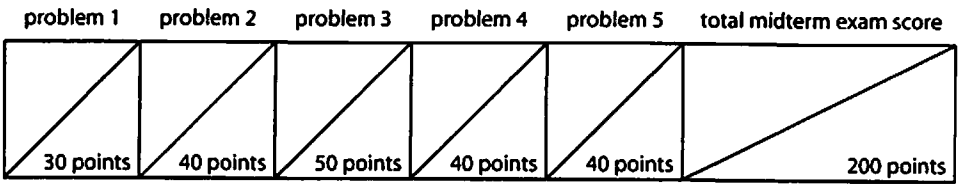
ECE4703 Midterm Exam

Your Name: SOLUTION Your box #: _____

November 18, 2010

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.



1. 30 points total.

   (a) 5 points. **True** / **False:** An advantage of floating point DSPs is that they are typically less expensive than fixed-point DSPs.

   (b) 5 points. **True** / **False:** An advantage of floating point DSPs is that they are typically faster than fixed-point DSPs.

   (c) 5 points. **True** / **False:** An advantage of floating point DSPs is that they typically use less power than fixed-point DSPs.

   (d) 5 points. **True** / **False**: An advantage of floating point DSPs is that they are typically more accurate than fixed-point DSPs.

   (e) 5 points. **True** / **False**: An advantage of floating point DSPs is that they are typically easier to program than fixed-point DSPs.

   (f) 5 points. **True** / **False**: Fixed-point DSPs can still do floating-point calculations, they just don't have a hardware floating-point ALU.

   (you can have floating point libraries - they just run very slowly)

2. 40 points total. You are given the following continuous-valued continuous-time (analog) signal

$$x_1(t) = a\cos(2\pi \cdot 1000t + \pi/10) \times \cos(2\pi f_0 t - \pi/3) + 1.$$

(a) 20 points. Suppose you convert this analog signal from continuous-time to discrete-time by ideally sampling at frequency $f_s = 8000$Hz and then convert this discrete-time signal back to continuous-time using an ideal reconstruction filter, i.e.,

$$x_1(t) \overset{\text{ideal sample at } f_s}{\longrightarrow} x[k] \overset{\text{ideal reconstruction}}{\longrightarrow} x_2(t).$$

Under what conditions on a and f_0 will $x_2(t)$ be equal to $x_1(t)$?

- If $a = 0$, then $x_1(t) = 1$ and it doesn't matter what f_0 is.

- If $a \neq 0$, then

$$x_1(t) = \frac{a}{2}\left(\cos\left(2\pi(1000+f_0)t + \frac{13\pi}{30}\right) + \cos\left(2\pi(1000-f_0)t - \frac{7\pi}{30}\right)\right) + 1$$

perfect reconstruction is possible if and only if

$$1000 + f_0 \leq \frac{f_s}{2} \iff \boxed{f_0 \leq 3\text{kHz}}$$

(b) 20 points. Suppose you set $f_0 = 1000$Hz and apply the signal $x_1(t)$ to the line-in input jack of the TMS320C6713 DSK sampling at 44.1kHz. Recall that the full-scale range of the DSK line-in input is approximately 5.9 volts peak-to-peak. What value should you set a to in order to maximize the SNR of the quantized signal? Explain.

$$x_1(t) = \frac{a}{2}\left[\cos\left(2\pi \cdot 2000t + \frac{13\pi}{30}\right) + \cos\left(\frac{-7\pi}{30}\right)\right] + 1$$

- $f_s = 44.1$kHz is more than enough to avoid aliasing

- the DC offsets in $x_1(t)$ is blocked by the analog circuitry in the DSK, so we can remove it from our analysis

So $x_1(t) = \underbrace{\frac{a}{2}\cos\left(2\pi \cdot 2000t + \frac{13\pi}{30}\right)}_{\text{The maximum value this can be is } \frac{a}{2} \leq \frac{5.9\text{ Vpp}}{2}} +$ blocked DC offset

Solve for a... $\boxed{a \approx 5.9}$

This maximizes SNR, but to be safe you should probably set a to about 90% of this value.

3. 50 points total. You are given the following infinite-precision FIR filter coefficients.

$$b = \begin{bmatrix} \sqrt{2} & -1/3 & \pi \end{bmatrix}$$

(a) 20 points. Suppose you are required to store these filter coefficients in a signed 4-bit fixed-point data type. Determine the optimum number of fractional bits to use in your fixed-point representation of the filter coefficients. Show your reasoning.

M	$\sqrt{2}$	$-1/3$	π	total squared error
0	1	0	3	pretty bad
1	1.5	-0.5	3	better
2	1.5	-0.25	1.75	← very bad due to saturation
no need to go further				

this is the best choice

M=1 fractional bit gives the best precision while avoiding overflow

(b) 30 points. Using your fractional bit specification from part (a), fill out the following table. Use the symbol Δ to show the binary decimal point and recall that negative binary numbers are represented in two's complement.

Coefficient Value	Quantized Value (decimal)	Quantized value (binary)	Quantization Error
$\sqrt{2}$	1.5	$001_\Delta 1$	0.086
$-1/3$	-0.5	$111_\Delta 1$ (twos comp)	0.1666...
π	3.0	$011_\Delta 0$	0.14159

4. 40 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 100
#define s1 ??
#define s2 ??
#define s3 ??
short a[N];  // Q-15
short b[N];  // Q-13
int r = 0;   // Q-??
int n;
//
// some code in here sets all the values of a and b
//
for (n=0;n<N;n++)
  r += ((a[n]>>s1)*(b[n]>>s2))>>s3;
```

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?

First, it should be immediately clear that $S_1 = 0$ and $S_2 = 0$ because the product of two 16 bit numbers can't overflow a 32-bit result. More formally,

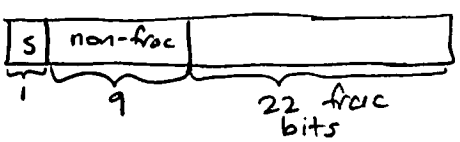
$$\frac{2^{N_r-1}-1}{2^{M_r}} \geq \left(\frac{-2^{15}}{2^{15}}\right)\left(\frac{-2^{15}}{2^{13}}\right) = 4$$

So, if $M_r = 28$ fractional bits, we can say that

$$2^{N_r-1} \geq 2^{30} \implies N_r \geq 31 \quad \text{(discarding the magnificant -1)}$$

so $\boxed{S_1 = 0}$ and $\boxed{S_2 = 0}$.

Now, summing up 100 of these results could lead to a sum as large as r=400. This would require 9 non-fractional bits to represent.



Hence $\boxed{S_3 = 6}$ to shift a Q-28 to a Q-22. The final result will be a $\boxed{\text{Q-22}}$.

5. 40 points total. Compare and contrast the "IIR Direct Form I" versus "IIR Direct Form II – Second Order Sections" realization structures in terms of:

(a) 10 points. **Ease of Programming**

This is, of course, subjective. Nevertheless, most would say that DF-I is conceptually simpler since it is just the direct implementation of the IIR input-output equation:

$$y[n] = \sum_{k=0}^{N-1} b[k]\, x[n-k] - \sum_{l=1}^{N-1} a[l]\, y[n-l]$$

(b) 10 points. **Memory Requirements**

DF-II has less memory requirements than DF-I because it only stores the intermediate results. SOS doesn't change that fact.

(c) 10 points. **Computational Requirements**

Essentially the same – DF-I might have a slight advantage, but this advantage is small for large order filters. Just count the # of multiply accumulates to see this.

(d) 10 points. **Robustness to Fixed-Point Coefficient Quantization Effects**

This is the primary advantage of DFII SOS. By factoring the transfer function into several cascaded SOS, you reduce the dyamic range of the coefficients and hence improve your ability to quantize them with less error.