ECE4703 Midterm Exam

Your Name: SOLUTION Your box #: _____

November 15, 2012

Tips:

- Look over all of the questions before starting.
- Budget your time to allow yourself enough time to work on each question.
- Write neatly and show your work!
- This exam is worth a total of 200 points.
- Attach your "cheat sheet" to the exam when you hand it in.

problem 1	problem 2	problem 3	problem 4	total midterm exam score
50 points	30 points	60 points	60 points	200 points

1. 50 points total. Consider the RT-DSP system shown in Figure 1. The analog input signal to the system is given as $x(t) = 1 + \cos(2\pi \cdot 1000t) + \cos(2\pi \cdot 10000t)$. The sampling rate is given as $f_s = 160000$ Hz.

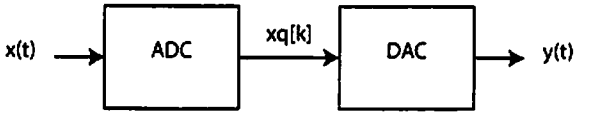
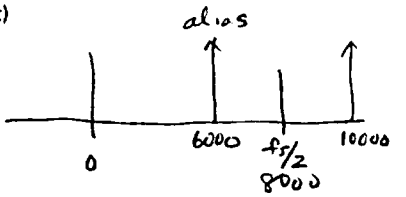


Figure 1: RT-DSP system.



(a) 25 points. Suppose the ADC and DAC are **ideal** such that $x_q[k] = x(kT_s)$ and $y(t)$ is generated by ideal sinc reconstruction of $x_q[k]$. There are no antialiasing filters in the system. Determine $y(t)$.

Since everything is ideal and we have no antialiasing filter, we get

$$x_q[k] = 1 + \cos\left(2\pi \cdot \frac{1000}{16000} k\right) + \cos\left(2\pi \frac{10000}{16000} k\right)$$

$$= 1 + \cos\left(\frac{2\pi}{16} k\right) + \underbrace{\cos\left(\frac{20\pi}{16} k\right)}_{\text{this term has frequency } > \pi \text{ hence there is aliasing}} = 1 + \cos\left(\frac{2\pi}{16} k\right) + \cos\left(\frac{12\pi}{16} k\right)$$

Ideal reconstruction $\Rightarrow y(t) = 1 + \cos\left(\frac{2\pi}{16} \cdot 16000\, t\right) + \cos\left(\frac{12\pi}{16} \cdot 16000\, t\right)$

$$\boxed{y(t) = 1 + \cos(2\pi \cdot 1000t) + \cos(2\pi \cdot 6000t)}$$

(b) 25 points. Now suppose you try to implement the system above with the TMS320C6713 DSK. The ADC and DAC functions are now performed by the AIC23 codec. What will different with respect to the ideal case in part (a)? Can you provide an approximate expression for $y(t)$ in this case?

The codec/DSK will block the DC term and the term above $f_s/2$ (the 10k Hz) term.

Hence, the only thing that gets through is the 1 kHz term.

The codec/DSK also delays this signal by τ, and cuts the amplitude by a factor of 2, even with no processing. Quantization error also has a small effect.

So $\boxed{y(t) \approx \frac{1}{2} \cos(2\pi \cdot 1000(t - \tau))}$

2. 30 points. The Texas Instruments TMS320C6457 is high performance fixed-point DSP with a maximum clock rate of 1.2 GHz. According to the TI documention, this DSP is "an excellent choice for applications including video and telecom infrastructure, imaging/medical, and wireless infrastructure". Suppose you use this processor to implement a filter in a wireless router which samples an incoming signal at a rate $f_s = 1$ MHz. What is the maximum number of cycles available to perform the filter processing?

$$\text{maximum cycles per sample} = \frac{\text{clock rate (cycles/sec)}}{\text{sample rate (samples/sec)}}$$

$$= \frac{1.2 \times 10^9}{1 \times 10^6} = 1.2 \times 10^3$$

Hence we would have at most **1200 cycles** to implement our filter

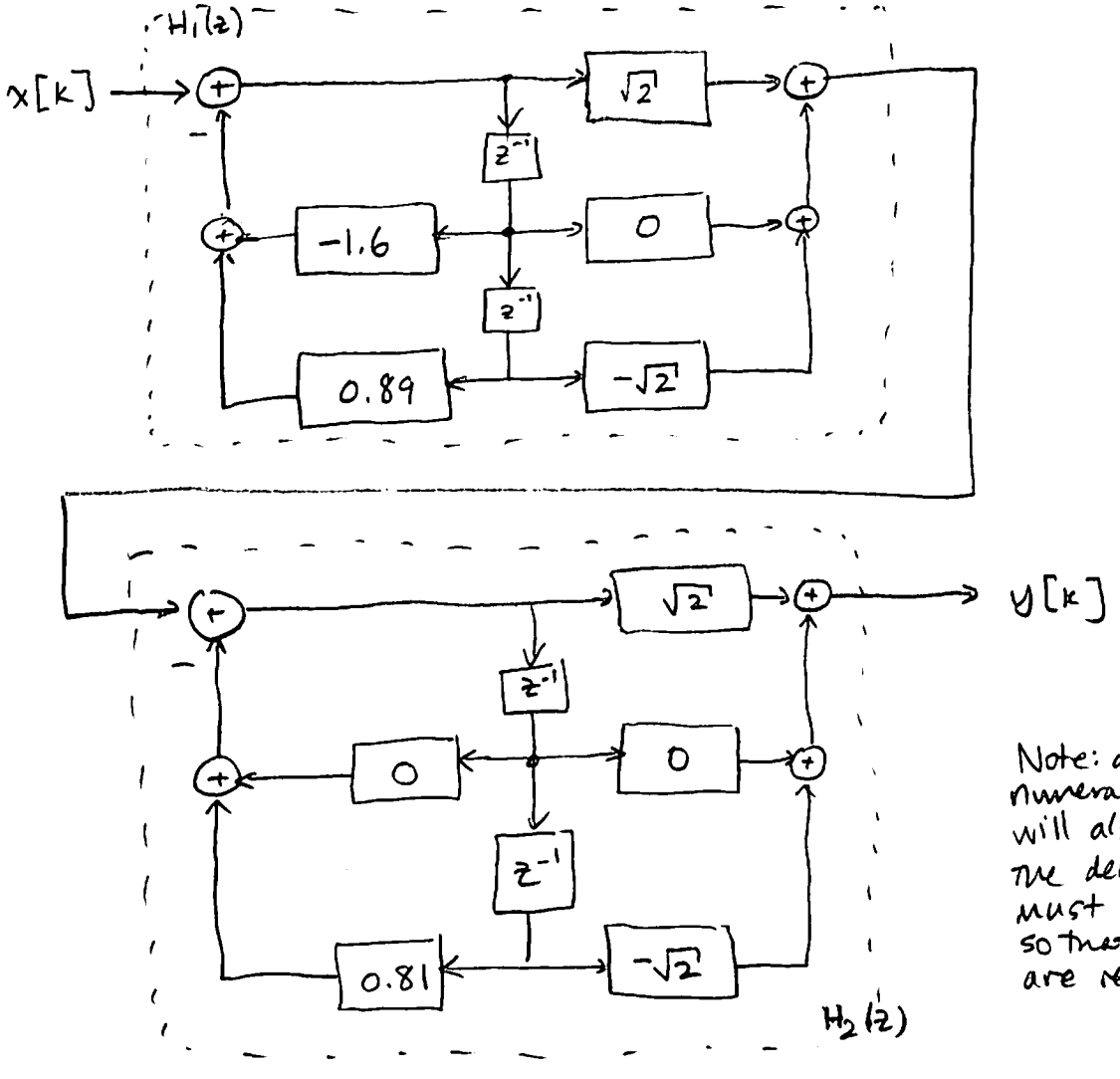
3. 60 points. Suppose you are given an IIR filter with the transfer function

$$\begin{aligned} H(z) &= \frac{Y(z)}{X(z)} \\ &= \frac{2(1-z^{-1})^2(1+z^{-1})^2}{(1-(0.8+0.5j)z^{-1})(1-(0.8-0.5j)z^{-1})(1-0.9jz^{-1})(1+0.9jz^{-1})} \\ &= \frac{2-4z^{-2}+2z^{-4}}{1-1.6z^{-1}+1.7z^{-2}-1.296z^{-3}+0.7209z^{-4}} \end{aligned}$$

with $j = \sqrt{-1}$. Draw a "Direct Form II - Second Order Sections" realization of $H(z)$ assuming infinite precision filter coefficients (each second order section should have *real-valued* coefficients). Draw neatly and label everything accurately for full credit.

$$H_1(z) = \frac{\sqrt{2}(1-z^{-1})(1+z^{-1})}{(1-(0.8+0.5j)z^{-1})(1-(0.8-0.5j)z^{-1})} = \frac{\sqrt{2}-\sqrt{2}z^{-2}}{1-1.6z^{-1}+0.89z^{-2}}$$

$$H_2(z) = \frac{\sqrt{2}(1-z^{-1})(1+z^{-1})}{(1-0.9jz^{-1})(1+0.9jz^{-1})} = \frac{\sqrt{2}-\sqrt{2}z^{-2}}{1+0.81z^{-2}}$$



Note: different numerator factorizations will also work but the denominators must be factored so that the coefficients are real.

4. 60 points. Suppose you have a C program that computes the dot product of two fixed-point arrays of 16-bit signed integers and stores the result as a fixed-point number in a 32-bit signed integer container. Your code looks like this:

```
#define N 50
#define s1 ?? 0
#define s2 ?? 0
#define s3 ?? 5
short a[N];  // Q-11
short b[N];  // Q-10
int q = 0;   // Q-?? Q-21
int r = 0;   // Q-?? Q-16
int n;
//
// some code in here sets all the values of a and b
//
for (n=0;n<N;n++)
  q = (a[n]>>s1)*(b[n]>>s2);    Q-21
  r += q>>s3;
```

(a) 30 points. Using **worst-case analysis**, determine the best choice for the shifts **s1**, **s2**, and **s3** so that the maximum precision is maintained for as long as possible while avoiding overflow. Explain your reasoning. What Q-format is the result **r**?

a[n] and b[n] are short. The product of two shorts can't overflow an int datatype, so $s1 = s2 = 0$.

The largest positive value q can take is 2^{30}.
If we sum 50 of those, we get $r_{max} = 5.3687 \times 10^{10}$.

This requires 36 bits + 1 sign bit to avoid overflow.
Hence $\boxed{s_3 = 5}$, based on worst-case analysis
The resulting sum will be $\boxed{Q\text{-}16}$

(b) 30 points. Suppose you run a test with some actual data (using your shifts **s1**, **s2**, and **s3** from part (a)) and determine the largest **q** value you ever see is 250×10^6. How does this affect your shift plan?

If we sum 50 of these, we get $r_{max} = 1.25 \times 10^{10}$
which requires 34 bits + 1 sign bit to avoid overflow

Hence, we could use $\boxed{s_3 = 3}$. Even though this doesn't satisfy the worst-case analysis, we will not have overflow since the q-values don't reach their maximum. In this case, the sum will be $\boxed{Q\text{-}18}$.