EECS 452 Practice Final Exam KEY
Winter 2009

Name: _____KEY_________________________    unique name: _____KEY_____

Sign the honor code:

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

______________________________

Scores:

<table>
<thead>
<tr>
<th>#</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Section I</td>
<td>/40</td>
</tr>
<tr>
<td>Section II</td>
<td>/30</td>
</tr>
<tr>
<td>Section III</td>
<td>/30</td>
</tr>
<tr>
<td>Total</td>
<td>/100</td>
</tr>
</tbody>
</table>

NOTES:

- Open book, open notes
- There are 8 pages including this one.
- Calculators are allowed, but no PDAs, Portables, Cell phones, etc.
- You have about 120 minutes for the exam.
- Be sure to show work and explain what you’ve done when asked to do so. Getting partial credit without showing work will be rare.
- In the context of the exam you are to assume all signed numbers are two’s complement numbers.
Section I -- Short answer/fill-in-the-blank -- 40 points

1) Convert the binary value 10001000 into base 10 assuming it is: [4]
   a) An unsigned binary number: \(128+8=136\)
   b) A 2’s complement number: \(-128+8=-120\)
   c) An unsigned Q7 number: \(1.0625\)
   d) A signed (two’s complement) Q7 number \(-0.9375\)

2) Consider a low-pass FIR filter with linear phase and a stopband at 100Hz.
   a) If a 1Hz sine wave entering the filter is delayed by 1ms, what is the phase you’d expect between the input and out of the filter for a 10Hz cosine wave? [3]
      \(-0.01\) periods \(-0.02\pi\) radians.
   b) If a 1Hz sine wave entering the filter has a delay of 1ms, what’s the delay you’d expect the filter to add to a 10Hz sine wave? [3]
      \(1\)ms

3) Say we perform the following operation using 4-bit signed integers. That is the result of each addition is stored as a 4-bit signed integer without saturation: \(1 + (-2) + 6 + 5 + 4\).
   a) If we add the numbers from left-to-right, what will be the result of the addition? [2]
      \(-2\)
   b) If we add the numbers from right-to-left, what will be the result of the addition? [2]
      \(-2\)
4) Say we perform the following operation using 4-bit signed integers. That is the result of each addition is stored as a 4-bit signed integer with saturation: $1 + (-2) + 6 + 5 + 4$.

a) If we add the numbers from left-to-right, what will be the result of the addition? [2]  \[ 7 \]

b) If we add the numbers from right-to-left, what will be the result of the addition? [2]  \[ 6 \]

5) A waveform is sampled using a sample rate of 10 kHz. How many samples are required in order to have a frequency spacing of 4 Hz in the DFT of the sample set? [3]

\[ 2500 \]

6) Say you have an ideal A to D converter which converts values from 16V to 0V to an 8-bit value.

a) What is the size of an LSB in this converter? [2]

\[ 2^4 \frac{V}{2^8} = 0.0625 V \]

b) What is the expected value for the magnitude of the quantization error in Volts? [3]

Max error is $\frac{1}{2}$LSB. Average is $\frac{1}{2}$ of that as it is linear. So 0.015625V

c) What is the expected value for the noise energy due to the quantization error? [2]

\[ (0.0625)^2/12 = 3.255E-4 \text{ Joules}. \]

7) Say you took a 16-point DFT of the following signals with $f_s=3200$Hz starting at time $t=0$. What would be what the non-zero DFT outputs would be (magnitude and phase)? [3 each]

a) $2 \cdot \sin(2 \pi \cdot 400t + \pi) + 1$

No answer provided.

b) $2 \cdot \cos(2 \pi \cdot 200t)$

No answer provided.
8) Say we sample an input waveform at 128KHz using an ideal 12-bit Analog to Digital converter. Assuming we only care about signals 8KHz and below in frequency, we could use this converter to have about the same SNR as a __14___ bit converter. [3]

9) Clearing Timer1 interrupt on the C5510 would involve writing a __1___ to bit __6___ of register __IFR1_______________. [3]
Section II – Longer answers -- 30 points

1) DFT vs. FFT
   a) Say we have an implementation of the radix-2 FFT which performs 48 operations to perform an FFT of 8 elements and 128 operations to perform an FFT of 16 elements. How many operations would you expect an FFT of 4 elements would take? Of 32 elements? Justify your answers. [5]

   We know that the radix-2 FFT should have a runtime proportional to \(N \cdot \log_2(N)\). For 8 elements we’d expect it to take \(8 \cdot 3 \cdot C\) (where \(C\) is an unknown constant) operations. So \(C\) seems to be 2. With 16 elements I get \(16 \cdot 4 \cdot C\) and again \(C\) seems to be 2. So for 4 elements I’d expect \(4 \cdot 2 \cdot 2\) or 16 operations and for 32 elements I’d expect \(32 \cdot 5 \cdot 2\) or 320 operations.

   b) Say we have a naïve implementation of the DFT which takes 32 operations to complete a 4-element DFT and 128 operations to complete an 8-element DFT. How many operations would you expect it would take to perform a 16-element DFT? Justify your answer. [4]

   Here we expect it to take a number of operations proportional to \(N^2\). So \(4 \cdot 4 \cdot C\) operations for a 4 element case (So \(C\) is again 2) and \(8 \cdot 8 \cdot C\) for the 8 element case (confirming \(C\) at 2 again). For a 16-element DFT we’re looking at \(16 \cdot 16 \cdot 2\) or 512 operations.

2) Many digital signal processors don’t have data caches, while most general microprocessor do have data caches. Why is that? [6]

   (Very long answer follows, just looking for the basics...)

   A cache is a fast (usually on-chip) memory that keeps a copy of the most recently accessed memory locations. It is worth noting that DSPs do have on-chip memory. In the case of the C5510 it is just part of the normal memory addressing scheme. The difference is that the programmer is responsible for choosing what data goes into it.

   The reason general microprocessors have caches is that the programmer doesn't generally want to have to deal with managing memory—she'd rather leave it for the hardware to figure out. When programming a DSP we generally are already worried about memory and are already thinking about what code really needs to run fast. So the extra work is the kind of thing we expect to do.

   Put another way, DSP programmers are the type to sit down and write assembly code to squeeze performance out of the processor, while your general programmers aren’t. Throw in the fact that the DSP programs tend to be a lot shorter and managing memory by hand seems reasonable.
3) Consider the following “front end” of a Delta Sigma converter. Assume that:
- After one clock the integrator’s output is equal to its prior value plus 0.5*current value.
- The threshold is 0.5V
- The impulse function either outputs 0V or 1V
Fill in the table showing the values of locations 1-4 in terms of volts and 5 in terms of a “1” or “0”.
You may leave off the units. [15]

*It isn’t clear what should happen at exactly 0.5V as an input to the threshold detector. So there’s two answers here (though both will be very similar)
1) JoeBlow Audio has hired you to build them a digital device which outputs a sine 300Hz sine wave which has an amplitude of 1 Volt (it ranges from -1 to 1 Volt). They want the sine wave you generate to never be more than 1mV different from an ideal (real number) sine wave with the same frequency and amplitude. You may assume you have a perfect frequency clock of whatever frequency you need and an ideal D to A converter.

After some thought, you’ve realized this seems fairly difficult because you will be introducing error due both to the time and amplitude quantization inherent in any DDS scheme. So you’ve decided not to let the time quantification cause more than a 0.5mV error and the amplitude quantification cause more than a 0.5mV error, thus insuring that there is no more than a 1mV error at any given point.

a) To achieve these goals, what is the minimum number of bits that must be in each entry of the sine table? Clearly show and explain your work. [7]

We know that quantization error is $\pm \frac{1}{2}$ LSB. So we need the LSB to be no more than 1mV. **11 bits** would give us $2V/2048$ for a step size, or just under 1mV for an LSB. This assumes our DtoA converter has a range of 1 to -1 V.

b) To achieve these goals, what is the minimum number of entries that must be in the sine table? Clearly show and explain your work. [8]

Now we are talking about time quantization. It’s pretty clear we’ll want exactly one period of a sine wave in the table. The key question is what is the smallest unit of time under which one would see a 0.5mV change? So we need to know how rapidly a sine wave changes. The derivative of a unit-amplitude sine wave is a cosine wave with the same frequency. So greatest rate-of-change will be when the sine wave is around zero. Back-of-the-envelope calculations tell us that as the derivative is 1 at that point, we’d expect to see a change of about 0.5mV in 0.5m periods. So our step size should be around $1/2000^{th}$ of a period. So $2^{11}$ entries looks good!
2) Consider a C function called biquad which has the following prototype:

```c
short biquad(short x, short a, short b, short c, short d, short e, int w[]);
```

You are to write C code which implements the filter shown below. The return value is to be “y”. You should assume that “short” values are 16-bit Q14 numbers and ints are 32-bit Q29 numbers. w[1] is w1 etc. and w[0] is unused. Further, you should change w[] so that w[] can be used for the next input value (much like you did in lab). Care should be taken in rounding and being sure to keep all values in the correct representation. You may assume that no overflow occurs. [15]

No answer provided.