EECS 452 Lab 2—Basic DSP Using theC5515 eZDSP Stick

1. Pre-lab

Q1. Print the file you generated (and modified) as described above.

Q2. The default structure of the FIR filter is "Direct-Form FIR"

a. How is this different from the Transposed-Form?
b. Draw the Direct-Form and Transposed-Form FIR. What do you think the advantage of each structure is?

Q3. Answer the following questions assuming you have the FIR filter you just created using Matlab.

a. What is the expected *group delay?*
b. What is the expected *phase shift* in a 1 KHz input?
c. What is the lowest frequency where you would expect to see a 180 degree phase shift?

Q4. In your own words, explain the role of each of the arguments to **fir()**.

Q5. Redraw the figure assuming index= 4.

	entry index = 2	lowest memory address
	x(j-nh+2)	
	x(j-nh+1)	
oldest x( ) entry ➔	x(j-nh)	
newest x( ) entry➔	x(j-0)	
	x(j-1)	
	x(j-2)	
	•	
	•	
	•	
	x(j-nh+5)	
	x(j-nh+4)	highest memory address
	x(j-nh+3)	

Q6. Let's suppose we have a buffer `in` to store audio samples, and this buffer has 256 entries. One way we could index the buffer is to use if statements and reset the index to 0 if it exceeds 255. How could we do this using bit-masking instead? Right one or two lines of code to accomplish this.

Q7. Read the in-lab section associated with DDS. Consider the figure found in that section with the caption "DDS block diagram". If the register were 16 bits but only the top 6 bits were used to index the sine table, what would be the output frequency of the generated sine wave if f_s were 48 KHz and FTV=2? Show your work.

Q8. If the sine table we are using doesn't have 64 entries and instead has only 48 entries, we can't use all the possible values of the register. *Given that*, what would be the output frequency of the generated sine wave if f_s were 48 KHz and FTV=2? Again, show your work.

Q9. Design a first-order passive low-pass filter. Assume you have only a 10k Ohm resistor and you want the "3dB-down" frequency to be 30 kHz. Assume you have any capacitor value you need.

a. Draw a picture of the circuit and label the values of the components.
b. If you were to use a 10 kHz unit-amplitude sine wave as an input, what would you expect the magnitude of the output to be?
c. As above but for a 30 kHz sine wave? A 100 kHz sine wave?

Q10. Consider a stereo tip-ring-sleeve (TRS) connector. Draw a diagram showing what each part of the connector is generally used for. Cite your source(s).

2. In Lab:

Q1. Input unit-amplitude 10, 30, and 100 KHz sine waves. For each input, list the output amplitude. How well does this match with your pre-lab answers?

Q2. If f_s=48 Khz and we are using the six most-significant digits of a 16-bit number to index our 64-entry table, what frequency sine wave would we expect to get if FTV=1? If FTV=1024? Explain your answers.

Q3. What impact, if any, does the filter have on smoothing out the signal?

Q4. While we've written the code for you, you will need to write similar code later. As such, let's examine a few different parts of this program and try to understand them.

a. Consider the FIR function. What is the array "in"?
b. In the FIR function, what role does "**sum = sum + 0x00004000;**" play in rounding?
c. In the while(1) loop, what role does "i" have?
d. What "pump" are we "priming" with the first for loop in the main? Explain what (if anything) would happen if we removed that code.

Q5. In the pre-lab, you were asked to predict the group delay. What group delay do you actually see? How did you go about measuring it?

Q6. What was the group delay? What was the lowest frequency at which the phase shift was 180 degrees?

Q7. Answer the following questions on filter timing.

a. If we have a 100MHz processor and we sample at about 50kHz (makes the math easier) what is the maximum number of cycles that our filter can use for each input before we need to worry about not being able to meet our real-time constraints?
b. How many cycles does our current filter take? To measure this, click the expressions tab, which is circled red in the following graphic and type into the text field delta_time. Launch the program then pause it to read delta_time. You can also toggle a break point at the beginning or end of a loop you're interested in so you're not randomly guessing where you've stopped the code.
c. Given the above, how large do you think we could make the filter before we started to run out of CPU time? (Hint: Try to half/double the length of the filter and see whether you half/double the CPU cycles, roughly. Then you have to consider the cycles taken by AIC_read2/AIC_write2.)

Q8. Now build and launch this code and measure the latency of filtering each sample. How does this compare to what we measured above? Why is that?

Q9. Now build and launch this code and measure the latency of filtering each sample. How does this compare with our base measurement?

Q10. Pick 3 of the assembly commands you see in the FIR filter function (such as MOV and SUB) and explain what they are doing. Also indicate how many cycles each of those instructions take. You will need to use the C5515 Mnemonic ISR reference (SWPU067E).

Q11. What are the advantages of using the if/else over the bit masking? What are the limitations of bit masking?

Q12. Using the fastest indexing protocol we have, about how many taps can we have in our filter?

Q13. What is the CPU latency per sample with your new filter?

Q14. Look at the libraries that are included in the project (right click on project in project window, go to "Properties". Then select "Build→ C5500 Linker"). Look under the "File Search Path". What library do you think is associated with TI's DSPlib?

Q15.

a. How many taps are there in your FIR filter?
b. How many cycles did the **fir()** function take to process a single input?

Q16.

a. How many taps are there in your FIR filter?
b. How many cycles did the **fir()** function take to process a single input?
c. Assuming there is some constant overhead associated with calling the fir() function and that runtime is otherwise linear, how many taps could you process in real-time? Show your work.
d. How does this compare to the code you wrote in the previous part?

Q17.

a. Sketch (roughly) the frequency response of your 500+ tap filter in the swept range.
b. How does your sketch compare to the frequency response that fdatool gave you when you designed the filter?

3. Post Lab:

Q1. One hardware construct that is commonly added to DSPs, but not to "standard" processors, is "circular buffers". See http://www.dspguide.com/ch28/2.htm for example.

a. Explain why they would be helpful here.
b. The C5515 supports circular buffering. Look at TI's manual named "spru371f.pdf" and read section 6.11. Explain, in your own words, how they work.

Q2. Consider the following FIR implementation and compare it to the one in fir_filter.c.

a. What's the advantage of using this one? (Hint: consider the case where you need to use multiple different FIR filters at the same time.)
b. This solution would still make it hard to do a low-pass 100-tap filter and a 50-tap high-pass filter. Why is that and how could you fix the problem?

```
Int16 FIR2(Int16* inBuf, const Int16* taps, Uint16 i)
{
        Int32 sum;
        Uint16 j, index;
        sum=0;

        //The actual filter work
        for(j=0; j<LPL; j++)
        {
                if(i>=j)
                        index = i - j;
                else
                        index = ASIZE + i - j;
                sum += (Int32)inBuf[index] * (Int32)taps[j];
        }
        sum = sum + 0x00004000;    // So we round rather than truncate.
        return (Int16) (sum >> 15);  // Conversion from 32 Q30 to 16 Q15.
}
```

Hand-in list:

Each group should hand-in the following material, neatly stapled:

- Your sign-off sheet. It should be on the front and include each partner's name and unique name.
- A typed set of answers to the questions from the in-lab and post-lab. If figures are required, neat, hand-drawn, figures are acceptable.
- A printout of your code from G2 and G6. If you modified any header files, be sure to include them also.