bit unsigned integer. **BufX** and **BufT** are 16-bit global buffers of length 100, **n** is a global variable initialized to 0. Which debugging code would you add to verify the correctness of this system? Circle your answer

A) `if(n<100){BufT[n]= TCNT; BufX[n]=x; n++;}`

B) `PTT=TCNT; PTAD=x;`

C) `SCI_OutUDec(TCNT); SCI_OutUDec(x);  // busy-wait`

D) `SCI_OutUDec(TCNT); SCI_OutUDec(x);  // interrupt driven`

E)
```
PTT |= 0x01;    // at beginning
PTT &= ~0x01;   // at end
```

F) `if(n<100){BufX[n]= x; n++;}`

(4) Question 9. List three rationalizations (faulty justifications) an engineer might have for being unethical. I.e., why would an engineer be unethical?

1
2
3

For one of these rationalizations, explain how the rationalization is flawed.

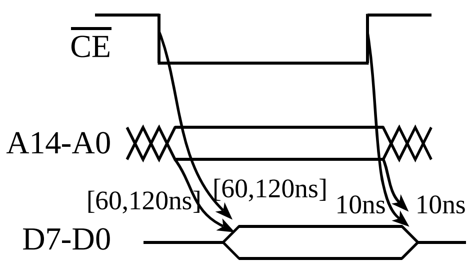
end of closed book section

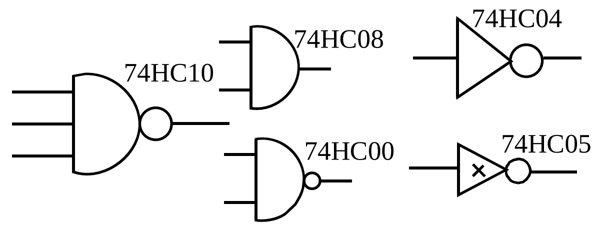
Jonathan W. Valvano First:__________________ Last:______________________

Open book, open notes, calculator (no laptops, phones, devices with screens larger than a TI-89 calculator, devices with wireless communication). You must put your answers on pages 5 to 10. Please don't turn in any extra sheets.

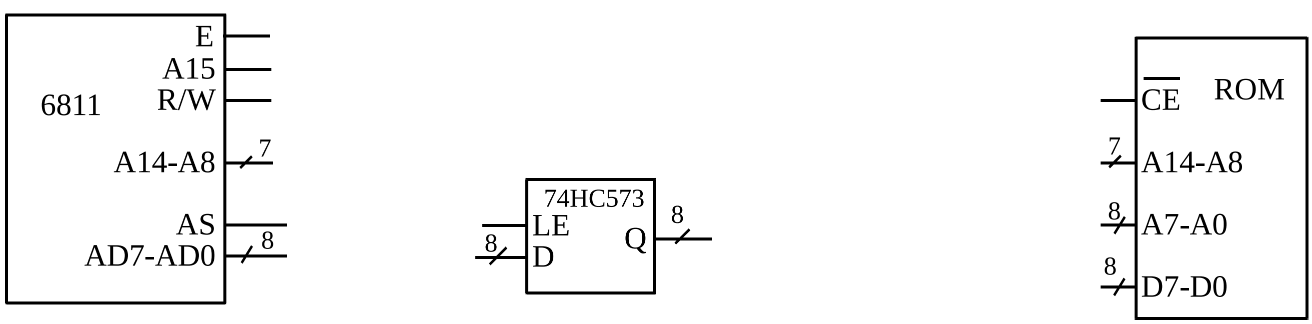
(10) Question 10. The goal of this question is to interface a DC motor. PT0 controls the On/Off and PT1 controls the direction of rotation. The software will output a digital low on PT0 to deactivate the motor (current = 0). The software will output (PT1=0, PT0=1) to spin in one direction and it will output (PT1=1, PT0=1) to spin in the other direction. To activate the motor you must deliver about 5V to the motor coil. The maximum current of the coil is 500 mA (at maximum torque) and its inductance is 0.001 mH. Show the circuit diagram that interfaces PT1, PT0 to the motor. There are two power supplies to choose from +5V and +12V. Give part numbers and resistor values.

(15) Question 11. Interface the following 32K ROM to a 6811 running at **1 MHz.** Assume the gate delay through each 74HC digital logic gate is [10ns min, 20ns max]. $\overline{CE}$ is negative logic. Assume there is no internal EEPROM or boot loader at BFxx. The full address decoder should select addresses $8000 to $FFFF. The ROM timing is described in the following figure:





Part a) Design the interface between the ROM to the 6811. You are limited to the digital devices shown on this page (you can use more than one copy if you want)

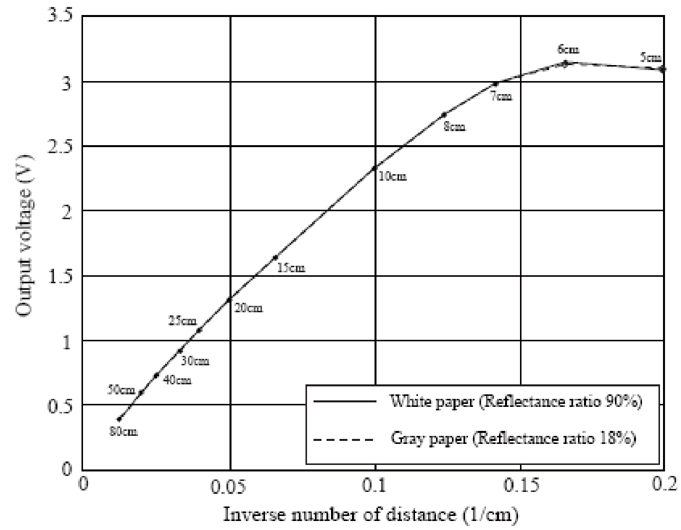


Part b) Since the E clock period is 1000 ns, the read data required is (950,1010). Calculate the worst-case **Read Data Available** interval for this interface and show that the timing requirements are satisfied.

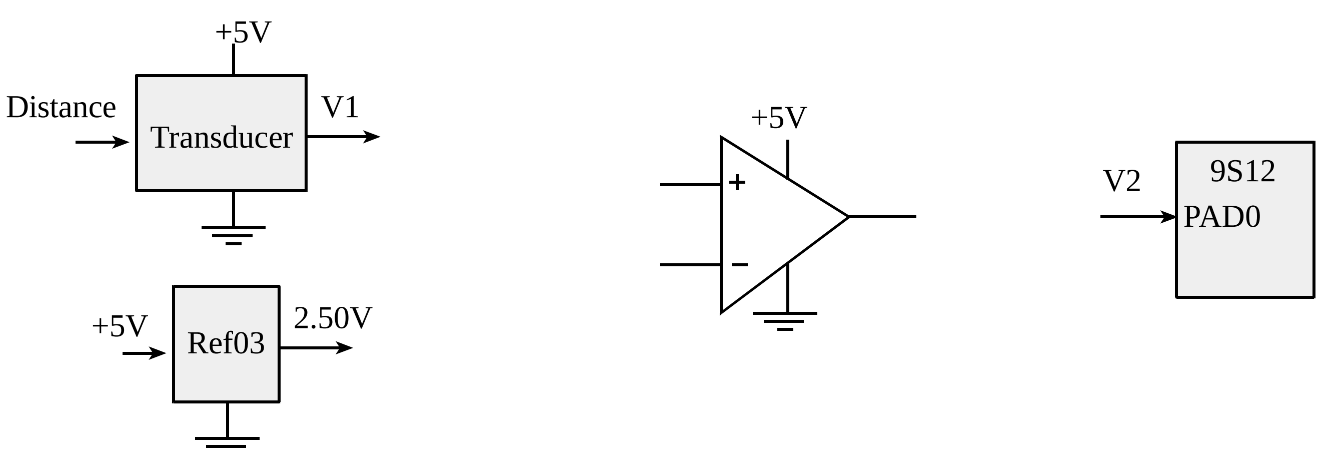
(10) Question 12. You will use a Sharp GP2Y0A21YK0F infrared object detector to measure distance from about 7 to 80 cm. This sensor creates an analog voltage that depends inversely on distance to object. You may approximate the response as

V1 = 0.25 + 20/**D**,

where **D** is distance in cm and **V1** is the transducer output in volts. The transducer output (V) is a single-ended voltage, with a range between 0.5 and +3V. Part a) Your analog circuit must map the 0.5 to 3V range of the transducer into the 0 to +5V range of the ADC. Derive a mathematical function relating the desired ADC input, **V2**, to the transducer output, **V1**.



Part b) Show resistor values, but not pin numbers. Just show the analog interface (no software required). No antialiasing analog low pass filter is required. Use one Ref03 2.50V reference and one rail to rail +5V op amp (like a TLC2272 or Max495).



(15) Question 13. The goal of this problem is to interface a joystick to the 9S12. The hardware consists of four binary switches (labeled U R D L). The joystick can be in one of 9 positions, as shown below. The software system checks the joystick 10 times a second. Every 100 ms, a joystick **`code`** (0 to 8) is placed in a FIFO by calling **`Fifo_Put`**. Any of the other 7 switch patterns means the joystick is broken, and the interface should return a **`code`** of 255.

Joystick position	Switch U	Switch R	Switch D	Switch L	**`code`**
Center	Off	Off	Off	Off	0
Up	On	Off	Off	Off	1
Up and Right	On	On	Off	Off	2
Right	Off	On	Off	Off	3
Down and Right	Off	On	On	Off	4
Down	Off	Off	On	Off	5
Down and Left	Off	Off	On	On	6
Left	Off	Off	Off	On	7
Up and Left	On	Off	Off	On	8

The 8-bit FIFO is available (you do NOT need to write the FIFO code) with these prototypes

```
void Fifo_Init(void);              // Initialize the Fifo
void Fifo_Put(unsigned char data); // Put data into fifo
unsigned char Fifo_Get(void);      // Get data from fifo
```

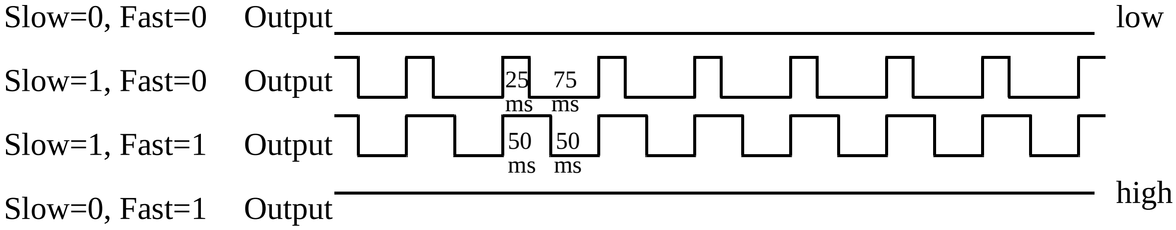
If the FIFO is full on a put, then data is lost and the routine returns immediately. If the FIFO is empty on a get, then the routine will wait with interrupts enabled until data arrives.

Part a) Show the hardware interface between the four switches and Port M of the 9S12. Periodic interrupts on output compare 7 will be used (no input capture, and no key wakeup)

Part b) Show the initialization code for the system, enable and arm output compare 7. Initialize Port M. Initialize any data structures, like the FIFO, that you need. You may assume the PLL is not active so the E clock is 4 MHz.

Part c) Show the output compare interrupt 7 ISR. Each interrupt will put exactly one element into the FIFO (0 to 8). Full credit will be given to the solution that defines a data structure, so that NO **`if`** or **`switch`** statements are used. I.e., the ISR has no branching.

(10) Question 14. An FSM will be used to control the power delivered to a motor. There are two inputs, called **Fast** and **Slow** and one output, **Output**. If **Fast** and **Slow** are both 0, the **Output** is 0. If **Slow** is 1 (and **Fast** is 0), the **Output** is a 10 Hz 25% duty-cycle squarewave. If **Slow** and **Fast** are both 1, the **Output** is a 10 Hz 50% duty-cycle squarewave. If **Fast** is 1 (and **Slow** is 0), the **Output** is 1.



Draw the Moore FSM graph. Give descriptive names for the states (not S1 S2, not SA SB). No C code is required, just draw the state graph. The controller sequence repeats these four operations: output, wait, input, next.