NAME: ARNAB KUMAR MONDAL.

UIN: 326006709

LAB - 3

In this lab, we implemented a low-pass IIR filter for a speech signal. The sampling rate of the speech is 44100 Hz. But the human ear processes speech at 3400 Hz. Therefore, we develop a filter to eliminate information out of the useful bandwidth. The parameters of the filter should be: Ap = 1db, As = 60db, Fs = 44100Hz, Fp = 3400Hz, Order = 6.

The IIR filter generated by the Fdatool looks like this:

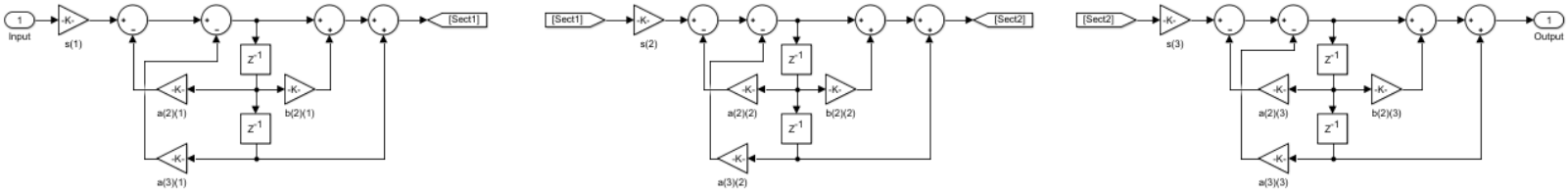


Figure 1 : The IIR Filter generated by the FDA Tool

The three stages of the IIR Filter are as follows:

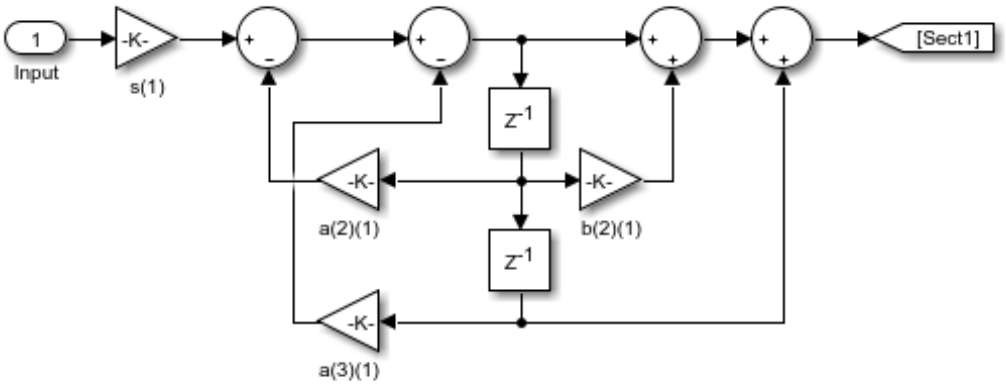


Figure 2 : The 1st stage of the IIR filter

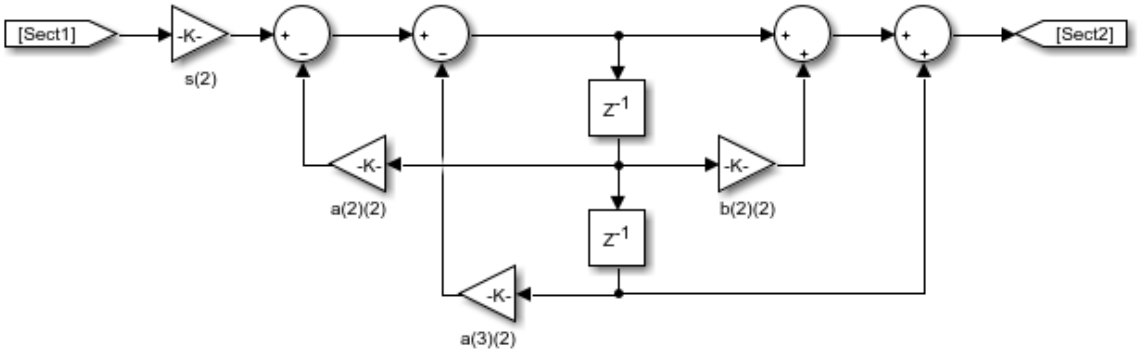


Figure 3: The second stage of the IIR filter.

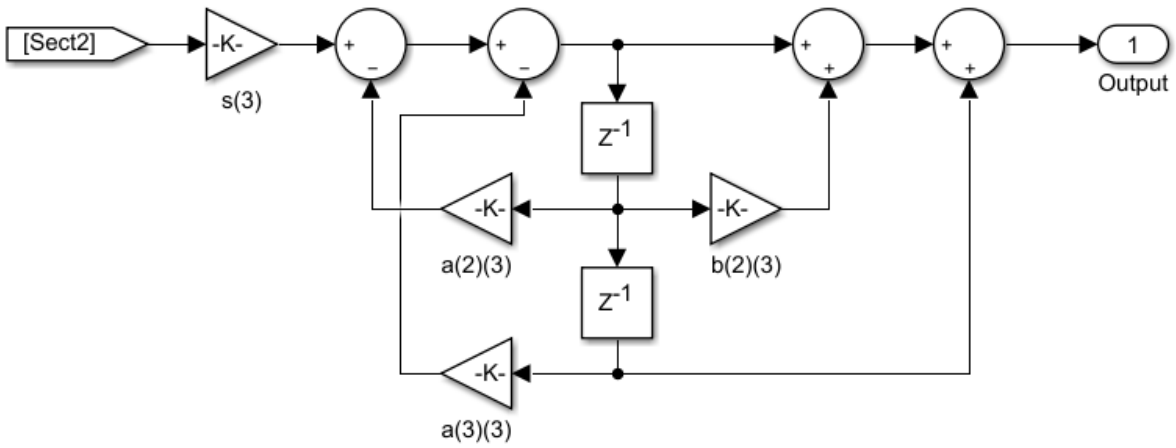


Figure 4: The 3rd stage of the IIR Filter.

The filter given from the FDA Tool gives us an exact IIR Filter with the coefficients being derived from the theoretical values. But the theoretical coefficients are cumbersome and impractical and therefore we need to quantize the input and output to determine least number of bits used in the I/O to satisfy the SNR to be above 35. Finally, we quantize the internal nodes.

The swing 's' gives information about the bits needed for the integer part.

The gain in the filter given is now modified to quantize the required number of bits for precision. So we use the function: "round(original gain coefficient * 2^(ck-1))/(2^(ck-1))" where 'ck' is the number of precision bits needed for quantization.

Finally we compare the quantized output from the theoretical output of a band limited white noise using a spectrum analyzer as shown in Figure 5 and we found out that for 's' as 1 'ck' as 11 we get difference in pass band less than 0.1 dB and less than 1 dB in stop band as shown in Figure 6 .

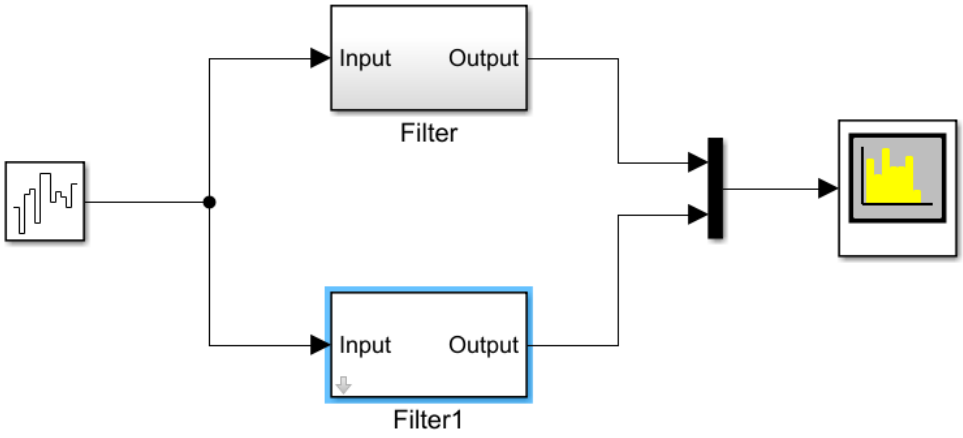


Figure 5

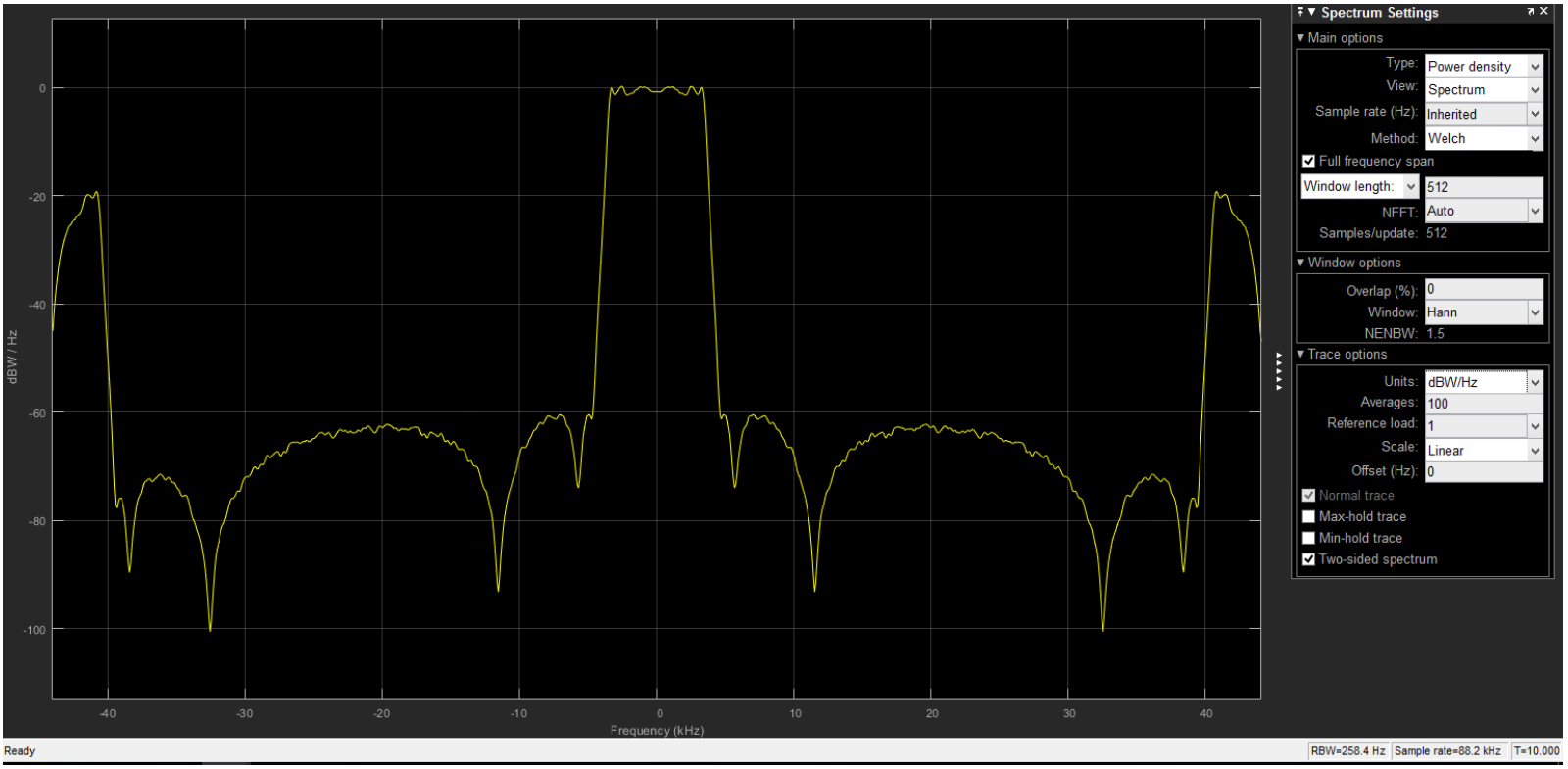


Figure 6: Spectrum Analyzer output

After realization of the model and quantizing it as per given instructions in the lab manual, we need to change k,s and ck values to get a minimum SNR of 35 dB even after complete bus quantization. I found the values of ***k=11, s=5 and ck=11*** to be the best possible combination.

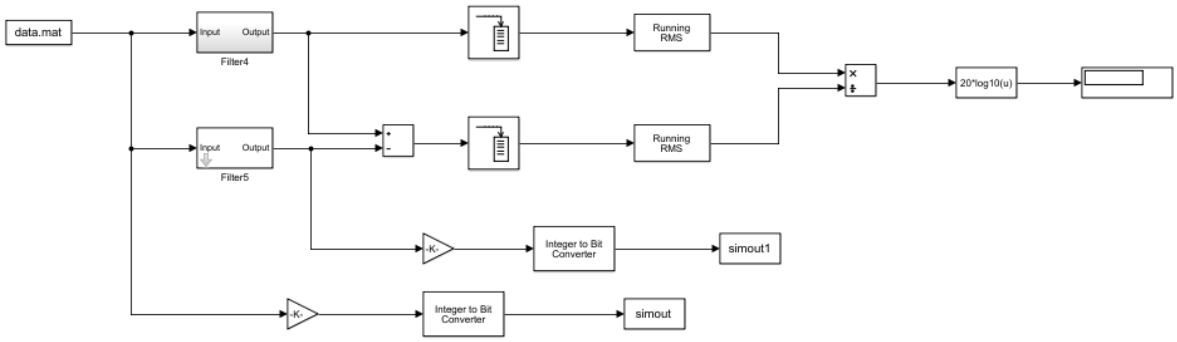


Figure 7: Final Simulink model for IIR SNR and I/O length testing/calculations

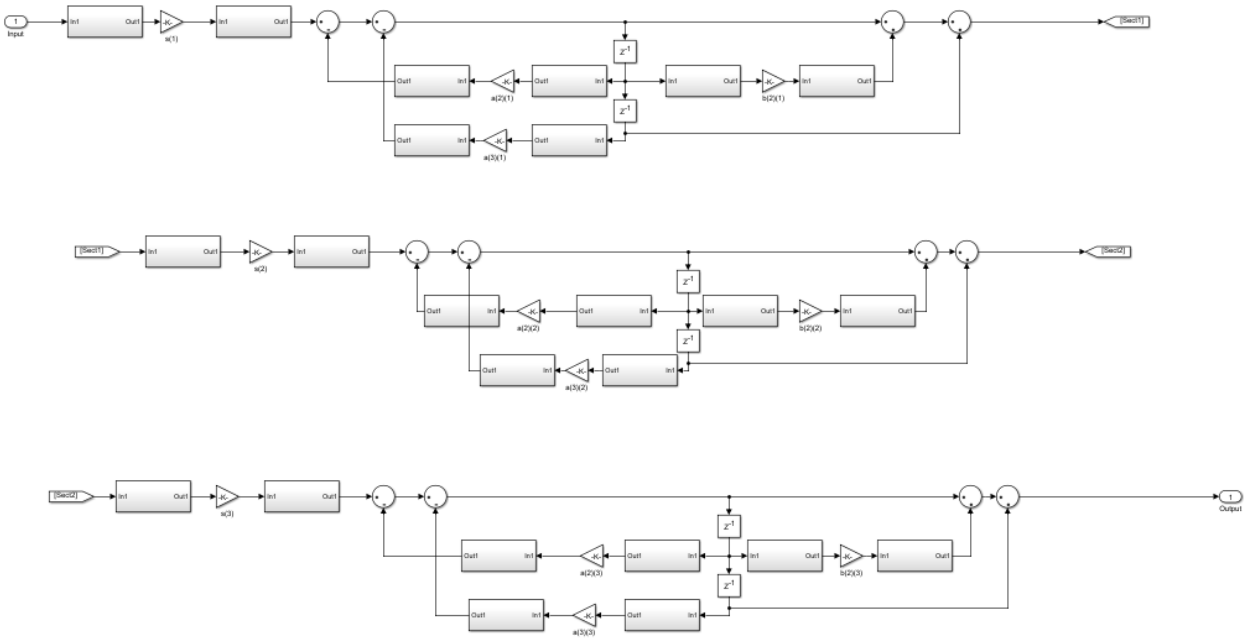


Figure 8: Quantized IIR section (3 stages) of Figure 7

2 files “Input_vectors.txt” and “Output_vectors_matlab.txt”, which are directly obtained from the simout2 and simout3 output blocks in the IIR filter as shown from the figure. To get these files a gain block of gain 1024 followed by a module called “Integer to Bit converter” which converted the rounded integer values to bits. A simple MATLAB program m11.m was written to save them to the 2 files as mentioned above. Simout3 is basically the output of the IIR filter when it gets an input of simout2. The code is as follows.

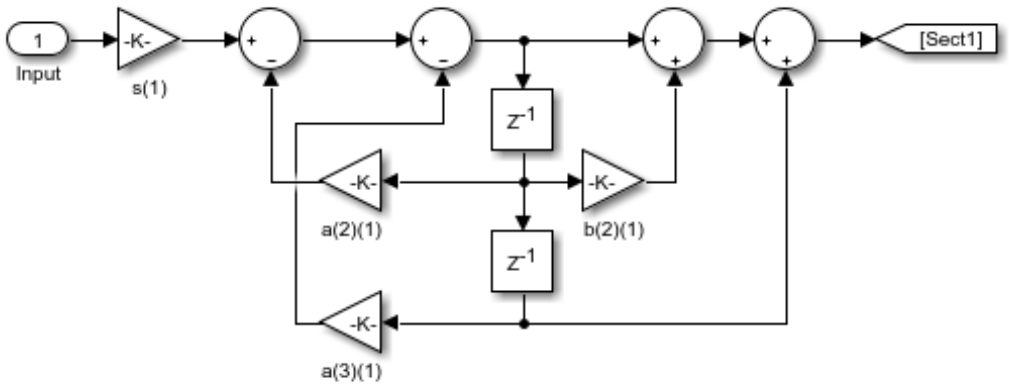
```
A = simout2.data;
B = simout3.data;
% Write this to file.
fid = fopen('Input_vectors.txt','w');
fid_1 = fopen('Output_vectors_matlab.txt','w');
k=size(A,1);
for i = 1:size(A,1)
fprintf(fid,'%d',A(i,:));
fprintf(fid,'\n');
fprintf(fid_1,'%d',B(i,:));
fprintf(fid_1,'\n');
end
fclose(fid);
fclose(fid_1);
```

Verilog Code :

From the IIR filter in Simulink we find that the IIR filter has been divided in 3 stages. So while designing the Verilog code we also used 3 modules to define the 3 sub-sections in the IIR filter named as iir_filter_1.v, iir_filter_2.v and iir_filter_3.v respectively. The modules were combined together in the

IIR_main.v file to create the IIR filter and to test its functionality we passed the IIR filter through a testbench known as iir_test_bench.v. A brief description of the IIR filter modules are given below.

1. iir_filter_1.v, iir_filter_2.v and iir_filter_3.v : These 3 modules are used to design 3 subsection in the 6th order IIR filter. A subsection looks like this :



The input through this module is a 32-bit signed bitstream, reset and clock. The output is a 32-bit signed bitstream. The Z^{-1} module is basically a delay element of unit delay which can be realized through a D flip flop. The gain modules are in the form of 'round(ideal value*2^(ck-1))/(2^(ck-1))'. This is in the fractional form and to represent it as a signed bit-stream we hardcode the 'round(ideal value*2^(ck-1))' value in the form of signed integers and based on the optimal value of 'ck' we can evaluate the gain modules of the individual blocks. For the division by 2^(ck-1) we can shift the gain value by (ck-1) units arithmetically towards the right.

2. IIR_main.v : This module is used for combining all the 3 submodules and form the complete IIR filter. The input through this module is a 32-bit signed bitstream, reset and clock. The output is a 32-bit signed bitstream.

3. iir_test_bench.v : This module drives the simulations for the IIR_main file. It gets the inputs from the 2 files 'Input_vectors.txt' and 'Output_vectors_matlab.txt'. the 2 files are derived from the simout2 and simout3 blocks we have in matlab. The text files consists of 32 bit signed bitstream of the input and the output values of the IIR filter in simulink. The data from the 'Input_vectors.txt' is passed through the IIR_main file and the output from the IIR_main module is compared with the bitstream from 'Output_vectors_matlab.txt'. The difference is ranging from 0-6 for the entire data set thereby signifying that in some cases the last 2-3 bits are altering. One of the causes for the non-zero difference is because we do not know how the 'Integer to bit converter' block operates internally and therefore our algorithm in Verilog may not match with the Simulink output.

The simulations were done in MODELSIM PE.

For the synthesis we use design vision and for the technology library we have the files osu018_stdcells.lib and generic.sdb. The timing, area and power reports were found out. The IIR synthesis module looked like this.

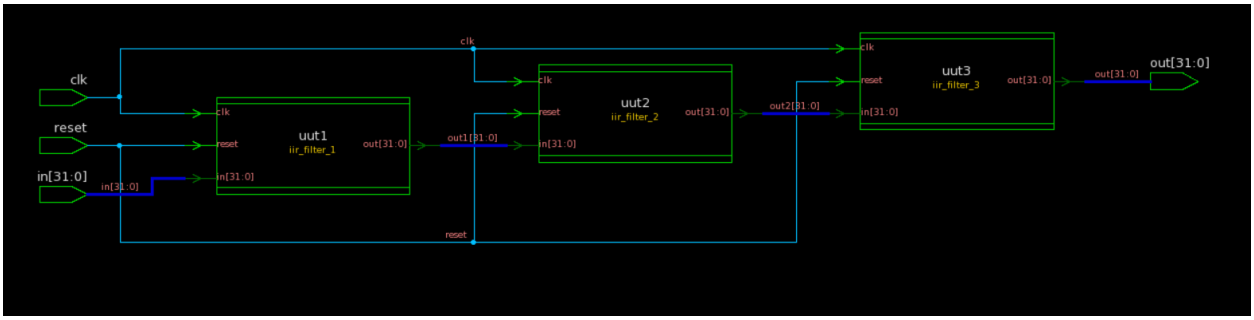


Figure 9: Synthesized IIR filter in design vision.

CALCULATIONS:

SNR before quantization: **37.09 dB**
SNR after quantization: **35.66 dB**
SNR degradation is: 37.09 - 35.66 = **1.43 dB**

K: **11**
S: **5**
ck: **11** (precision bit length for quantized coefficients)
Bit length used for Verilog computations: **32**
Total cell area: **353022.000000**
Data arrival time: **24.62**
Cell Internal Power = 46.7607 mW (48%)
Net Switching Power = 50.9033 mW (52%)
Total Dynamic Power = 97.6640 mW (100%)
Cell Leakage Power = 674.4273 nW
P.S.: All Verilog files and power, area, timing reports are attached with this pdf.