Jonathan W. Valvano First:__________________ Last:____Solution________

November 20, 2015, 10:00-10:50am. Open book, open notes, calculator (no laptops, phones, devices with screens larger than a TI-89 calculator, devices with wireless communication). You have 50 minutes, so please allocate your time accordingly. ***Please read the entire quiz before starting***.

(10) Question 1. Design a circuit that takes a 3.7V Li-Ion battery and produces a 3.3V power supply for an embedded system. In sleep mode the system needs 0.05 mA of current, and in active mode the system requires 20 mA of current. Show all part numbers, but do not worry about exact values for the resistors, capacitors and inductors. It runs in sleep mode 99% of the time and runs in active mode 1% of the time.

See Table 9.5 in the book

Need output voltage 3.3V

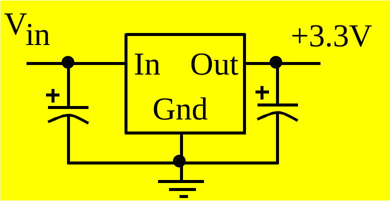
Need output current more than 20 mA

Need low dropout voltage, VDO<0.4V, so use any of these

LP2950-3.3

LP2981-3.3

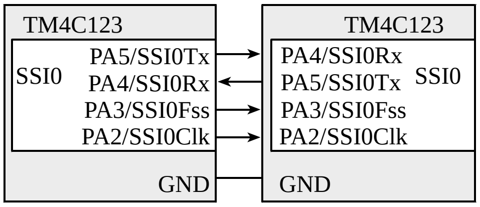
TPS78233



(5) Question 2. Consider a quantitative data acquisition system that measures voltage versus time. The input voltage range is 0 to 2.5 V. The ADC is 14 bits. The microcontroller crystal is 48 MHz. The bus clock is 48 MHz. The ADC is clocked such that the maximum sampling rate is 125 kHz. A periodic timer triggers an interrupt at 1 kHz (every 1 ms), and the timer ISR starts the ADC, waits for the ADC to complete, and stores the result into an array. The array can hold 256 samples. Hardware averaging is selected so each sample is the average of four ADC conversions. The array fills up in 256ms. Once the array is full the timer interrupt is disarmed, and a DFT (FFT) is performed on the data. What is the frequency resolution represented in these digital samples? Show your work.

Resolution is range over precision; $\Delta f = f_s/N$ = 1000Hz/256 = 3.9Hz

On the DFT the frequency range is -½ f_s to +½ f_s with N points



(15) Question 3. The goal is to transmit data from left to right using SSI. The microcontroller on the left is master and the one on the right is the slave. When the left microcontroller wishes to transmit 16 bits of data, it calls this function

```
void SSI_Out(uint16_t data){
  while((SSI0_SR_R&0x00000002)==0){};
  SSI0_DR_R = data; }
```

Part a) Assume the master (one on the left) has **SPH**=1, **SPO** =0, and **DSS**=15 in the **SSI0_CR0_R** register. What are the correct configurations for **SPH**, **SPO**, and **DSS** in the slave (one on the right)?

SPH = 1
(same as master)

SPO = 0

DSS= 15

Part b) The master will be calling **SSI_Out** inside a periodic ISR, so we have to assure the time to complete the ISR is short and bounded. Assume the SSI is idle and the master calls **SSI_Out** *N* times in order to transmit *N* halfwords to the slave. What is the maximum value that *N* can be so that the master is guaranteed that the busy-wait loop in **SSI_Out** never spins? Assume the software is faster (bus clock at 80 MHz) than the SSI hardware (SSI clock at 1 MHz). Justify your answer.

Hardware FIFO queue can hold 8 frames, so ISR can output 8 times without stalling

Part c) Which pins in the hardware interface can be disconnected and still have the interface function as described? If no pin can be removed, specify none. If multiple pins could be removed, list them all.

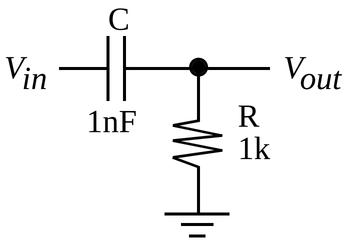
1) FSS is not needed
2) slaveTx-master Rx

(15) Question 4. This software increments the counter on the falling edge of PC4. Edit the following two functions so the counter is incremented on the **rising edge** of **PC7** instead of the falling edge of PC4. Change the priority to level 1. Cross out parts of the code you wish to delete and insert necessary additions.

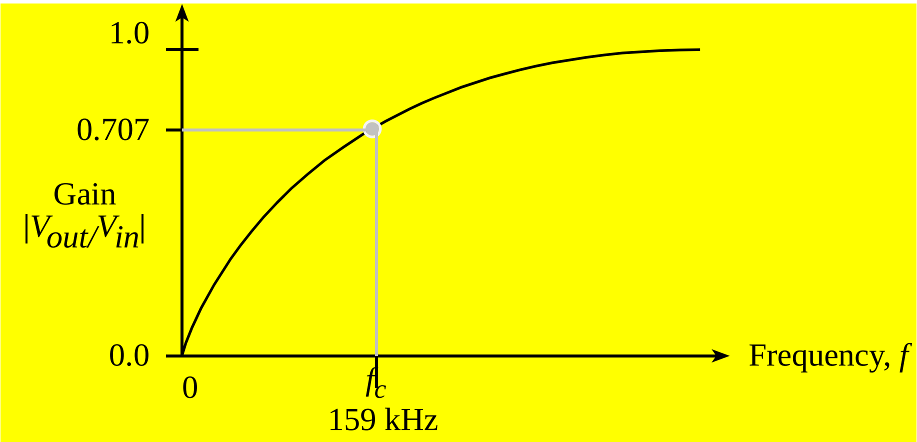
```
volatile uint32_t Counter = 0;

void EdgeCounter_Init(void){

  SYSCTL_RCGCGPIO_R |= 0x04;

  Counter = 0;

  GPIO_PORTC_DIR_R &= ~~~0x10~~; 0x80;

  GPIO_PORTC_DEN_R |= ~~0x10~~; 0x80;

  GPIO_PORTC_IS_R &= ~~~0x10~~; 0x80;

  GPIO_PORTC_IBE_R &= ~~~0x10~~; 0x80;

  GPIO_PORTC_IEV_R ~~&= ~0x10~~; |=0x80; // rising

  GPIO_PORTC_IM_R |= ~~0x10~~; 0x80;

  NVIC_PRI0_R=(NVIC_PRI0_R&0xFF00FFFF)|~~0x00A00000~~;0x00200000;

  NVIC_EN0_R = 4;

  EnableInterrupts();

}

void GPIOPortC_Handler(void){

  GPIO_PORTC_ICR_R = ~~0x10~~; 0x80;

  Counter = Counter + 1;

}
```

(15) Question 5. Consider the following circuit with a 1-nF capacitor and a 1-kΩ resistor.
(5) Part a) Derive a relationship for the gain (V_{out}/V_{in}) as a function of frequency, *f*. Show your work.

$Z_c = 1/(j2\pi fC)$. $Z_r = R$. Ohms Law $V_{out}/V_{in} = Z_r /(Z_r + Z_c)$
$V_{out}/V_{in} = R /(R + 1/(j2\pi fC)) = (j2\pi\, f\, R\, C)/(1+ j2\pi\, f\, R\, C)$
$|V_{out}/V_{in}| = (2\pi\, f\, R\, C)/\mathrm{sqrt}(1+ (2\pi\, f\, R\, C)^2) = (f/f_c)/\mathrm{sqrt}(1+ (f/f_c)^2)$
where $f_c = 1/(2\pi\, R\, C) = 159$ kHz



(5) Part b) Draw a rough plot of the gain versus frequency for the circuit. Include three points, *f*=0, *f* at the point where is gain is sqrt(2)/2=0.707 and at very large *f*.



(5) Part c) Assume V_{in} is limited between 0 and 1 V. Can we connect V_{out} to the ADC input of the TM4C123 microcontroller? If yes, justify your answer. If no, explain why not.
No, the HPF will shift the signal to -½ to +½ V

(10) Question 6. Write the C code to implement the following equation using fixed-point math

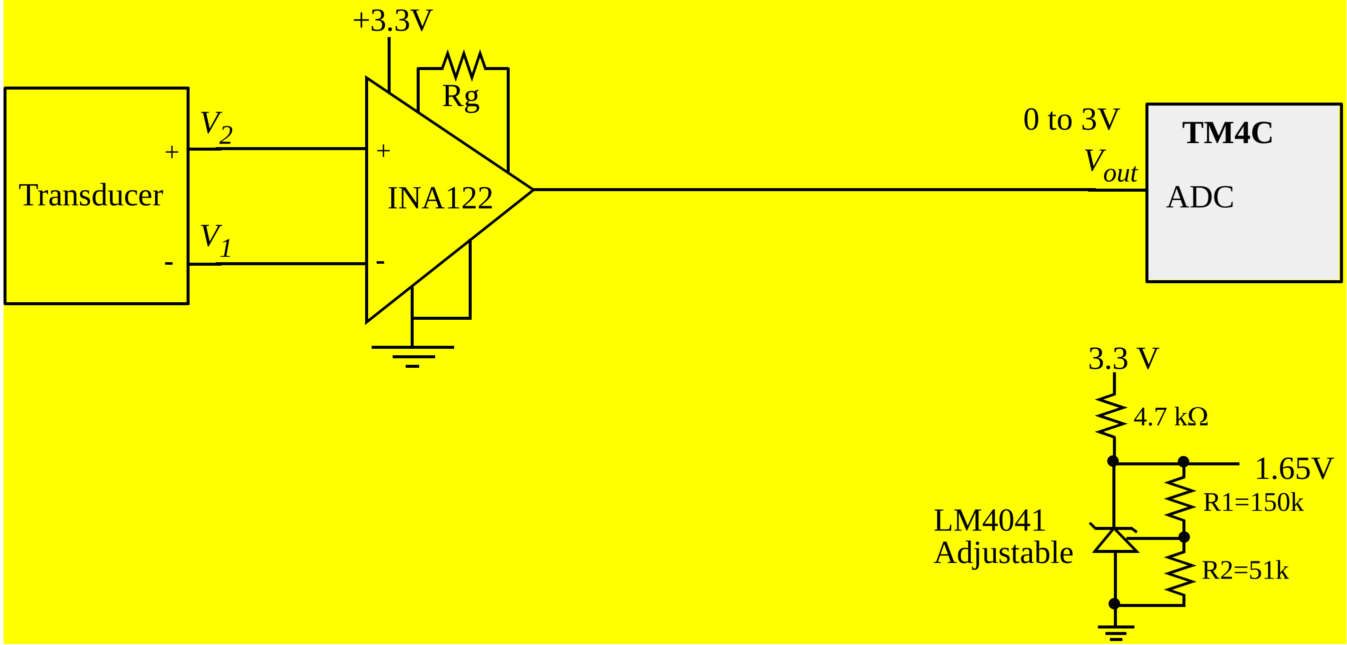
Y = sqrt(0.3)*X

Floating point is not allowed. The sqrt(0.3) ≈ 0.5477225575051661134569697828008. X and Y are 16-bit signed integers stored in global variables. Make it as accurate as possible without causing overflow during an intermediate calculation.

```
// choose 54772 so intermediate calculation does not overflow
// -32768*54722 and +32767*54722 both fit into a 32-bit register
// reduces by a factor of 0.54, so the result will fit into 16-bit Y
  Y = (54772*X)/100000;
```

(15) Question 7. Design an analog circuit that implements V_{out} = 500*V_2 – 500*V_1. The information in the signal is encoded as the difference between V_2 and V_1. The output, V_{out}, is connected to the microcontroller ADC. You may assume the inputs are bounded such that V_{out} will be between 0 and 3V. R1 and R2 are already chosen such that the analog reference is 1.65V. 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.

Gain is 500 = 5+200k/*Rg*, so *Rg* = 200k/495 = 404Ω


+3.3V
Rg
Transducer
V_2
+
V_1
-
+
INA122
-
0 to 3V
V_{out}
TM4C
ADC
3.3 V
4.7 kΩ
1.65V
R1=150k
LM4041
Adjustable
R2=51k