First:________________ Last:__________________

November 20, 2018, 3:30 to 4:45 pm. Open book, open notes, calculator (no laptops, phones, devices with screens larger than a TI-89 calculator, devices with wireless communication). You have 75 minutes, so please allocate your time accordingly. ***Place your EID at the top of each page. Please read the entire quiz before starting***. We will use **Gradescope**, so you must put your answers in the boxes provided.

(18) Question 1. Put your answers A – S in the boxes. Exactly one letter per box (some letters not used)

A) LM4041, a shunt-diode
B) TPA731, a differential input, Class-AB amplifier
C) TPS78233, a 3.3V linear regulator
D) INA122, an instrumentation amp
E) TPS63001, a 3.3V buck-boost regulator
F) OPA2350, a rail-to-rail op amp

(3) Part a) Which device do you use to create a low noise analog reference voltage?

A) LM4041, a shunt-diode

(3) Part b) Which device do you use to create a +3.3V power source from a 7.4V battery, assuming you wish to maximize battery life?

E) TPS63001, a 3.3V buck-boost regulator

(3) Part c) Which device do you use to create a +3.3V power source from a 7.4V battery, assuming you wish to minimize noise?

C) TPS78233, a 3.3V linear regulator

Consider the two regulators used in the Lab 7 starter file (LP2950-3.3 and LM2937-3.3). Let the input voltage be V_{in}, and let the output voltage be V_{out}. Also, let the input current be I_{in}, and let the output current be I_{out}. Consider the following possible relationships between input and output:

G) $V_{out} = V_{in}$
H) $I_{out} = I_{in}$
I) $V_{out} * I_{out} = V_{in} * I_{in}$
J) $V_{out} / I_{out} = V_{in} / I_{in}$
K) $V_{out} * I_{in} = V_{in} * I_{out}$
L) V_{out} = 3.3V for all possible values of V_{in}

(2) Part d) Which relationship best describes the LP2950-3.3?

H) $I_{out} = I_{in}$

(2) Part e) Which relationship best describes the LM2937-3.3?

H) $I_{out} = I_{in}$

Again, consider the two regulators used in the Lab 7 starter file (LP2950-3.3 and LM2937-3.3). Let the dropout voltage be V_{do}. Consider how dropout affects the relationship between input and output voltage:

$V_{out} < V_{in} - V_{do}$

These are the possible symbols for ??? and ¿¿¿:

M) = P) < R) +
N) > Q) * S) -

(2) Part f) Which symbol should you insert for ???

P) <

(3) Part g) Which symbol should you insert for ¿¿¿

S) -

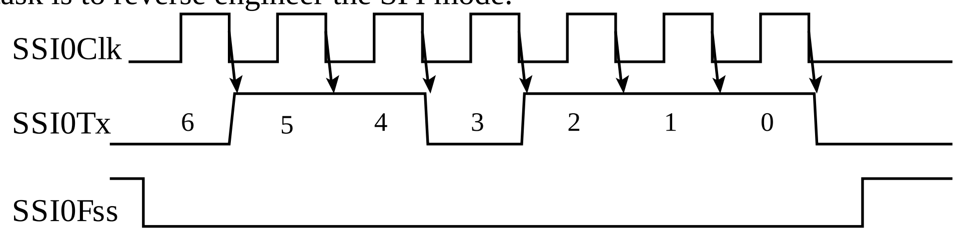
(16) Question 2. Three high-speed digital inputs are connected to PA2, PA3, and PA4. The overall goal is to maintain three counters, **`Cnt2 Cnt3 Cnt4`**, containing the total number of times each digital input falls (1 to 0). You may assume Port A has been initialized so an edge-triggered interrupt is requested on the falling edge of any of these three inputs. The other five Port A pins are not used. You may assume the signals do not bounce; i.e., you will count all the falling edges. Your system must handle simultaneous or *near-simultaneous* falling edges. I.e., the time t0 may be any value.
You may assume the minimum period (t1) of any one input is larger than 20 µs. However, the minimum pulse width (t2) may be as small as 100 ns.
The trick to handling 100ns pulse width is to look at RIS and not DATA
The trick to removing the race is to acknowledge only the bits that have been serviced

```
uint32_t Cnt2=0; // number of falling edges of PA2
uint32_t Cnt3=0; // number of falling edges of PA3
uint32_t Cnt4=0; // number of falling edges of PA4
```

Show the edge-triggered interrupt service routine. No time delays are allowed in the ISR. You do not need to show the main program or the initialization. The goal of the ISR is to update the above three globals

```
void GPIOPortA_Handler(void){
  if(GPIO_PORTA_RIS_R&0x10){  // could also look at MIS
    Cnt4++;
    GPIO_PORTA_ICR_R = 0x10;
  }
  if(GPIO_PORTA_RIS_R & 0x08){
    Cnt3++;
    GPIO_PORTA_ICR_R = 0x08;
  }
  if(GPIO_PORTA_RIS_R & 0x04){
    Cnt2++;
    GPIO_PORTA_ICR_R = 0x04;
  }
}
void GPIOPortA_Handler(void){ uint32_t flags;
  flags = GPIO_PORTA_RIS_R; // could also look at MIS
  if(flags & 0x10){
    Cnt4++;
  }
  if(flags & 0x08){
    Cnt3++;
  }
  if(flags & 0x04){
    Cnt2++;
  }
  GPIO_PORTA_ICR_R = flags; // ack the one that are serviced
}
```

(12) Question 3. An output device is interfaced to the microcontroller using SPI. The TM4C123 uses Freescale mode with the TM4C123 as master. The following response was measured with the logic analyzer. Your task is to reverse engineer the SPI mode.



(3) Part a) What value did the software write to **DSS** during initialization?
7 clocks mean 7 bits

DSS=6 (7 bits)

(3) Part b) What value did the software write to **SPO** during initialization?
Receive on rise, transmit on the fall

SPO=0

(3) Part c) What value did the software write to **SPH** during initialization?
Clock is low during idle means SPH=0

SPH=0

(3) Part d) What data value is being transmitted (in hexadecimal)?
MS bit first

0110111 = 0x37

(5) Question 4. The following C code is included in all the starter projects used for the ADC in this class

```
ADC0_PC_R = 0x01;       // 2) configure for 125K samples/sec
```

What does this one line of code establish? Answer A-F

A) Sets the actual sampling rate to f_s = 125 kHz.
B) Sets the resolution of the ADC to 12 bits
C) Specifies the speed of the ADC conversion, this is the slowest possible setting with the best SNR
D) Sets hardware averaging in order to improve SNR
E) Adds a low pass filter to reduce anti-aliasing errors
F) Sets the priority of the ADC interrupt

C (maximum f_s)

(+3 bonus) Question 5. We can configure the ADC to take multiple samples and perform hardware averaging. The Central Limit Theorem (CLT) states as the number of samples increase, the calculated average (your data) will approach the theoretical mean (true signal). The CLT also states that regardless of the original probability density function (pdf) of the noise, the pdf of the averaged signal will become Gaussian. List the assumptions required in order to apply the CLT in this data acquisition system.

Noise is random variable
Noise is independent from each other
Noise has zero mean

(15) Question 6. Design a set of software functions to implement fixed-point math. The resolution is specified with the following define statement, such that the resolution is **1/scale**. For example:

```
#define scale 1000
```

For this scale of 1000, the resolution of the fixed-point number would be 0.001. E.g., if the value is 1.234, the integer would be 1234. Your software function should continue to operate properly simply by changing the **#define** statement and recompiling. The number system is 32-bit signed. No floating point is allowed.

(5) Part a) Implement the addition operation, Extra credit for handling overflow.

```
int32_t fixed_add(int32_t x, int32_t y){
 int64_t r;
  r = (int64_t)x+(int64_t)y; // promote
  if(r > 2147483647) r = 2147483647;
  if(r < -2147483648) r = -2147483648;
  return (int32_t) r;
}
int32_t fixed_add(int32_t x, int32_t y){
 int32_t r=x+y;
  if((x>0)&&(y>0)&&(r<0)) r= 2147483647;
  if((x<0)&&(y<0)&&(r>0)) r= -2147483648;
  return r;  // demote
}
int32_t fixed_add(int32_t x, int32_t y){
  return x+y;
}
```

(5) Part b) Implement the multiplication operationExtra credit for handling overflow.

```
int32_t fixed_multiply(int32_t x, int32_t y){
int64_t r;
  r = ((int64_t)x*(int64_t)y)/SCALE; // promote
  if(r > 2147483647) r = 2147483647;
  if(r < -2147483648) r = -2147483648;
  return (int32_t) r;  // demote
}
// multiply does not flip sign on overflow
// e.g., 4million*4million=16trillion truncates to 1246822400,
//    which is still positive
int32_t fixed_multiply(int32_t x, int32_t y){
  return (x*y)/SCALE;
}
```

(5) Part c) Implement the division operation, divide by 0 errors.

```
int32_t fixed_divide(int32_t x, int32_t y){
  return (x*SCALE)/y;
}
```

(5) Question 7. What is the best definition for the **ADC sequencer**? Put your answer in the box.

A) When the ADC is operating, the sequencer controls the conversion process: first the sample/hold latches the analog input, and then the successive approximation technique is used to convert the bits from most significant to least significant bit conversion.
B) When the ADC samples multiple channels, the sequencer determines the list of channels and the order in which the ADC channels will be sampled.
C) The sequencer specifies which of many sources that could be used to trigger the ADC sequence (e.g., software start, timer triggered or input GPIO triggered).
D) When the ADC samples multiple channels the sequencer specifies whether it samples from low to high, or from high to low. E.g., sample channels 0, 1, 2, 3 or 3, 2, 1, 0.
E) The sequence specifies how fast the ADC sequence will take.

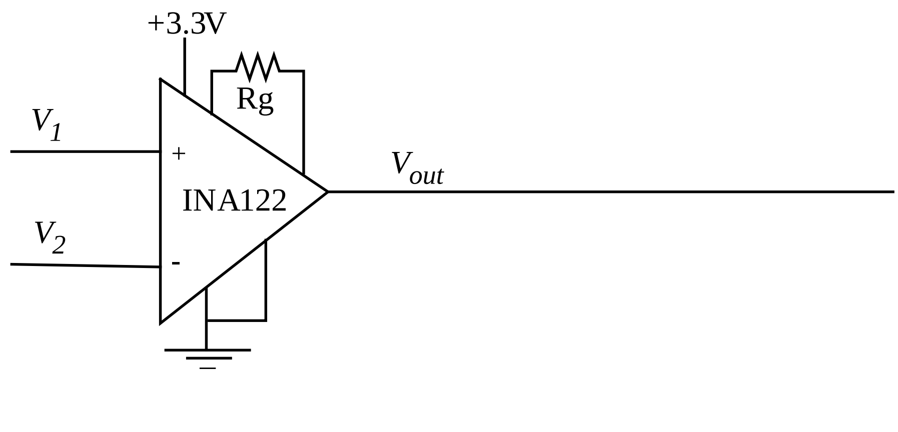
B

(5) Question 8. A microcontroller uses a 6-bit, 0 to 2V ADC. The microcontroller samples this signal at 2 kHz. The bus clock is 100 MHz. What is the resolution of this ADC in dB_{FS}? Show your work

Full scale signal is 2V
Resolution is 2V/64 = 0.01325V
Resolution/Fullscale =2V/64 /2V= 1/64
$dB_{FS} = 20 \log_{10}(1/64) = -36.1$ dB

(10) Question 9. Design an analog circuit with the following transfer function: $V_{out} = 333*(V_1-V_2)$. The input (V_1-V_2) ranges between 0 to 0.01V, so the output range will be 0 to 3.3V. Design the circuit with large input impedance and good CMRR. You may use any chips shown in the book or presented in class. Show your work and label all chip numbers and resistor values. You do not have to show pin numbers. You do not need to add a low pass filter.

Must use an instrumentation amp because of differential input, large Z_{in}, good CMRR
Rg = 200k/(333-5) = 610Ω



(14) Question 10. Your job is to interface this transducer to the ADC on the TM4C123. The transducer output is a single voltage called V_{in}, such that V_{in} varies from -0.01 to +0.01V. Your goal is to make the output voltage vary from 1 to 2 V. The output, V_{out}, is connected to the microcontroller ADC. R1 and R2 are already chosen with a LM4041 circuit such that the analog reference is 1.5V. You may use any chips shown in the book or presented in class. Show your work and label all chip numbers and resistor values. You do not have to show pin numbers. You do not need to add a low pass filter. Use E24 values

10	11	12	13	15	16	18	20	22	24	27	30
33	36	39	43	47	51	56	62	68	75	82	91

Table 9.2. E24 Standard resistor and capacitor values for 5% tolerance.

Gain = (2 – 1)/(0.01 – -0.01) =1/0.02 = 50
Input of 0 goes to Output of 1.5, so offset is 1.5V
V_{out} = 50 V_{in} +1.5
V_{ref} = 1.5V (given), V_{out} = 50 V_{in} + V_{ref}
Let V_g =0, add ground gain to main sum of gains 0, V_{out} = 50 V_{in} -50 V_g + V_{ref}
Build, R_f = 100 kΩ (a common multiple of 1, 50 that has E24 values)
R_{in} = 2 kΩ (gain, 100 kΩ / R_{in} = 50) The ratio could have been 750k/15k
R_g = 2 kΩ (gain, 100 kΩ / R_g = 50)
R_{ref} = 100 kΩ (gain, 100 kΩ / R_{ref} = 1)
It is very bad to use an instrumentation amp for situations like this that do not have a differential input