Jonathan W. Valvano First:__________________ Last:______________ EID:___________

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

(2) Question 1. Bluetooth low energy (BLE)

(2) Question 2. RS485, Ethernet, CAN, and USB.

(2) Question 3. Boost convertor.

(2) Question 4. Tantalum capacitors.

(2) Question 5. Non-compete.

(2) Question 6. CMRR.

(2) Question 7. DAC monotonicity.

(5) Question 8. Instrumentation amp over an op amp.

(5) Question 9. UART baud rate.

(4) Question 10a. Critical section (given exact memory addresses). ………………

(4) Question 10b. How to fix critical section (A-F). …………………………………

(5) Question 11. DAC techniques, enter **RS**, **R-2R**, **BW**, and/or **none**

A) Which works well for a high-precision DAC (12 bits and over)? --------------------

B) Which is best for low-precision, low-cost sampling? --- (one answer) --------------

C) Which is used in the TLV5616 (Lab 5)? ----------------- (one answer) ---------------

D) Which has a cost/complexity linearly related to the number of bits? -----------------

E) Which has a cost/complexity exponentially related to the number of bits? ---------

(8) Question 12. *area* equals *width* times *length*. Show C code

(5) Question 13. Characterize the system (specify A-G). You can have multiple …

(2) Question 1. Give an important mechanism **Bluetooth low energy (BLE)** uses to achieve low energy.

(2) Question 2. Explain why communication channels like RS485, Ethernet, CAN, and USB all use **differential drive outputs**?

(2) Question 3. Briefly explain how a **boost convertor** uses an inductor to operate.

(2) Question 4. Why do we **not** use tantalum capacitors for analog high pass filters? I.e., which capacitor parameter determines why we use ceramic rather than tantalum capacitors for high pass filters?

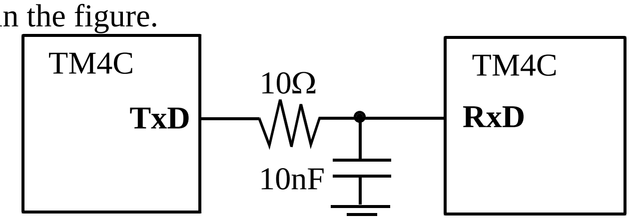
(2) Question 5. You are CEO of a company. Why would you have your employees sign a "**non-compete clause**" (NCC) as a requirement for employment?

(2) Question 6. Why does a state of the art cell phone employ **modular** design? In particular, how is a cell phone fundamentally different from an EE445L lab, such that modular design is necessary?

(2) Question 7. Briefly define DAC **monotonicity**. Draw a figure to explain.

(5) Question 8. You are asked to design a linear analog amplifier using rail-to-rail components for an embedded system powered with a single 3.3V power. You want it to be both effective (implement the desired transfer function) and to be low cost. What criteria in the problem statement would lead you to choose an instrumentation amp like the INA122 ($6.32) over an op amp like the OPA2350 ($4.23)?

(5) Question 9. This problem addresses the issue of using a 100-foot cable to implement UART serial communication between two microcontrollers. To implement simplex communication, the cable has two wires: signal and ground. The transmit UART pin of one TM4C123 is connected via the 100-foot cable to the receive UART pin of a second TM4C123. As the cable increases in length so will the effective resistance and capacitance of the cable. This low quality cable has a capacitance of about 100 pF/foot; so we will assume this cable has an effective 10-nF capacitance between signal and ground. 30-gauge wire has a resistance of about 0.1Ω/foot; so we will assume this cable has an effective 10-Ω resistance between output and input, as shown in the figure.



Estimate the fastest baud rate that can be implemented reliably. Show both your work and your estimate of maximum baud rate in bits/sec.

(8) Problem 10. Consider the following SysTick interrupting system with its corresponding assembly code. You may assume SysTick interrupts occur slowly enough that SysTick will not attempt to interrupt itself. The listing includes absolute addresses. ROM exists from 0x00000000 to 0x0003FFFF. **`Count`** is a 32-bit variable in RAM. There are one or more critical sections.

Part a) Give the exact ROM location(s) of the critical section(s). For example, if you were to answer 0x04C4-0x04C8, then you would mean a mistake could happen if the interrupt occurred between the **`PUSH`** and **`MOVS`** instructions, or between the **`MOVS`** and **`BL`** instructions of the SysTick handler.

```
volatile int32_t Counts = 0;


void static Add(int32_t n){
  Counts = Counts + n;
}

void SysTick_Handler(void){
  Add(1);
}

int main(void){
  Init(); // includes SysTick_Init
  EnableInterrupts();
  while(1){
    Add(-1);
  }
}
```

```
EnableInterrupts:
0x00000324       CPSIE I
0x00000326       BX    lr

DisableInterrupts:
0x00000328       CPSID I
0x0000032A       BX    lr

Add:
0x000003C4       LDR   r1,=Count
0x000003C6       LDR   r1,[r1,#0x00]
0x000003C8       ADD   r1,r1,r0
0x000003CA       LDR   r2,=Count
0x000003CC       STR   r1,[r2,#0x00]
0x000003CE       BX    lr

SysTick_Handler:
0x000004C4       PUSH  {lr}
0x000004C6       MOV   r0,#0x01
0x000004C8       BL    Add
0x000004CC       POP   {pc}

main:
0x00000510       BL    Init
0x00000514       BL    EnableInterrupts
0x00000518       MOV   r0,#0xFFFFFFFF
0x0000051E       BL    Add
0x00000522       B     0x00000518
```

Part b) How would you solve this critical section? Your solution must maintain overall function of the system, such that **`Count`** will be the difference between ISR executions and main loop executions.

A) Move **`Count`** to be a local variable of the function **Add**

B) Disarm SysTick and rearm SysTick around the critical section(s)

C) Remove the **`volatile`** designation on **`Count`**

D) Remove the **EnableInterrupts** call from **main**

E) Add **DisableInterrupts** at beginning and **EnableInterrupts** at end of **SysTick_Handler**

F) Remove the **`static`** designation from the function **Add**

G) None of the above will remove the critical section

(5) Question 11. Consider three different DAC techniques: resistor string, R-2R ladder and binary weighted. Pick the DAC technique that best answers each question. Place an **RS** for resistor string, an **R-2R** for R-2R ladder, or a **BW** for binary weight. If two answers are ok, list both. If no answer is correct, list **none**.

(8) Question 12. You will use **binary fixed-point** to implement *area* equals *width* times *length*. Assume *width* and *length* are fixed-point numbers with 2^{-8} cm resolution; **`W`** and **`L`** are the integer parts respectively. Assume *area* is a fixed-point number with 2^{-10} cm^2 resolution; **`A`** is the integer part of *area*. Write **C code** that has **`W`** and **`L`** as inputs and has **`A`** as an output. Minimize dropout, ignore overflow.

(5) Problem 13. The SysTick background thread runs at 22 kHz. The ADC is set to 125 kHz. The FIFO is created with the index macro in Program 3.10 from the book. The system streams data from ADC to DAC.

```
#define FAIL 0
#define SUCCESS 1
AddIndexFifo(D,32, uint32_t, FAIL, SUCCESS);
uint32_t ADClost,DAClost; // error counts
void SysTick_Handler(void){uint32_t data;
  if((DFifo_Get(&data)==SUCCESS){
    DAC_Out(data); // pass data from ADC to DAC
  }else{
    DAClost++; // no data for DAC because empty
  }
}
void main(void){uint32_t data;
  PLL_Init();      // bus clock at 80 MHz
  DAC_Init();      // enable TLV5616 DAC using SPI
  ADC_Init();      // initialize TM4C123 12-bit ADC, busy wait
  SysTick_Init();  // SysTick running at 22 kHz
  DFifo_Init();    // length 32, size 32 bits
  ADClost=DAClost=0;
  EnableInterrupts();
  while(1){
    data = ADC_In(); // busy wait sample (no hardware averaging)
    if(DFifo_Put(data)==FAIL){ // stream data to background
      ADClost++; // ADC data lost because FIFO full
    }
  }
}
```

How would you characterize the system implemented above? List all that apply

A) The FIFO fills up and ADC will be lost (**`ADClost`**>0)
B) The use of the FIFO represents nonreentrant code
C) The FIFO will become empty and some DAC outputs will be skipped (**`DAClost`**>0)
D) The system is an example of Round Robin execution
E) ADC input is real time
F) DAC output is real time
G) System has a critical section

Jonathan W. Valvano First:__________________ Last:______________ EID:___________

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 these pages. Please don't turn in any extra sheets.

(20) Question 14. The overall goal of this problem is to design a system that sinusoidally oscillates a high-power LED at 1 Hz. In other words, the LED goes bright-dim-bright once a second. You must use the following table to implement 32 different brightness's. For example, 98 means 98% brightness.

```
const uint32_t SinTable[32]={98,97,94,90,84,77,68,59,50,41,32,23,16,
  10,6,3,2,3,6,10,16,23,32,41,50,59,68,77,84,90,94,97};
```

(5) Part a) Show the hardware interface between the microcontroller and the LED. Full brightness occurs at 1.5V 500 mA. The 3.3V is the only power. You may use any pin(s) on the TM4C123. Label all parts and give resistor and capacitor values. Please show your work. Assume the PN2222 has h_{fe}=100, V_{be}=0.6V, V_{ce}=0.3V. Assume the TIP120 has h_{fe}=1000, V_{be}=1V, V_{ce}=0.7V.

(15) Part b) Show all software required to run the system. The body of the main MUST be a do-nothing while loop. You may call any C function listed in the book or lecture notes without showing the body of that function. Use comments to explain your approach.

(15) Question 15. Design a system with one analog input and one digital output. The input impedance must be larger than 1 MΩ. The input range is 0 to 3.3V. If the input is less than 1V, the output is digital low (about 0V). If the input is greater than 1V and less than 2V, the output is digital high (about 3.3V). If the input is greater than 2V, the output is digital low (about 0V). If you use a chip not in the book/lecture notes, please describe its behavior. Label all chips and resistors. You may use any software function listed in the book/lecture notes without showing the body of that function. Show your work.

(10) Problem 16. Draw a Moore FSM graph that has two binary inputs and four binary outputs. For each state show name, 4-bit binary output, time delay in ms, and next state arrows. The machine will control a bipolar stepper motor with 100 steps/rotation. The change {5 to 6} steps the motor once clockwise. The change {5 to 9} steps the motor once counter clockwise. If the input is 00, the motor should stop. If the input is 01, the motor should spin clockwise at 1 rps. If the input is 10 or 11 the motor should spin counterclockwise at 10 rps. The controller pattern will be 4-bit output, wait, input, next. No software or hardware are required, just the FSM graph. Use good state names to clarify operation.

(5) Problem 17. Create two #define MACROs that implement minimally-intrusive debugging instruments. Assume PA7 is already initialized as an output. One #define MACRO sets PA7 and the other clears it. Show the C code. For full credit your solution must be friendly and have no critical sections. Running at 80 MHz, estimate the intrusiveness of these instruments.