Jonathan W. Valvano First:________________ Last:____________________

October 11, 2018, 3:30-4:45pm. This is a closed book exam, with one 8.5 by 11-inch crib sheet. You have 75 minutes, so please allocate your time accordingly. ***Please read the entire quiz before starting***.

(5) Question 1. Consider the IoT system from Lab 4 that sends time information from the TM4C123 to the phone. For this problem, the goal is to display current time on the phone, where the time maintained by the TM4C123 interrupt service routine is defined as truth. The current time is periodically sent from the TM4C123 to the phone as a set of virtual pins. However, because the data are first sent to the ESP8266, then sent to the hot spot, then sent to the Blynk server, and finally sent to the phone, there will be a delay or lag between true time on the TM4C123 and displayed time on the phone. Regardless of this delay, the system has value, but the value of the system increases as this lag decreases. I.e., all things being equal, the system performance improves as the lag gets shorter. **Is this system real time?** If so, what type of real-time system is it? If not real time, explain why it is not real time. Put your answer in the box.

Yes, it is real time
Because value descreases over time, we classify this as soft real time

(5) Question 2. How do you use a voltmeter or a regular oscilloscope (not a spectrum analyzer) to **quantify the amount of noise** on the 3.3V power line?

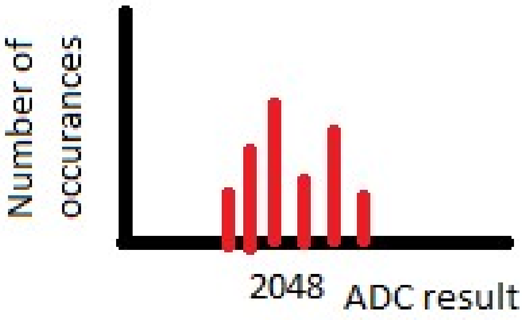
Voltmeter: attach probes across 3.3V and ground, set mode to AC (the meter gives the RMS magnitude of noise in volts)
Scope: attach probes across 3.3V and ground, set mode to AC, use scope functions to perform an RMS measurement (or perform peak-to-peak measurement and multiply by 0.707/2)

(5) Question 3. In order to improve **signal to noise ratio** on data sampled by the ADC we can deploy *over-sampling*. Assume for this problem, hardware averaging is not activated. This means for every one output we need, we will sample the ADC N times. Which of the following operations will improve the signal to noise ratio over sampling just once? There may be multiple answers, if so list all operations that improve the signal to noise ratio.

A) Calculate the maximum of the N samples.
B) Calculate the average difference between successive samples.
C) Calculate the mean of the samples.
D) Calculate the minimum of the N samples.
E) Perform a linear regression and calculate the slope of the data.

C, the average of the samples will improve SNR

(5) Question 4. One can use a **probability mass function** to describe noise on the ADC. Assume the input to the ADC is 1.65V, and therefore the expected ADC sample is 2048. For this question, **draw a pmf graph** illustrating noise typical of the TM4C123, and label the two axes.



(10) Question 5. Consider the following C code (with corresponding compiler generated assembly). Periodic SysTick interrupts occur every 1 ms.

```
0x00000440 4806       LDR   r0,[pc,#24]
0x00000442 6800       LDR   r0,[r0,#0x00]
0x00000444 F0800004   EOR   r0,r0,#0x04
0x00000448 4905       LDR   r1,[pc,#20]
0x0000044A F8C103FC   STR   r0,[r1,#0x3FC]
0x0000044E 4805       LDR   r0,[pc,#20]
0x00000450 6800       LDR   r0,[r0,#0x00]
0x00000452 F1000001   ADD   r0,r0,#0x01
0x00000456 4903       LDR   r1,[pc,#12]
0x00000458 6008       STR   r0,[r1,#0x00]
0x0000045A 4770       BX    lr


0x0000050A 4809       LDR   r0,[pc,#36]
0x0000050C 6800       LDR   r0,[r0,#0x00]
0x0000050E F0800008   EOR   r0,r0,#0x08
0x00000512 4908       LDR   r1,[pc,#32]
0x00000514 F8C103FC   STR   r0,[r1,#0x3FC]
0x00000518 E7F7       B     0x0000050A
```

```
// Interrupt service routine
void SysTick_Handler(void){
  GPIO_PORTF_DATA_R ^= 0x04;
  Counts = Counts + 1;
}


int main(void){
// some initialization code
  while(1){
    GPIO_PORTF_DATA_R ^= 0x08;
  }
}
```

There is a critical section in this code. Assume each instruction takes exactly 2 bus cycles to execute (25ns). Also, assume there are no other interrupts. Estimate the probability that an error will occur due to the critical section for each instance of the SysTick interrupt. Show your work.

Output will be corrupted if the interrupt were to occur between these 3 pairs of addresses
50C-50E, 50E-512, 512-514

Output will be correct if the interrupt were to occur between these 3 pairs of addresses
50A-50C, 514-518, 518-50A

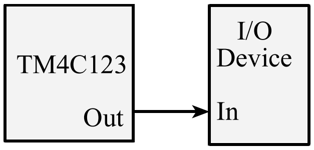
Since we assume each instruction takes the same amount of time, the probability of corruption is 3/6 or 50%

(5) Question 6. Explain how the connection socket and client socket are used the **https://www.blynk.cc/** communication.

Blynk is publicly accessible, so the connection socket (conSocket) is where the server waits for connection requests from clients. Data packets are not sent through the connection socket. The same connection socket is used initially by all clients to initiate and establish the connection. Actual data is sent through a separate dedicated client socket between the server and that one client. Each client has its own client socket.

(10) Question 7. You wish to connect a device to a GPIO output of the TM4C123. These are the parameters of the I/O device:

$I_{IL} = 1\text{mA}$, $I_{IH} = 1\text{mA}$,
$V_{IL} = 0.3\text{V}$, $V_{IH} = 2.0\text{ V}$



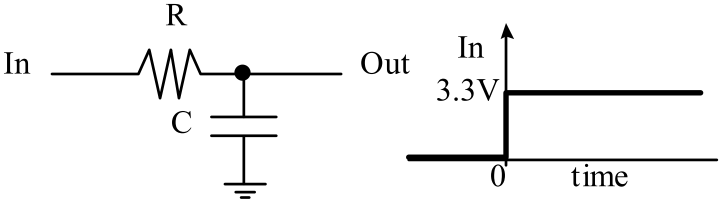
Can you directly connect a TM4C123 output to this device? Select Yes or No:

no

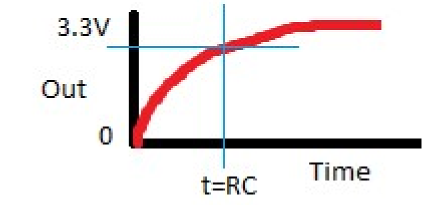
If yes, prove it. If no, show at least one parameter/equation not satisfied.

It doesn't work because the output low voltage of the TM4C123 may be higher than the maximum input low voltage required. $V_{OL} > V_{IL}$
0.4V > 0.3V

(10) Question 8. Consider the following *RC* circuit. The input, **In**, is a 3.3-V step at time 0. Give the equation for **Out** as a function of time, *R*, and *C* (for times greater than 0). This in essence is what we have on every digital signal where we connect an output pin to an input pin. *R* is the output impedance of the output pin, and *C* is the input capacitance of the input pin.



This is a review from EE411 (and commonly asked on interviews)
Out = 3.3V – 3.3V*$e^{-t/RC}$



(25) Question 9. Consider the interaction between these three ISRs.

Define two FIFOs, one for A and one for B

```
void GPIOPortA_Handler(void){
  GPIO_PORTA_ICR_R = 0xFF;
  FifoA_Put(GPIO_PORTA_DATA_R);
}
```

```
void GPIOPortB_Handler(void){
  GPIO_PORTB_ICR_R = 0xFF;
  FifoB_Put(GPIO_PORTB_DATA_R);
```

```
void SysTick_Handler(void){

uint32_t data;
  if(FifoA_Get(&data){
    ProcessA(data);     // Receive and process Port A data
  }
  if(FifoB_Get(&data){
    ProcessB(data);     // Receive and process Port B data
  }
}
```

(20) Question 10. There are 8 digital inputs connected to Port B (assume the inputs do not bounce).

(10) Part a) Complete the initialization routine. You may add globals.

```
void Init((void)*rise(void),(void)*fall(void)){
  SYSCTL_RCGCGPIO_R |= 0x02;
  delay = SYSCTL_RCGCGPIO_R;
  GPIO_PORTB_PCTL_R = 0x00000000;
  GPIO_PORTB_DIR_R  = 0x00;
  GPIO_PORTB_DEN_R  = 0xFF;
  GPIO_PORTB_IS_R  =       ;
  GPIO_PORTB_IBE_R =       ;
  GPIO_PORTB_IEV_R =       ;
  GPIO_PORTB_IM_R =        ;
  NVIC_EN0_R =             ;
```

0x00

0xFF

Irrelevant (any)

0xFF

0x02

```
TheRise = rise;
TheFall = fall;
Last = GPIO_PORTB_DATA_R;
```

```
// Put globals here
// function pointers
void *TheRise(void);
void *TheFall(void);
uint32_t Last;
```

(10) Part b) Show the interrupt service routine

```
void GPIOPortB_Handler(void){
  uint32_t current,change;
  current = GPIO_PORTB_DATA_R;
  change = current^Last; // which bits have changed
  if(change&current){ // changed bits are high means rise
    *TheRise();       // invoke user function call back
  }else{
    *TheFall();       // invoke user function call back
  }
  Last = current;
  GPIO_PORTB_ICR_R = 0xFF; // acknowledge
}

void GPIOPortB_Handler(void){
  uint32_t current;
  current = GPIO_PORTB_DATA_R;
  if(current > Last){ // a bit goes 0 to 1 means rise
    *TheRise();       // invoke user function call back
  }else{
    *TheFall();       // invoke user function call back
  }
  Last = current;
  GPIO_PORTB_ICR_R = 0xFF; // acknowledge
}
```