Assignment: Delta Sigma Simulation

Kaan Dincer, Yangruo Ma

1.0 Abstract

In this lab, we designed a 2nd Order Delta Sigma Modulator as described in Figure 14(b) in Hauser's Paper. We implemented our modulator with a sine wave input. We implemented the modulator in two cases: one with a "brick wall" filter with decimation ranging from factors 8 to 256, and one with a 2-stage decimator. The maximum Signal-to-Noise Ratio (SNR) and Estimated Number of Bits (ENOB) were calculated for each of the cases. The ideal Maximum SNR and ENOBs were also calculated for comparison purposes. SNR vs. Decimation Factor and ENOB vs. Decimation factor were graphed both for the brick wall and 2-stage decimator outputs for our modulator along with Equation 29 from Hauser's Paper. Finally, MACs/s were minimized while keeping ENOB within 0.5 bit-range of the brick wall output, in order to optimize operational cost of our Delta-Sigma modulator.

2.0 Introduction

This Lab explores the design of a noise-shaping Analog to Digital Converter (ADC) using a Delta-Sigma Modulator. We were concerned with the specific question: what if we had the same amount of quantization noise but instead we were able to modify it so the spectrum wasn't white. Figure 1 shows the difference of this noise-shaping ADC vs. a traditional ADC.

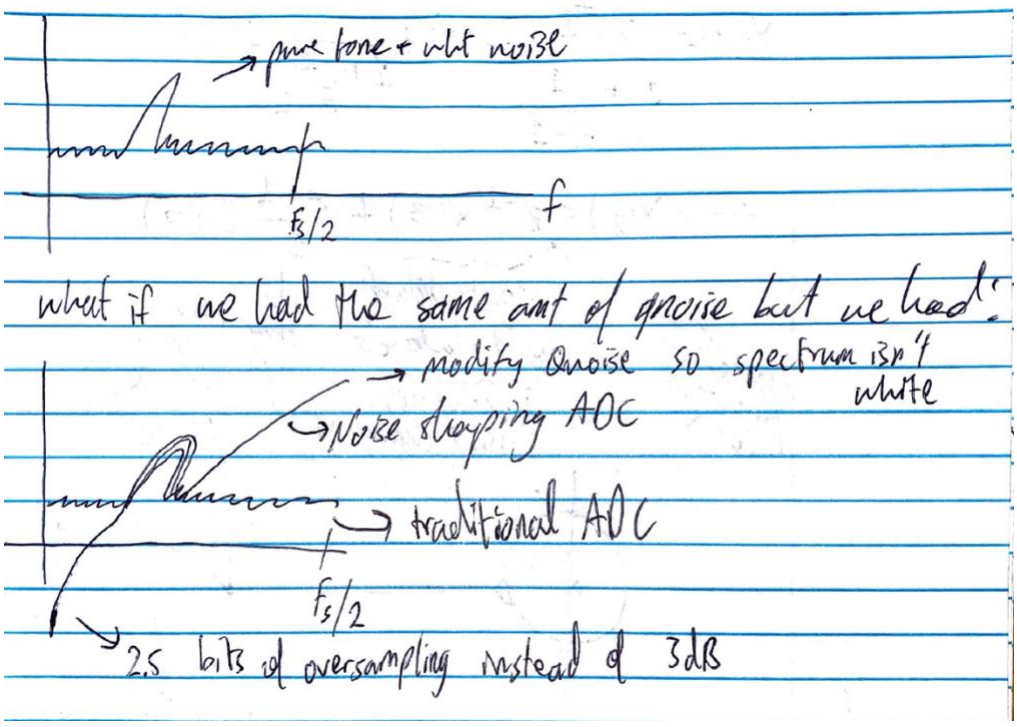

Figure 1: Noise Shaping ADC vs. Traditional ADC

We couldn't use the 1st order modulator as a 1st order modulator gave less noise shaping and posed problems, as it can be seen from Figure 2.

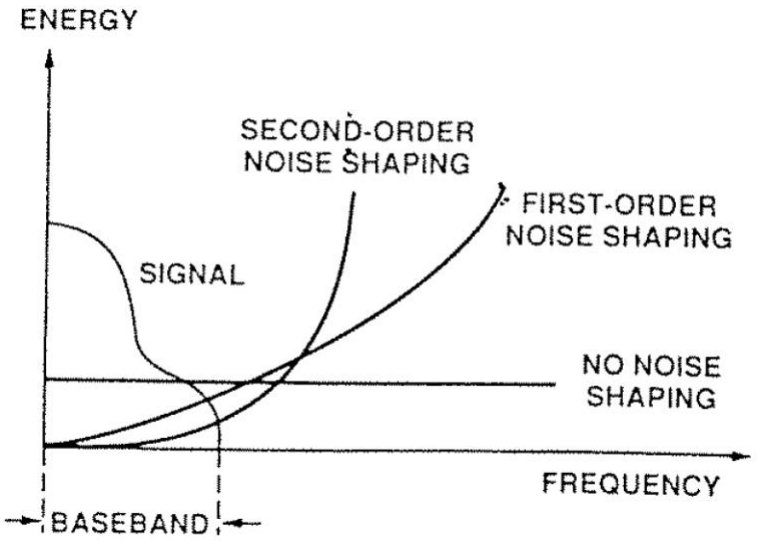

Figure 2: Spectra of signal and quantization error components in oversampled, quantized signal [1]

So in order to design this ADC we analyzed a common 2nd order delta-sigma modulator configuration seen in Figure 3, which was taken from Hauser's Paper.

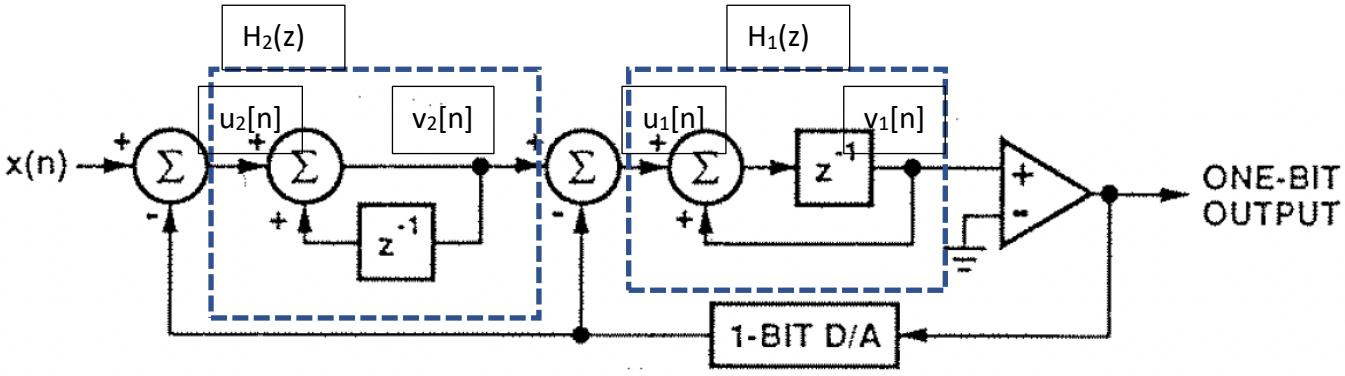

Figure 3: 2nd Order Delta-Sigma Modulator Block Diagram [1]

We analyzed the block diagram to derive the transfer functions for our inputs and Quantization Noise, given in Equation 1. The analysis of this block diagram was done as follows:

- $H_1(z)$:

$$v_1[n] = u_1[n-1] + v_1[n-1]$$

$$V_1(z) = z^{-1}U_1(z) + z^{-1}V_1(z)$$

$$H_1(z) = \frac{V_1(z)}{U_1(z)} = \frac{z^{-1}}{1-z^{-1}} = \frac{1}{z-1}$$

- $H_2(z)$:

$$v_2[n] = u_2[n] + v_2[n-1]$$

$$V_2(z) = U_2(z) + z^{-1}V_2(z)$$

$$H_2(z) = \frac{V_2(z)}{U_2(z)} = \frac{1}{1 - z^{-1}} = \frac{z}{z-1}$$

- The overall diagram:

$$[\,(X - Y)H_2 - Y\,]\,H_1 + Q = Y$$

$$Y = \frac{H_1 H_2}{1 + H_1 H_2 + H_1} X(z) + \frac{1}{1 + H_1 H_2 + H_1} Q(z)$$

$$Y = \frac{\frac{1}{z-1} \times \frac{z}{z-1}}{1 + \frac{1}{z-1} \times \frac{z}{z-1} + \frac{1}{z-1}} X(z) + \frac{1}{1 + \frac{1}{z-1} \times \frac{z}{z-1} + \frac{1}{z-1}} Q(z)$$

$$Y = z^{-1}X(z) + \frac{(z-1)^2}{z^2} Q(z)$$

Equation 1: Transfer Functions for Input and Quantization Noise

We see that the input is delayed by one sample, and that we have a 2nd order high-pass filter for quantization noise. This analysis allowed us to design a 2nd order Delta-Sigma modulator and implement it using a brick-wall filter and a 2-stage decimator to get a noise-shaping ADC.

3.0 Design

In order to simulate the 2nd order modulator we replicated the design on Figure 3 in LabView as shown in Figure 4, where we pass in the magnitude of our input signal into a for loop that runs until the whole magnitude array is iterated through. In the loop we use shift registers in order to implement the feedback loop seen in Figure 3. We use plus and minus operations to employ the summation operations on Figure 3 in LabView. Finally, for the modulator, we use a case structure to apply the integrator into LabView, in which we check if the output from the second feedback loop is greater than zero, and if so we pass in Vref, and if not we pass in -Vref into our magnitude array.

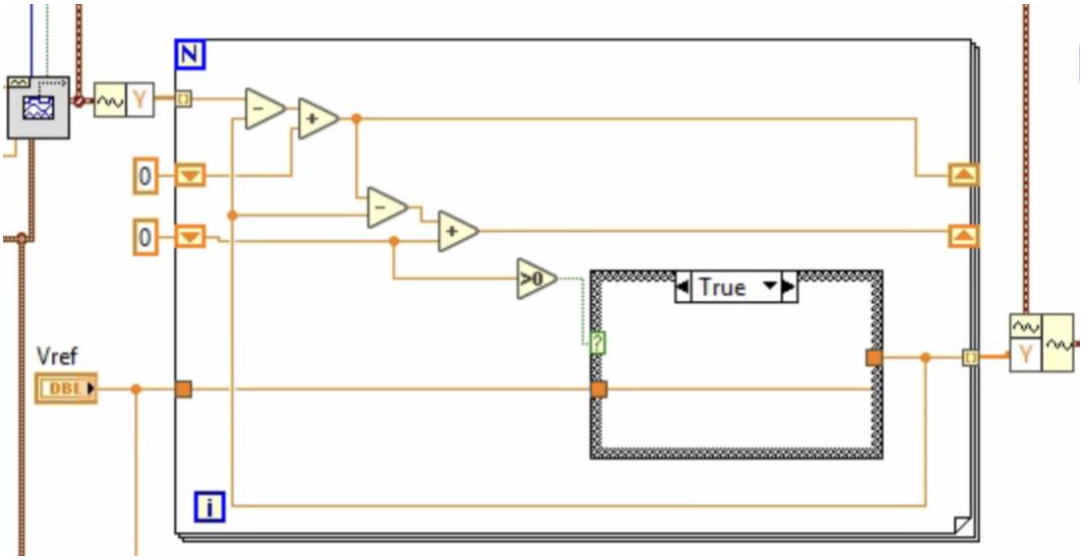

Figure 4: Implementation on LabView of the Block Diagram on Fig. 3

In the second part of our design we contrive the output from our 2nd order modulator and apply a brick-wall filter and a 2-stage decimator to it separately. In order to use decimation factors 8, 16, 32, 64, 128 and 256 a for loop was used which ran in iterations in our octaves (d) from 3 to 9. The loop and the general overview of the 2-stage decimation, brick wall and Equation 29 from Hauser's paper is shown in Figure 5.

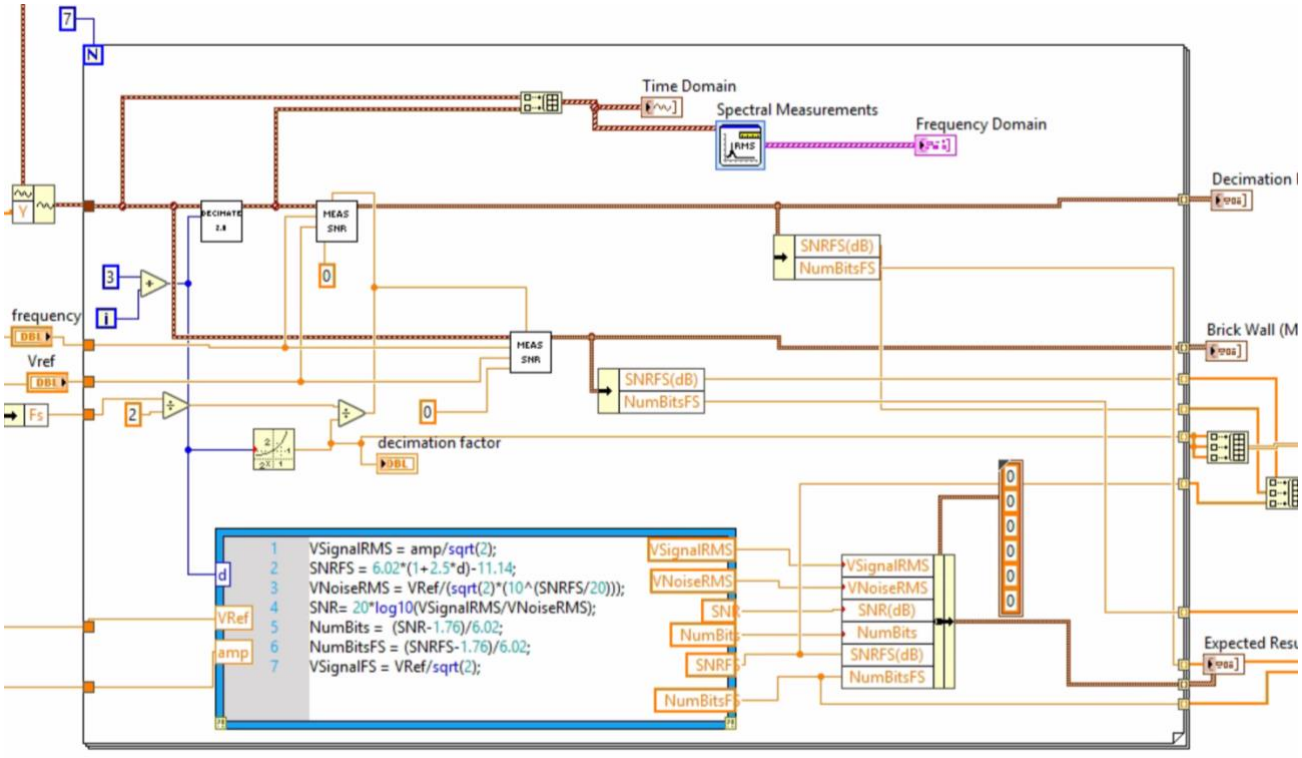

Figure 5: Brick-wall, 2-stage decimation and Eq. 29 Implementation on LabView

The brick wall filter was implemented using the MeasSNR sub VI created in a past assignment. The MeasSNR VI takes in a lower and upper bandwidth for integrating the

noise, the signal's frequency and maximum amplitude (Vref), as well as the signal for which SNR and ENOBs is to be calculated. The lower bandwidth was set to 0, the max amplitude was set to the Vref control created, and the frequency of our input sine wave was passed in. The upper bandwidth the noise is to be integrated is equal to the sample rate divided by 2 ($F_s/2$). But with decimation the upper bandwidth becomes the initial bandwidth, $F_s/2$, divided by the decimation factor, which is 2^d, where d was d = 3, 4, 5, 6, 7, 8, 9 in our case. The implementation of the brick wall filter is shown in Figure 6.

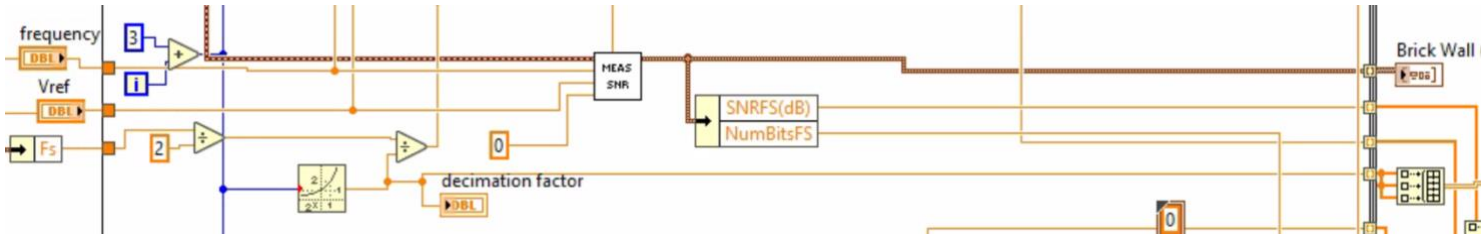

Figure 6: Brick Wall Filter implementation

The design of the 2-stage decimator in LabView was very similar to that of the brick wall, where the sub VI MeasSNR was implemented in the exact same way as the brick wall filter. However, for the 2-stage decimator another sub VI, Decimate 2.0, was used before MeasSNR. The modulated signal and the bit value (d) to determine the decimati0on factor were passed into the Decimate 2.0 VI. The Decimate 2.0 VI takes in an input, the modulated sine wave in our case, and passes it into a FIR filter, down samples it by a factor consistent with d passed in, passes it again into an FIR filter, and down samples it by a factor consistent with d passed in and the decimation factor of the initial down sampling. The Decimate 2.0 sub VI outputs the modulated signal that has undergone 2-stage decimation and passes it into the MeasSNR sub VI to output the desired results. This implementation of the 2 stage decimation is shown on Figure 7.

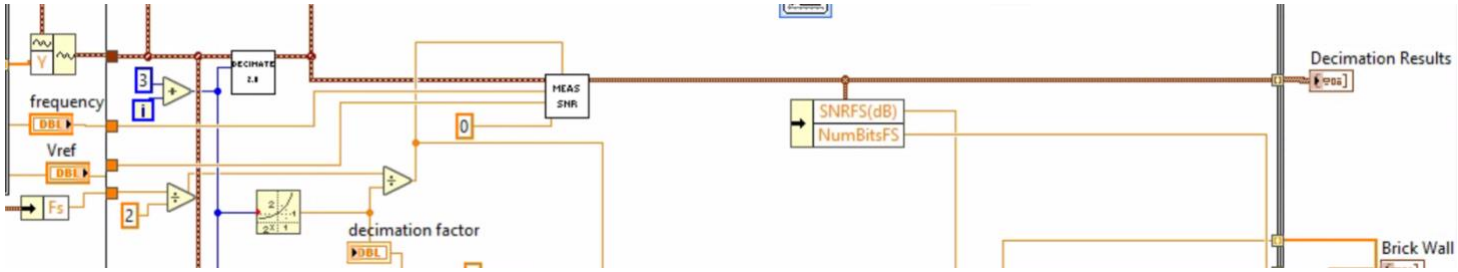

Figure 7: 2-Stage Decimation implementation

The expected results from our MeasSNR sub VI were manually calculated on LabView using a MathScript scripture and MATLAB programming language. The calculations for implementing Equation 29 from Hauser's paper is shown in Figure 8.

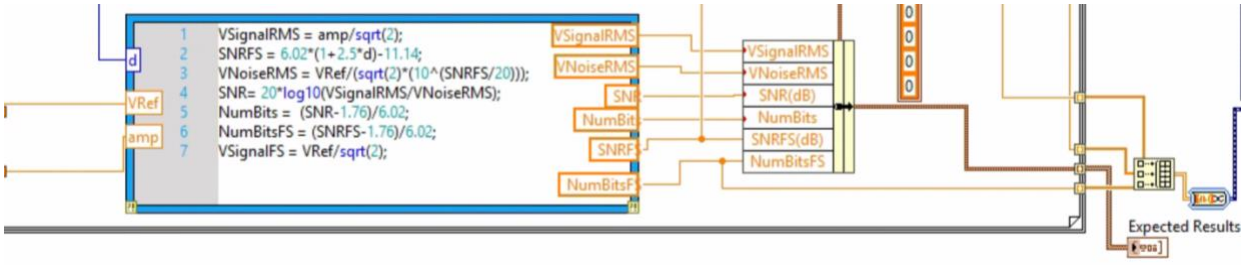

Figure 8: Calculations to Execute Eq. 29 in Hauser's Paper in LabView

Finally, the Estimated Number of Bits and Signal-to-Noise Ratio from the Brick wall filter, the 2-stage decimator and expected results were graphed vs. the decimation factor ($D = 2^d$) using the build array function and the Build XY graph function. The whole block diagram of the implementation on LabView is shown on Figure 9.

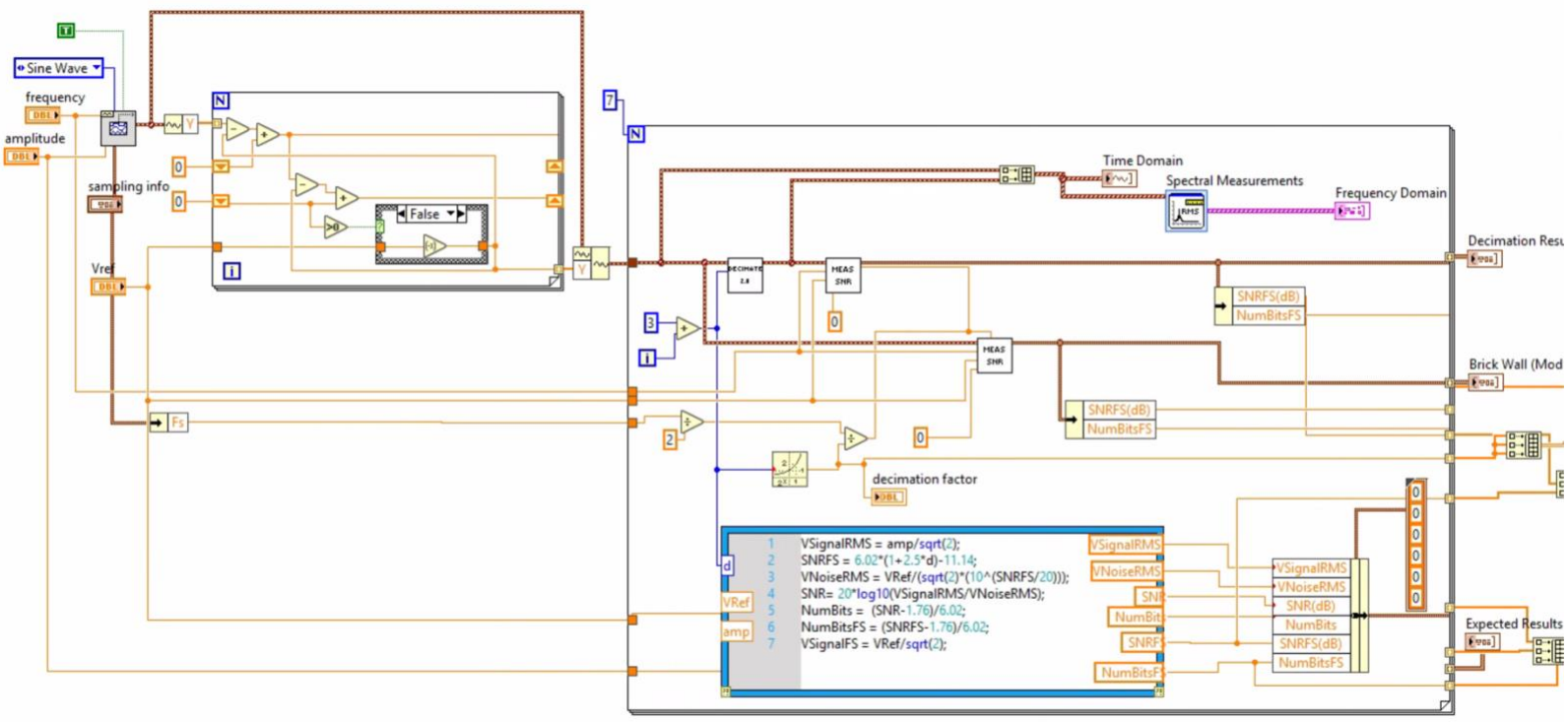

Figure 9: Delta Sigma Block Diagram

Finally, we tried minimizing the MACs/s in our simulation by adjusting the Stop Band and Pass Band Gain of both our filters in the Decimate 2.0 Sub VI. The MACs/s were minimized while keeping the Estimated Number of Bits within a 0.5 bit range of the modulator brick-wall's Estimated Number of Bits, which could be seen from Figure 13 below. The minimization of MACs/s while keeping ENOB within half-a-bit range resulted in a minimum of 1.42×10^9 MACs/s.

4.0 Operation/Testing

The testing of the design was done by graphing ENOBs vs. D and SNR vs. D for the three scenarios: brick-wall, 2-stage decimator and Equation 29 from Hauser's paper (expected results), as we knew the values and the graph we expected to see. The expected graph can be seen in Figure 10. Our resulting graphs can be seen in Figure 11, and Figure 12. We tested our implementation by comparing the graphs in Figure 10 and Figure 11. We also analyzed the graph in Figure 12 to test if the estimated number of bits we were getting for each decimation factor was accurate.

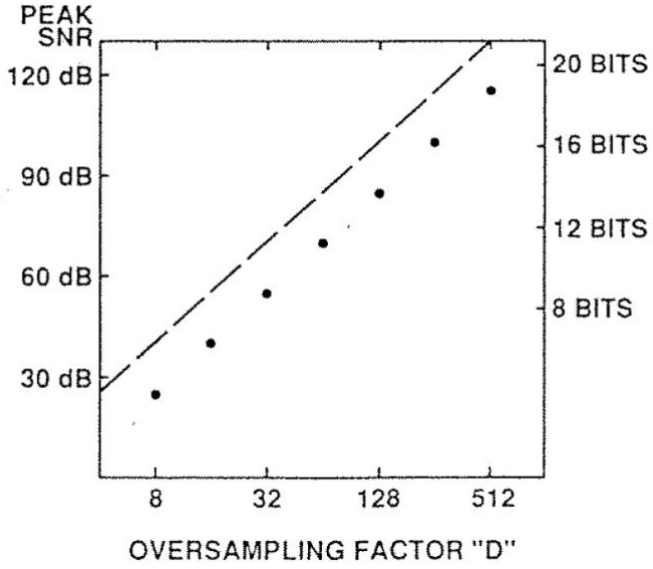

Figure 10: Expected Quantitative Error in Additive-Quantization-Noise

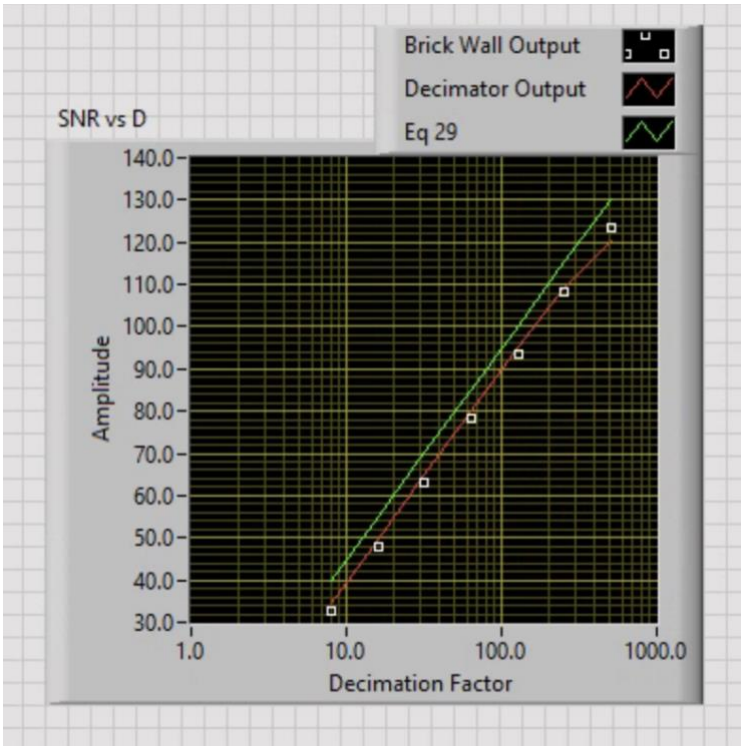

Figure 11: Signal-to-Noise Ratio vs. D

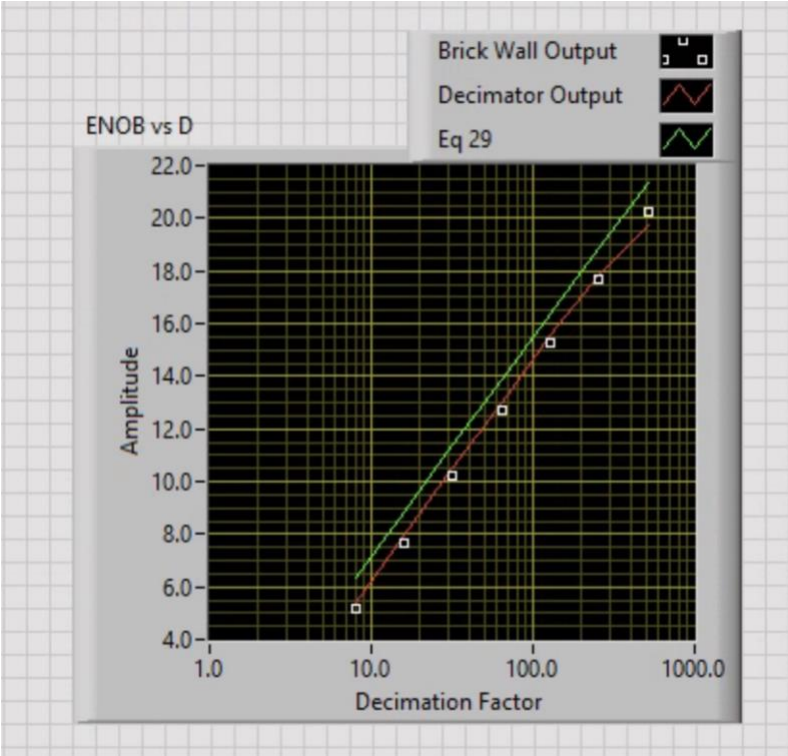

Figure 12: Estimated Number of Bits vs. D

MeasSNR sub VI also outputted the Max SNR and ENOB values for the brick wall filter, 2-stage decimator and expected results as shown in Figure 13. We tested whether the max SNR output by the two-stage decimator and brick wall were 7 dB less than the expected results. We compared these outputs to evaluate the results of our simulation.

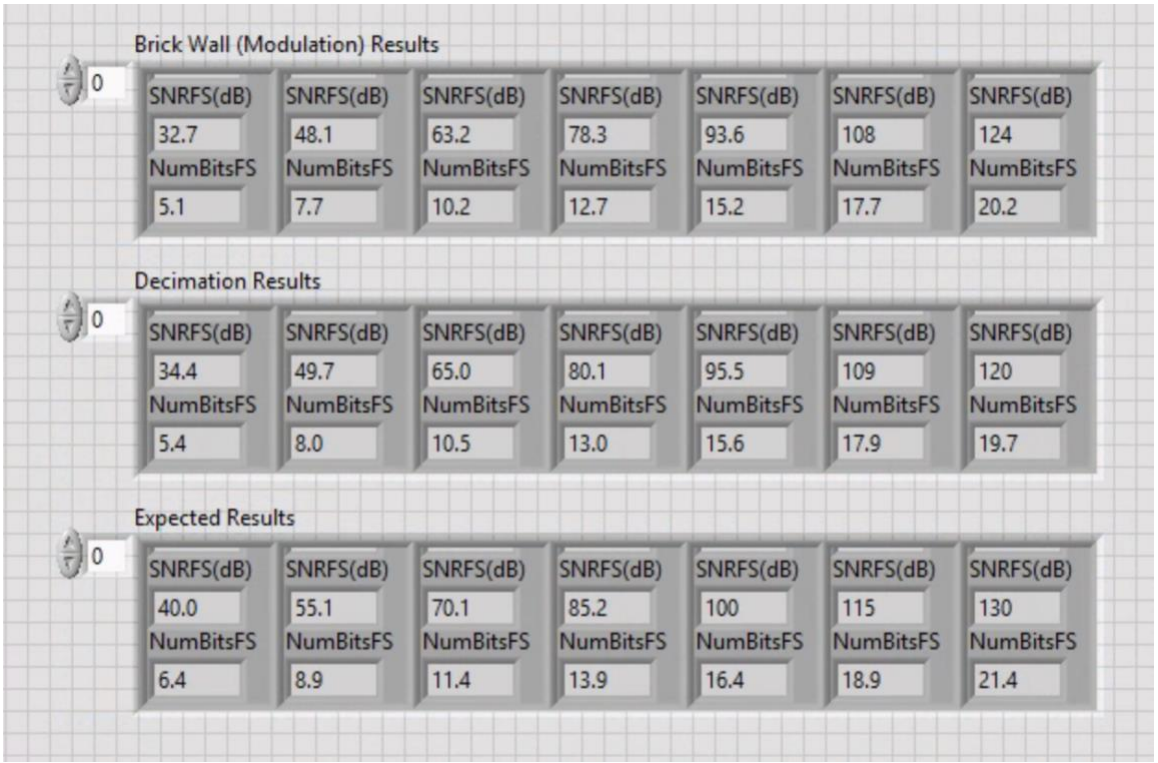

Figure 13: SNR and ENOB Values

As it could be seen from Figure 13 the Estimated Number of Bits are kept in 0.5 bit range of the brick-wall filter while minimizing MACs/s.

5.0 Discussion/Conclusion

Looking at the SNR vs. D graph our simulation produced in Figure 11 and comparing it to the expected graph in Figure 10, we see our simulation produced accurate max SNR results. Moreover, we can double check this by looking at Figure 13, as the max SNR values from the brick wall and two-stage decimator outputs should be 7 dB less than those of Expected Results. Comparing max SNR in Expected Results with two-stage decimation and brick wall filter we can verify that max SNRs outputs of our simulation are approximately 7 dB less than those of our calculated (expected) results.

Looking at the ENOB vs. D graph in Figure 12 we see that our Estimated Number of Bits were consistent with decimation factors our signal was decimated by. Plus, our estimated number of bits from our decimator output were in half-a-bit range of the estimated number of bits from our brick wall modulator output. Therefore, we can conclude that our simulation worked as expected, giving us accurate and correct results.

6.0 References

[1] MA. W.. Hauser, "Principles of Oversampling A/D Conversion," J. Audio Eng. Soc., vol. 39, no. 1/2, pp. 3-26, (1991 January/February.). doi: