EECS 452 *Midterm* <u>Closed book part</u>

Winter 2013

Name: ____________________________________ unique name: _______________

Sign the honor code:

I have neither given nor received aid on this exam nor observed anyone else doing so.

____

Scores:

#	Points
Closed book	
Page 2	/17
Page 3	/10
Open book	**/73**
Total	**/100**

NOTES:

- There are **<u>3</u>** pages including this one.
- On the *closed book* section you may **not** have books, notes, or calculator. Nothing but a writing utensil.
- Don't spend too much time on any one problem.
- You have about 120 minutes for the exam. You probably shouldn't spend much more than 20 minutes on this section.

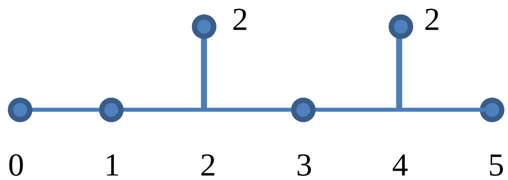
Closed book portion

1. Say you took a 16-point DFT of the following signals with f_s=3.2 KHz starting at time t=0. What would the non-zero DFT outputs be (magnitude and phase)? **[3 points each]**

   a. $2*\sin(2\pi*400t+\pi/2)$
      $X(2)=16\angle 0°$; $X(14)=16\angle 0°$

   b. $2*\cos(2\pi*800t) - 2$
      $X(0)=32\angle \pi°$; $X(4)=16\angle 0°$; $X(12)=16\angle 0°$;

2. An FIR filter has the impulse response shown below. Assume f_s=1ms. **[2 points each]**



   a. What is the difference equation for this filter?
      $y(n)=2*x(n-2)+2*x(n-4)$

   b. What is the transfer function for this filter?
      $H(z)=2*z^{-2} + 2*z^{-4}$

   c. Is this filter linear? If so, what is the group delay? If not, why not?
      Yes. 3ms.

3. Write a legal C function named "mult16" which takes two 16-bit Q15 numbers as arguments and returns their product as a 16-bit Q15 rounding appropriately. Use "short" as a 16-bit signed data type. **[5 points]**

```
short mult16(short a, short b)
{
      int tmp=(int)a*b;    //NOTE: cast isn't needed here as
                           //as multiplication defaults to an int.
      return((short)(tmp+0x4000)>>15));  //cast also not needed
}
```

4. *Fill in the blank or **circle** the **best** answer*. Provide all numbers in decimal.
   [10 points, -1 per wrong or blank answer, minimum 0]

   a. If we multiply an 8-bit *Q*6 value by an 8-bit *Q*6 value the result would be

      a __***16***______-bit Q___***12***_____ number.

   b. ___***-4***___ is the smallest number (closest to negative infinity) that can be exactly represented by a signed 11-bit Q8 number. __***1.875***__ is the largest number (closest to positive infinity) that can be exactly represented by a signed 5-bit Q3 number.

   c. In signed 5-bit two's complement Q3, what is the value of 10010? __***-1.75***__.

   d. We say a filter has *linear phase* if the group delay is ***independent of the input frequency / proportional to the input frequency / greater than the input frequency***.

   e. Say you have an ideal A to D converter which converts values from 4V to -4V to a

      5-bit value. In the ideal case, the worst-case quantization error is ±__***.125***__ Volts.

   f. One advantage of IIR filters compared to FIR filters is that IIR filters tend to ***use fewer total components to implement the same functionality / be easier to make linear phase / be easier to make stable***.

   g. FIR filters ***can easily / cannot / can with great difficulty*** implement a typical RC analog filter.

   h. Most computers don't have a MAC instruction because
      - ***it would likely cause the clock period to get longer.***
      - ***they don't have an accumulator.***
      - ***it interferes with floating point support.***

   i. DSPs, on the other hand, do tend to have a MAC instruction because
      - ***they generally do have an accumulator.***
      - ***they commonly can take advantage of MAC instructions.***
      - ***they don't generally have floating point instructions.***
      - ***they have a very slow clock period to begin with.***

   j. An N-point implementation of the FFT algorithm discussed in class requires approximately ***N^2 / $N^{1.5}$ / N*log(N) / N*** operations.

   k. A waveform is sampled using a sample rate of 42 KHz. If you desire a frequency

      spacing of 2 Hz in the DFT of the sample set, you'll need to sample for _***500***____ ms.

EECS 452 *Midterm* <u>Open book part</u>
Winter 2013

Name: ________________________________ unique name: _____________

Sign the honor code:

I have neither given nor received aid on this exam nor observed anyone else doing so.

____

Scores:

#	Points
Open book	
Page 2	/11
Page 3	/11
Page 4	/12
Page 5	/15
Page 6	/11
Page 7	/13
Total	**/73**

NOTES:

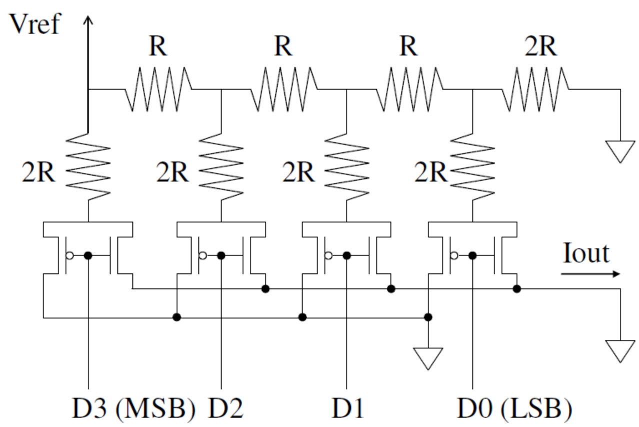
- There are <u>**7**</u> pages including this one.
- On the *open book* section you may use calculators, books and notes, but no PDAs, Portables, Cell phones, etc.
- Don't spend too much time on any one problem.
- You have about 120 minutes for the entire exam.
- Some questions are much harder than others…
- <u>***When asked to show your work do so.***</u> <u>***If***</u> *the question asks you to show your work and you don't you will get no points, even if the answer is correct.*
- If you need to use the back of the exam paper for your answers, you must make it clear that the grader is to look there.

Short answer:

1. Say you took a 32-point DFT of the following signals with f_s=1.6 KHz starting at time t=0. What would be what the non-zero DFT outputs would be (magnitude and phase)? Briefly justify your answers.
   [6 points, 3 points each]
   - 2*sin(2 π *1000t)
     freq spacing=50Hz.
     n=20, so X(20)=32∠-90°; X(n-20) is X(12)=32∠90°;
   - 2*cos(2 π *1600t- π/8)

   This one is probably harder. Can do it the same as above. In that case, X(n)=X(32-n)=X(0). So really taking the two vectors and adding them. Could do that.

   Another way to tackle this is to note we are sampling 2*cos(π/8) 32 times. So 32*2*cos(π/8) = 59.13∠0.
2. Answer the following questions about the ideal digital-to-analog converter shown below.
   [5 points]



   a. Assuming Vref=12V and R=1kΩ, if D[3:0]=1001, what is Iout? **[3]**
      I_{in}=12mA. 9*12mA/16=6.75mA

   b. If we were going to make a 12-bit converter using this R/2R ladder scheme, how many resistors would we need? **[2]**
      24

3. Say you are building an FIR bandpass filter that will have a passband of 1KHz centered at 8KHz. Further, say that there is a fair bit of signal in the input up to 40KHz. [**8 points]**
    - Without an anti-aliasing filter, what is the lowest frequency you could sample the data? **[3]**

      48.5KHz.

    - What would happen if you sampled at a lower frequency than that? **[2]**

      Some of that signal would end up in the frequency we wanted to sample. Draw the picture, it's pretty easy to see.

    - If you do have a (low-pass) anti-aliasing filter which has a passband that ends at 10KHz and a stopband that starts at 20KHz, what is the lowest frequency you could sample the data? You are to assume the gain of the anti-aliasing filter is 1.0 in the passband, 0 in the stopband, and unknown otherwise. **[3]**

      28.5KHz

4. Draw a block diagram which corresponds to the following difference equation. **[3 points]**

$$y[n]= x[n-1] +x[n-3] – y[n-2]$$

No answer provided

5. Consider the following difference equation **[6 points]**

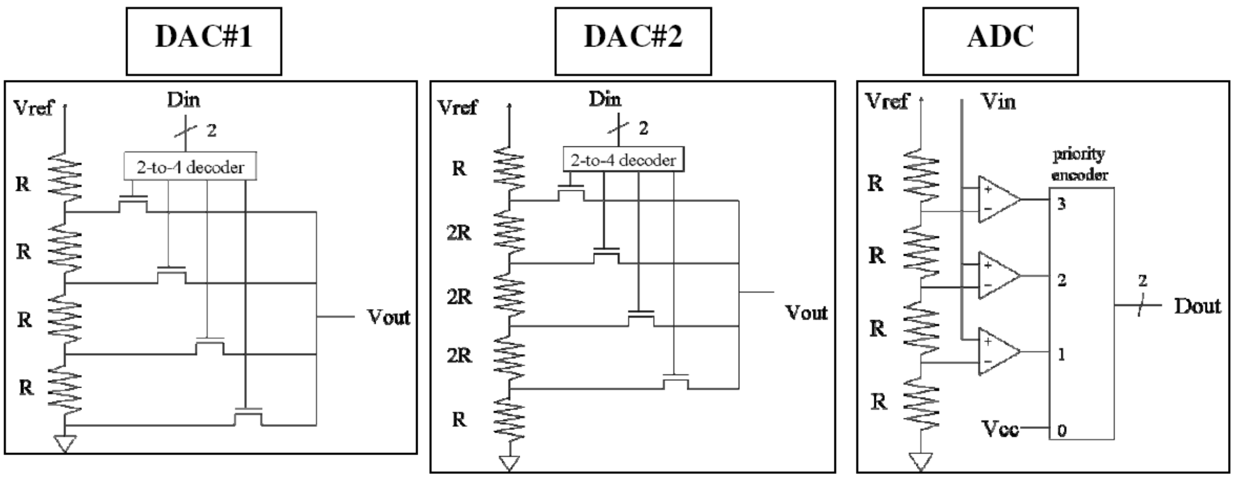
$$y[n] = x[n] + 0.2 * x[n-2] + 0.4 * y[n-2]$$

- Write the transfer function which corresponds to that difference equation. **[3]**

$$H(z) = \frac{1 + 0.2z^{-2}}{1 -.0.4z^{-2}}$$

- Is that filter stable? Show your work. **[3]**

Yes, it's stable if the poles are in the unit circle. z^2+0.4 has roots of about +/-0 .63, so they are in the unit circle.



6. Answer the following question using the above figures. Assume Vref=4V and that all converters are ideal. **[6 points]**

   a. If 2.4V is put into the ADC and D_{out} is connected to D_{in} of of DAC#1, what is the value found on V_{out} of DAC#1? **[2]**
   2.0V

   b. If 2.4V is put into the ADC and D_{out} is connected to D_{in} of of DAC#2, what is the value found on V_{out} of DAC#2? **[2]**
   2.5V

   c. If D_{in} of DAC#2 is "01" and the output of that DAC is connected to the V_{in} of the ADC, what is the output of the ADC? **[2]**
   01

Longer answer

7. Consider the following filter transfer function:

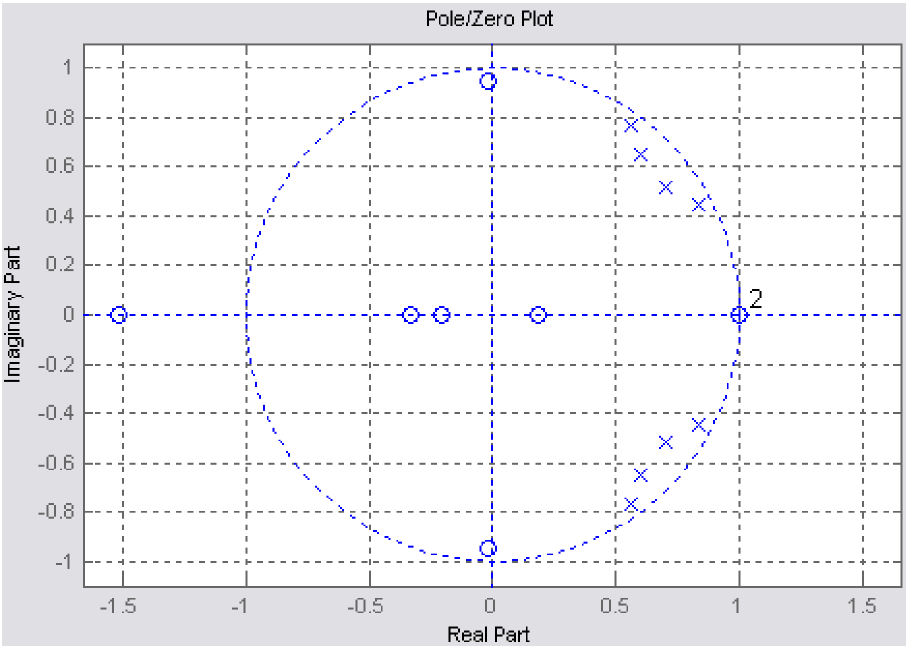
$$H(z) = 0.25 * z^{-1} - 0.5 * z^{-3}$$

Write a function, `int trans(int x, int *w)` which takes a single input x, and returns the output of the filter. You may assume that the array w is of any fixed size you find useful. All inputs are in 32-bit Q31 and the output is also to be in 32-bit Q31. If the output is determined to be outside of the range of a 32-bit Q31 you should saturate. You are to write this for a machine where an `int` is 32 bits and a `long` is 64 bits. This function will be called iteratively with a new value of x passed in each cycle but with a pointer always to the same w, so your code needs to update w appropriately in addition to returning the output value. **[15 points]**

```
int trans(int x, int *w)
{
    long y;
    w[3]=w[2];
    w[2]=w[1];
    w[1]=w[0];
    w[0]=x;
    y=(long)w[1]-2*w[3];
    return((y+0x2)>>2);
}
```

The above answer is taking advantage of a lot of characteristics of this FIR filter. For one, overflow is impossible as the largest output at any given time is 0 .75. For another, both coefficents are powers of two, making it really easy to write those terms as Q2 numbers. Then we need to round and divide/shift. Done. Most people had more complex answers and thus had more complex errors…

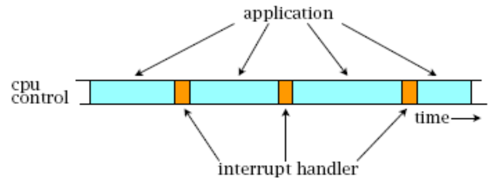
8. Consider the following pole/zero plot **[11 points]**



a) What type of filter (lowpass, highpass, bandpass, bandstop) is this? **[2]**
Bandpass.

b) Draw a rough sketch of the frequency response of this filter assuming f_s=10kHz. **[3]**

c) Circle the pole-zero combination that you would use for each biquad section to minimize the likelihood of overflow in your fixed-point implementation. Label each combination (1, 2, etc.) corresponding to the order you would cascade the biquad sections (the biquad section closest to the input being 1, the next 2, etc.) **[6]**
No answer provided. There are a lot of correct answers to this one given that a number of poles look equidistant to the unit circle.



9. FFTs in the real world: **[13 points]**
On this question we will explore an FFT implementation on a processor and explore the impact interrupts have on our ability to do DSP work. For this entire problem, assume the following:
   - An N-point implementation of the FFT requires N/2 log N complex arithmetic operations, where N must be a power of 2. Was clarified to be log base-2 at the exam.
   - Each complex arithmetic operation consumes 5 clock cycles on a CPU with a clock frequency of 100MHz.
   - We are sampling audio data at a rate of 50 kHz.
   - Our application requires us to not miss any of our samples.

a) Say we've written our program such that we grab X samples from our ADC as they come in and then after the last has arrived we need to start and complete our computation of the FFT over those X samples before the next sample arrives. (So we can do real-time spectrum analysis.) What is the largest FFT we'll be able to compute (that is, the largest value of X)? What will be the frequency resolution of that FFT? _Clearly_ show your work! **[5]**
(During exam it was clarified that log was intended to be log base 2!)
100MHz/50kHz=2000 cycles. 5 cycles/op=400 operations. N/2*lg(N)<400 is N=64 (64*6/2)=192. That gives us 50kHz/64=781Hz.

b) Now let us assume that:
   - We have a timer that interrupts our processor at a rate of 50 kHz and that that interrupt routine grabs a sample for us.
   - The interrupt routine takes 50 cycles to run.

i) If we want to do real-time spectrum analysis, what is the largest FFT would be able to compute? Don't worry if your answer seems unrealistic. _Clearly_ show your work. **[4]**

Get 390*N operations total if doing an N point FFT. Need N/2*lg(N) operations. That quickly gives us 390N ≥ N*lg(N)/2. 780 ≥ lg(N). The greatest N this holds for is 2^{780}. Not realistic (that would take longer to grab than the age of the universe by a massive amount!), but tells us that the sky is the limit.

ii) If all other assumptions remain the same, what is the slowest clock frequency the processor could have and still be able to do a 1024 point FFT in real-time? _Clearly_ show your work! **[4]**

Well, we need to be able to have 1024/2*10*5=25600 cycles over 1024 samples. Those samples take 1024*50=51200 cycles for the interrupt routine to run. So basically 75 cycles per sample. That comes out to 75*50kHz=3.750MHz.