August Byrne

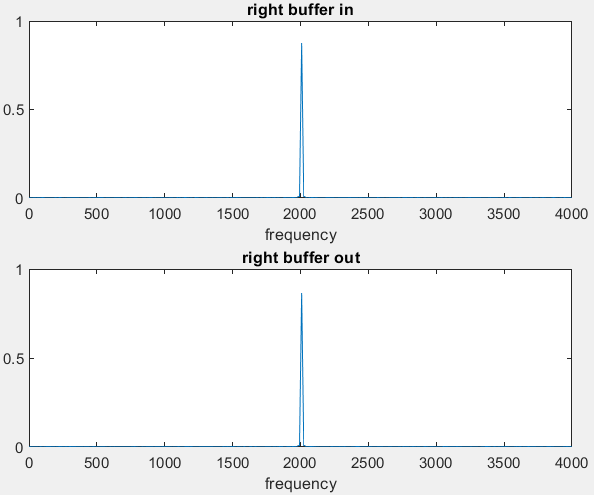
EECE 433

4/30/2021

Lab 2

Part 1:

1. The default sampling rate is 48kHz, using DSPSampleRateSet(CODEC\_SRATE\_CODE\_48K); while inside DSPInit() of AppDSP.c
2. Samples per block is set with #define DSP\_SAMPLES\_PER\_BLOCK 1536 inside App.DSP.h
3. We can use the board as a passthrough to pass audio from the inputs of the boards (converting using the ADC), and out of the output of the board (using the DAC).
4. dspInBuffer
5. dspOutBuffer
6. You can include your own DSP processing code in the dspTask() of AppDSP.c right before setting the output buffer to have the values in the input buffer.

Here is the frequency response of the input and output buffers to a 2kHz input sine wave. The only difficulty I faced when doing this was that I continued to have a bug where only the first data gathering of a debug session from the terminal gave me correct data. The output (as heard through sensitive earbuds) was always correct, but after the first collection, if I did not create a new debug session, the data I would now get would have a frequency response with a doubled frequency than what was inputted, and now with a lot of noise.

Part 2:

For this part, I used the functions arm\_mean\_q31 and arm\_rms\_q31 to get the mean and rms of an array that was in fixed-point type q31\_t. These functions both calculated fixed-point values, and calculated the mean to be 0, and the rms to be 5. The rms function in this case was producing the wrong result because of overflow, so I first had to bit shift all the values in the array and then calculated the rms. I then bit-shifted the result back and came out with an acceptable answer.

Part 3:

For this part, I used the functions arm\_mean\_f32 and arm\_rms\_f32 to get the mean and rms of an array that was in floating-point type f32\_t. These functions both calculated floating-point values, and calculated the mean to be 0, and the rms to be 5.47722578. The rms function in this case produced the correct result without the need to modify the input, and it calculated the rms to a more accurate value.