Jonathan W. Valvano

First Name: _______________ Last Name:____________________

April 1, 2011, 11:00 to 11:50am

Open book, open notes, calculator (no laptops, phones, devices with screens larger than a TI-89 calculator, devices with wireless communication). Please don't turn in any extra sheets.

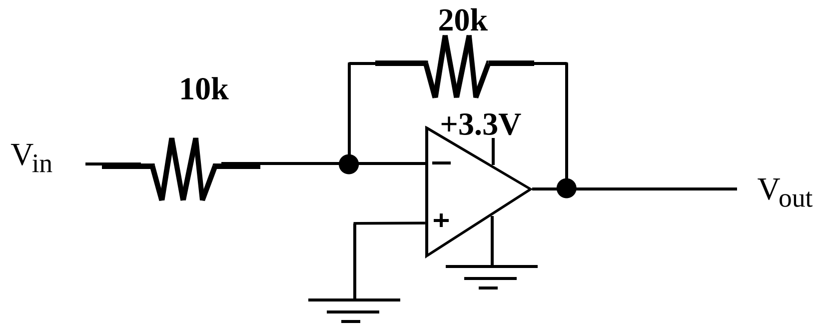
(15) Question 1. The goal is to study the frequency components of a signal using the FFT. The frequencies of interest are 10 kHz to 100 kHz. The desired frequency resolution is 100 Hz. The SNR of the signal is 80 dB. The slowest possible sampling rate is obviously 200 kHz because of the Nyquist Theorem. You may assume the sampling rate is fixed at 200 kHz.

Part a) Why should you add windowing before executing the FFT?

Part b) What is the smallest FFT size would choose, assuming the FFT must have a size that is a power of 2? Why?

Part c) What is the smallest number of ADC bits you could you choose and still capture all the information in the signal? Why?

(10) Question 2. Consider this analog circuit built with a rail-to-rail op amp powered with a single +3.3 V supply. Make a rough sketch (a plot) of the output voltage (**V_{out}**) as a function of the input voltage (**V_{in}**). I am asking for the DC response, not the frequency response.



(10) Question 3. First, derive the **H(z)** transform for this filter. Then, use the transform to identify all poles and zeros. Draw the pole-zero plot in the z-plane of this digital low-pass filter.

$$y(n) = (x(n) + x(n-1))/2$$

(30) Question 4. Write software to implement this moving averaging low-pass digital filter.

$$y(n) = (x(n) + x(n-1) + x(n-2) + x(n-3) + x(n-4) + ... + x(n-63))/64$$

The prototype for the function is

```
unsigned long Filter(unsigned long data);
```

where the input parameter is the 10-bit ADC sample, and the output parameter is the filter output. An example usage of this filter is

```
void Producer(void){ unsigned long data,result;
  data = ADC_In(1);       // sample channel 1
  result = Filter(data);  // call your function
  oLED_Plot(result);      // display filter output
}
```

Assume each addition, each memory read, and each memory store takes one bus cycle. Logical shift operations require no time because of the barrel shifter. We are also not counting bus cycles needed to fetch instructions, because fetching op codes often runs in parallel with data operations. If you were to implement a MACQ by shifting the data, it would take 63 reads and 64 stores to enter new data into the MACQ. The above filter itself has 64 reads, 1 store, and 63 additions, which adds up to a total of about 255 operations. The goal is to improve execution speed by reducing memory data bus cycles and by reducing arithmetic operations.

Part a) Prove that the following implementation is the same as the above filter.

$$y(n) = (64*y(n-1) + x(n) - x(n-64))/64$$

Notice to implement this filter we need only 3 reads, 1 store, and 2 additions.

Part b) Show the C function that implements this faster filter using unsigned integer math. Use a data structure that requires a lot less than 63 reads and 64 stores to enter new data. If you need initialization, define a second function that initializes the data structure.

Part c) For your solution in part b), count the number of memory bus cycles (reads and stores) required to execute the entire filter (entering data and performing calculations).

(20) Question 5. An electronic disk is made with flash EEPROM. Assume each block has a finite number of erase/program cycles before it begins to fail. For example, after 100,000 erase/program operations, individual block failures may occur. You may assume disk reads do not affect long-term failure rates. The goal is **not** to design the file system that is robust in the face of failure. The answer is **not** to employ redundancy using a system like a disk mirror. We define *life span* as the time the disk runs until its first failure. The goal of this problem is to maximize life span. *Wear-leveling* is a process that attempts to erase/program all the blocks at about the same rate.

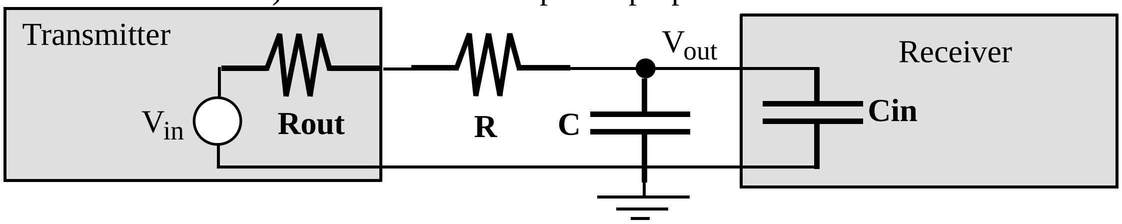
Part a) How would you design the **free space management** to improve the life span of the disk? It is not about allocation (contiguous, indexed, linked). It is about managing free space.

Part b) Assume the directory fits in one block. How would you manage the **directory** to improve the life span of the disk?

(15) Question 6. Consider a file system that uses indexed allocation. The disk size is 1 gibibyte (2^{30} bytes). A *cluster* is a group of blocks, such that one allocates, uses and deallocates individual clusters. This is a way to increase "block size" without changing the physics of how the disk works. This disk is clustered, such that the cluster size is larger than the physical block size of 512 bytes. The cluster size must be a power of 2. Each file has one index table, and the index must entirely fit in one cluster. Assume all cluster addresses are 32-bit numbers. Choose a cluster size that creates no external fragmentation and minimizes internal fragmentation.

Bonus question, skipped

(10) Question 2. Consider the following circuit model for the impedance in a cable (two long wires from transmitter to receiver). **R**=10Ω and **C**=10 pF are properties of the cable.



Part a) Assuming both **Rout** and **Cin** are zero, consider the response to a falling edge at $\mathbf{V_{in}}$. At time less than zero $\mathbf{V_{in}}$ is 3.3V, and for times greater than zero $\mathbf{V_{in}}$ equals 0V, what will be the transient output? Give both an explicit equation $\mathbf{V_{out}(t)}$ and a rough sketch of $\mathbf{V_{out}}$ versus time.

Part b) Let the output impedance of the transmitter **Rout** be 0.1Ω. In order to improve the frequency response (bandwidth) of this digital signal which of the following should I do?

A) Spend money and decrease **Rout** to half, **Rout** = 0.05 Ω.

B) Leave it as it is at 0.1 Ω.

C) Spend money and increase **Rout** to double, **Rout** = 0.2 Ω.

Part c) Let the input capacitance of the receiver is **Cin** be 100 pF. In order to improve the frequency response (bandwidth) of this digital signal which of the following should I do?

A) Spend money and decrease **Cin** to half, **Cin** = 50 pF.

B) Leave it as it is at 100 pF.

C) Spend money and increase **Cin** to double, **Cin** = 200 pF.