ECE4703 Midterm Exam

Your Name: SOLUTION Your box #: __________

November 19, 2015

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! Points may be deducted for a disorderly presentation of your solution.
- 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	50 points	50 points	50 points	200 points

1. 50 points total. Suppose you implement a real-time signal processing system on the C6713 DSK with the interrupt service routine listed below (this should be similar to code you wrote for Laboratory assignment 1).

```
interrupt void serialPortRcvISR()
{
    union {Uint32 combo; short channel[2];} temp;
    float inL, inR, outL, outR;

    temp.combo = MCBSP_read(DSK6713_AIC23_DATAHANDLE);
    // right channel is in temp.channel[0] and left channel is in temp.channel[1]

    inL = (float) temp.channel[1];  // convert input to float
    inR = (float) temp.channel[0];  // convert input to float

    inL = inL*INSCALE; // scale channel to [-1,+1]
    inR = inR*INSCALE; // scale channel to [-1,+1]

    outL = inL;
    outR = inR*inR;  // square right channel

    // unscale and convert back to short
    temp.channel[1] = (short) (outL*OUTSCALE);
    temp.channel[0] = (short) (outR*OUTSCALE);

    MCBSP_write(DSK6713_AIC23_DATAHANDLE, temp.combo); // output L/R channels
}
```

(a) 10 points. What are the appropriate values for INSCALE and OUTSCALE?

Since both left and right channels are originally 16-bit short int with the dynamic range of [-32768 32767], to scale them to [-1, +1] we have to set $\boxed{INSCALE = \frac{1}{32768} = 3.052 \times 10^{-5}}$

to unscale and convert back to short we have to set

$\boxed{OUTSCALE = 32768}$

(b) 20 points. Suppose you have a DSK running the ISR shown on the previous page and that the AIC23 codec sampling frequency is set to 16 kHz. You apply a sinusoidal analog input signal the left and right line input channels of the DSK as $x(t) = a\cos(2\pi f t)$ with $a = 1$ volts and $f = 1$ kHz. What signal would we expect to see on the left and right outputs of the DSK?

On the left channel, since no processing occurs, the DSK only adds a delay and scales the amplitude by approximately $\frac{1}{2}$ So the left output is approximately

$$y_L(t) \approx \frac{a}{3}\cos\left(2\pi f\left(t-\tau\right)\right)\cdot(1.5) = \frac{1}{2}\cos\left(2\pi f\left(t-\tau\right)\right)$$

↑ full scale input (o-pk) ↑ full scale output (o-pk) ↑ delay

The right channel is squared. We know the DSK blocks the DC component resulting from the squaring. The double frequency component is below $\frac{f_s}{2}$, so it does not alias. Hence, the right output will be approximately

$$y_R(t) \approx \left(\frac{a}{3}\right)^2 \frac{\cos\left(4\pi f\left(t-\tau\right)\right)}{2}\cdot(1.5) = \frac{1}{12}\cos\left(4\pi f\left(t-\tau\right)\right)$$

↑ delay.

(c) 20 points. Suppose you have the same situation as part (b) except $f = 10$ kHz. What signal would we expect to see on the left and right outputs of the DSK?

In this case, since $f > \frac{f_s}{2}$, the DSK anti-aliasing filter blocks the input and we can expect the output to be close to zero.

2. 50 points total. Suppose you implement a floating point FIR filter on the C6713 DSK. Profiling the interrupt service routine shows that the number of cycles required is

$$\text{cycles} = 200 + 20N$$

where N is the number of coefficients in the FIR filter. Recall that the TMS320C6713 DSK is clocked at 225 MHz.

(a) 25 points. What is the maximum number of coefficients that your filter can have and still run in real-time if $f_s = 44100$?

$$N_{max} = \frac{\text{max(cycles)} - 200}{20}$$

$$\text{maximum Processing time} = \frac{1}{f_s} = \frac{1}{44100} = \boxed{22.68\ \mu s} \Rightarrow$$

$$\text{Max (cycles)} = (22.68 \times 10^{-6}\ s)(225 \times 10^{6}\ \tfrac{\text{cycle}}{s}) = \boxed{5103 \text{ cycles maximum}}$$

$$\Rightarrow N_{max} = \frac{5103 - 200}{20} = 245.15 \Rightarrow \boxed{N_{max} = 245}$$

(b) 25 points. What are some things you could do to get a higher-order filter running in real-time? Be specific and discuss any tradeoffs.

1- reducing the sampling rate but we should be careful not to go below Nyquist rate or we cause aliasing and loss of fidelity.

2- try the optimizing compiler

3- switch to fixed-Point arithmetic but you have to do more Complex calculations and you may loss precision.

4- try hand-optimizing critical parts of the code in .asm file.

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

$$b = \left[\frac{-1}{2}, \frac{1}{3}, \frac{1}{4}\right]$$

(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.

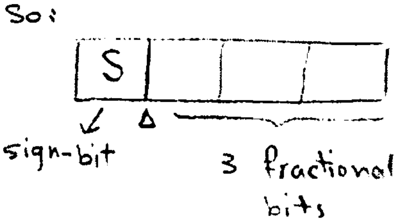
M=3

$-\frac{1}{2} = -0.5 \rightarrow$ is the largest coefficient requiring no non-fractional bit

$\frac{1}{3} = 0.333\overline{3}$

$\frac{1}{4} = 0.25$

So:



OR

M=4

4 fractional bits actually slightly better;

Largest positive value $= \frac{7}{2^4} \approx 0.4375$

Largest negative value $= \frac{-8}{2^4} = -0.5 \leftarrow$ exactly matches $\min(b)$

$\frac{1}{2} = 0.5 \Rightarrow$ best quantized value $0_\Delta 100$

for getting -0.5 we have to do two's complement

$$\begin{array}{r} 1011 \\ +\ 0001 \\ \hline 1100 \end{array}$$

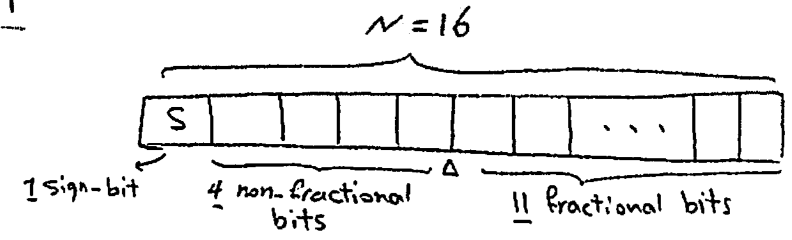
(b) 30 points. Using your fractional bit specification from part (a), fill out the following table. Use the symbol Δ to show the binary decimal point and recall that negative binary numbers are represented in two's complement.

Coefficient Value	Quantized Value (decimal)		Quantized value (binary)		Quantization Error
$\frac{-1}{2}$	M=3 - 0.5	M=4 -0.5	M=3 $1_\Delta 100$	M=4 $1_\Delta 000$	0
$\frac{1}{3}$	M=3 0.375	M=4 0.3125	M=3 $0_\Delta 011$	$0_\Delta 101$	$\approx -41.7\times10^{-3}$ ← M=3 $\approx 20.8\times10^{-3}$ ← M=4
$\frac{1}{4}$	M=3 0.25	M=4 0.25	M=3 $0_\Delta 010$	$0_\Delta 100$	0

4. 50 points total.

(a) 20 points. Suppose you implement a floating point IIR direct form II filter and test the filter with a white noise input. Further suppose the largest intermediate value observed in your tests was $|u[n]| \leq 10$. If we wish to implement this filter in fixed-point with 16-bit signed (short) datatypes for the intermediate values, what is the optimum number of fractional bits for the intermediate values?

for showing numbers up to 10 we need at least 4 non-fractional bits, so the optimum number of fractional bits are 11 or Q-11



(b) 15 points. What would be the effect of using *less* fractional bits in the intermediate values?

we will lose Precision in calculating u[n] and also it will cause loss of Precision in samples we read from input.

This may not have a drastic effect on the filter performance, but it will generally make the output noisier.

(c) 15 points. What would be the effect of using *more* fractional bits in the intermediate values?

with more fractional bits we have less than minimum required non-fractional bits so u[n] can not reach to it's maximum value.

This will probably lead to overflow in the intermediate values and poor filtering performance.