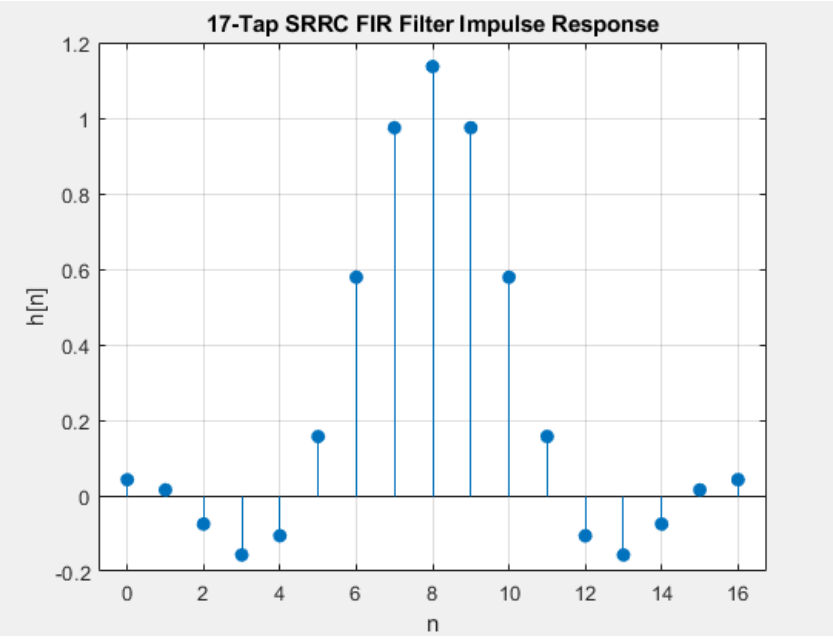
DSP in VLSI

HW2

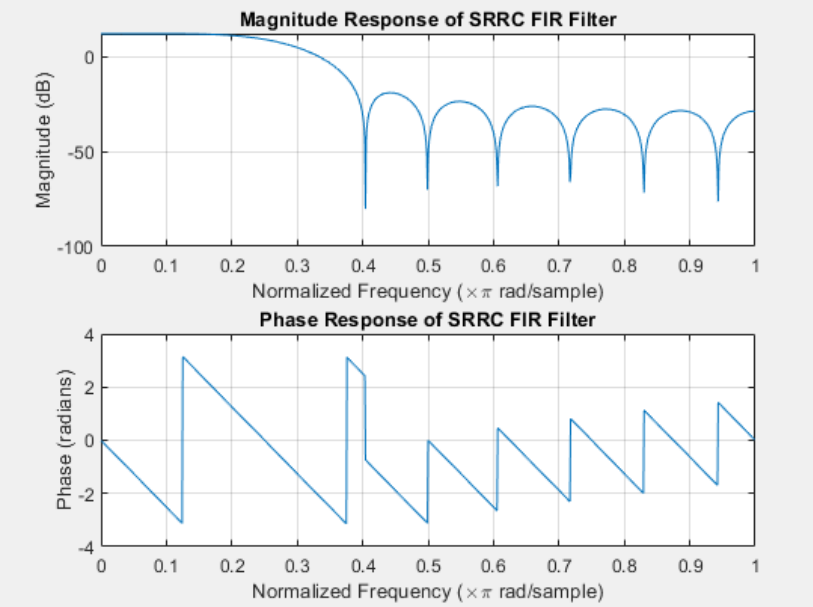
電子所 ICS 組, R13943015, 張根齊

1. (Step 1) Please use Matlab/Python to draw the impulse response and frequency response (containing magnitude and phase) of the 17-tap square-root raised-cosine FIR filter. Note that you need to use dB scale for the magnitude of the frequency response and use radian for the phase of frequency response versus normalized frequency. The x-axis must be marked with the correct label (20%). Please explain whether the filter is high-pass, band-pass or low-pass and why. (5%)

1.1 impulse response:



1.2 Frequency response:

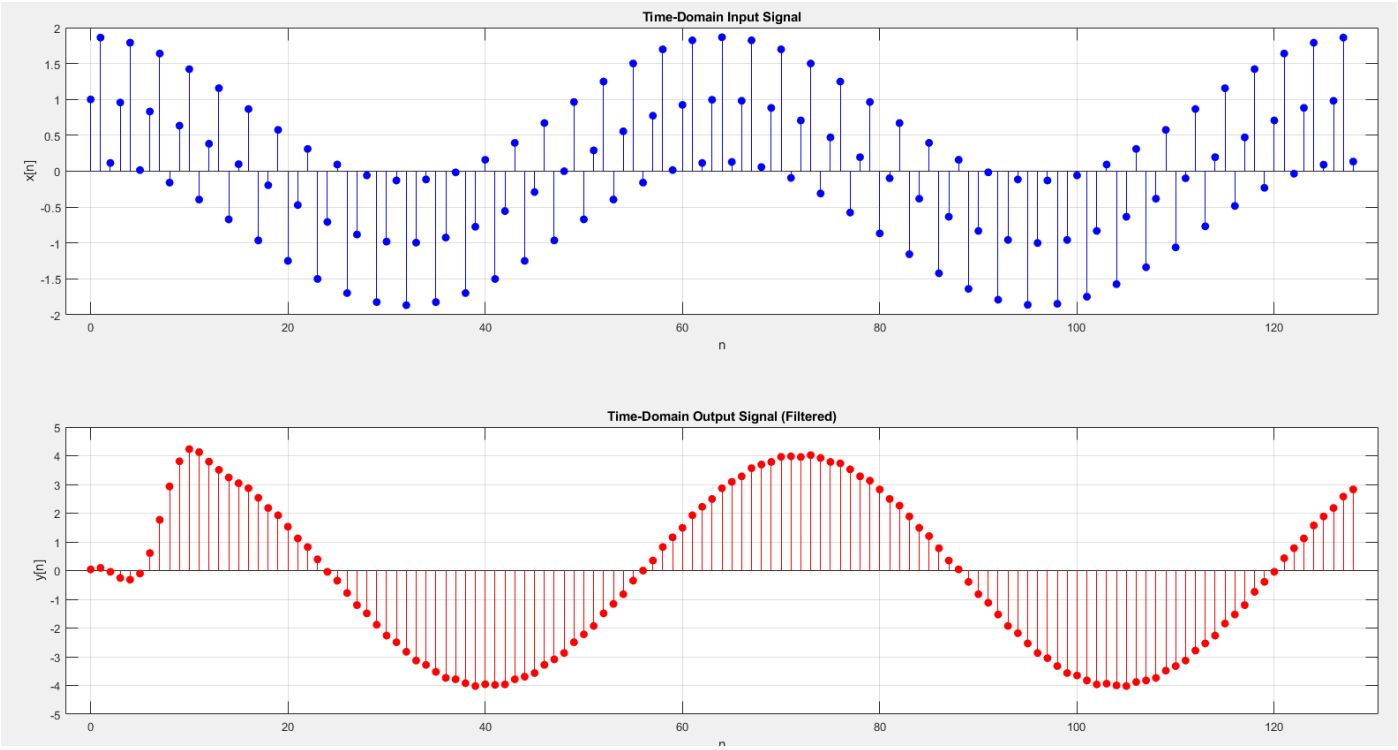


1.3 high-pass, band-pass or low-pass:

I think it is a low-pass filter because the magnitude of the FIR filter at low frequency (0π) is higher than at high frequency (1π).

2. (Step 2) Draw the time-domain input and output waveforms after you feed 128-point input (20%)

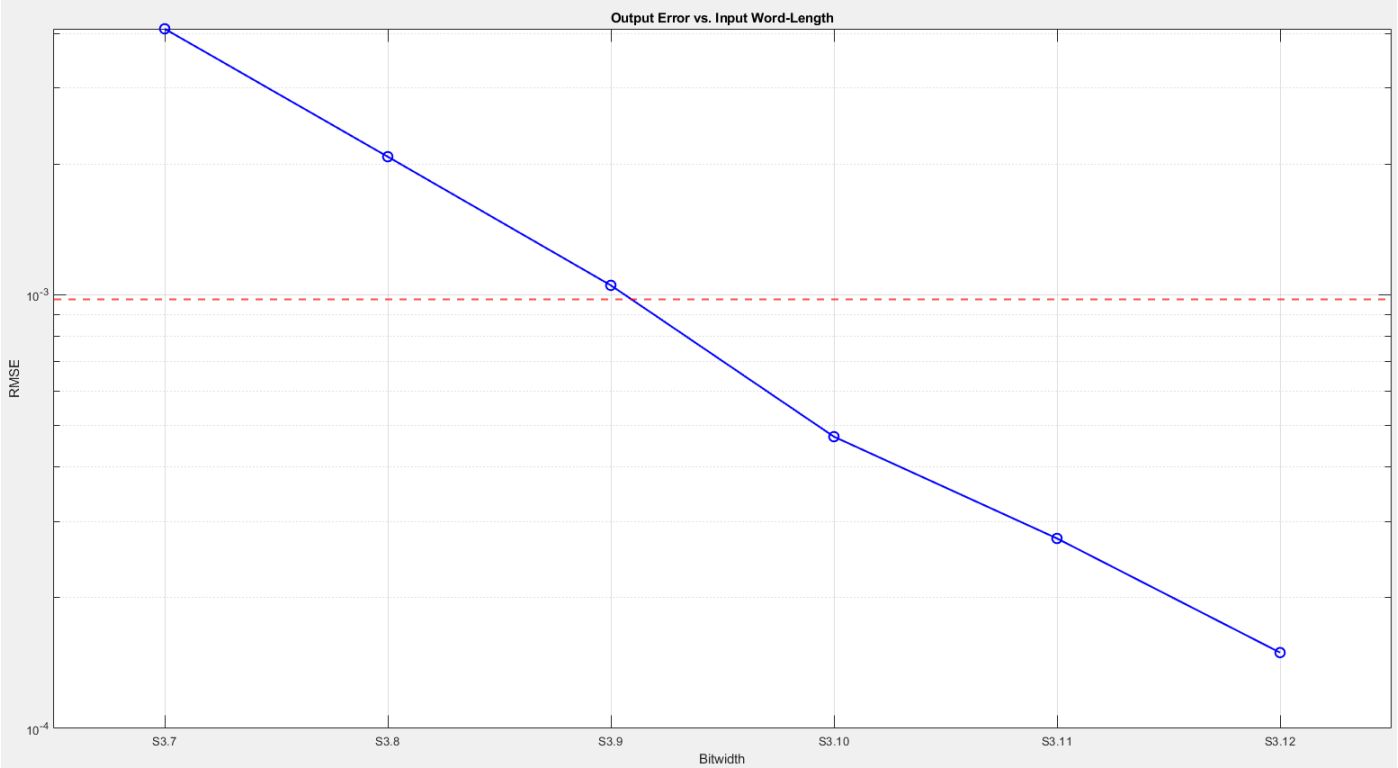
Output time domain waveform is obtained by transposed form FIR filter.



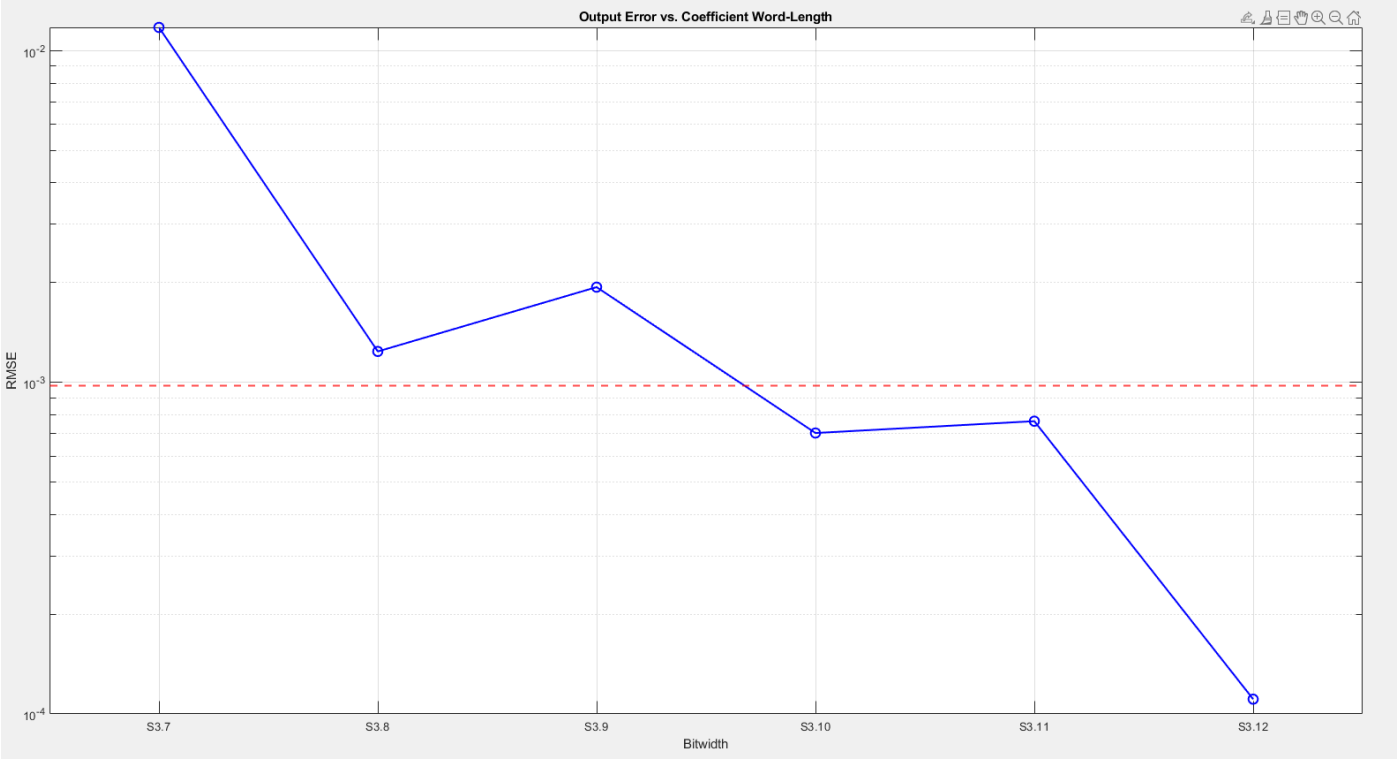
3. (Step 3) To show how you determine the word-length, please use the word-length of coefficients as the X-axis and error as the Y-axis. Scan the quantization error versus the word-lengths for at least 6 settings. (Four figures each having at least 6 word-length settings in its x-axis). According the following simulation results, mark the final word-length settings in the block diagram of the transposed form FIR filter. (20%)

The red dotted line is required RMSE.

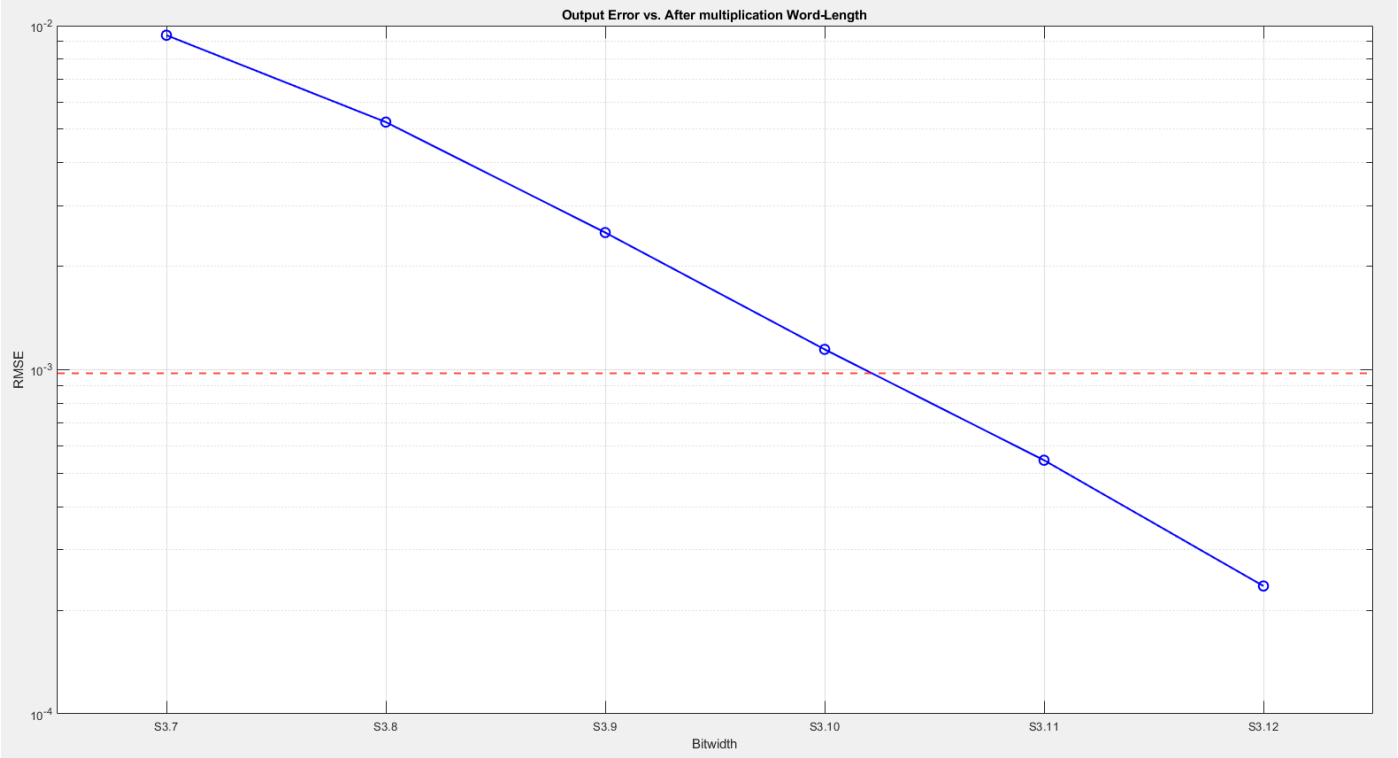
a. Output error versus input word-lengths



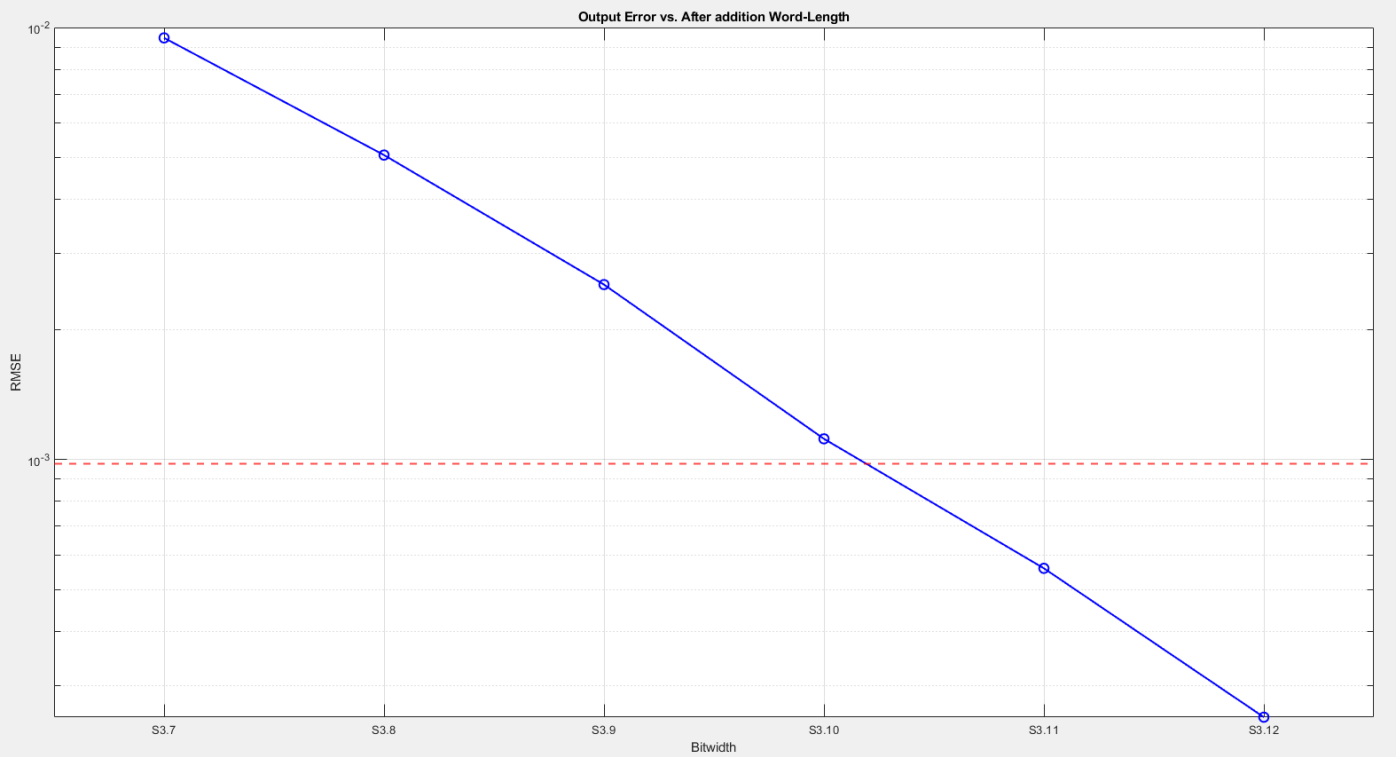
b. Output error versus coefficient word-lengths



c. Output error versus word-lengths after multiplication



d. Output error versus word-lengths after addition



e. Mark the final word-length settings in the block diagram of the transposed form FIR filter.

If I choose S3.11 for the bitwidth after multiplication and addition, the RMSE will be higher than the required RMSE (considering the combined quantization effect of the input and filter coefficients). Therefore, I chose S3.12 for the bitwidth after multiplication and addition.

Should use sequential quantization.

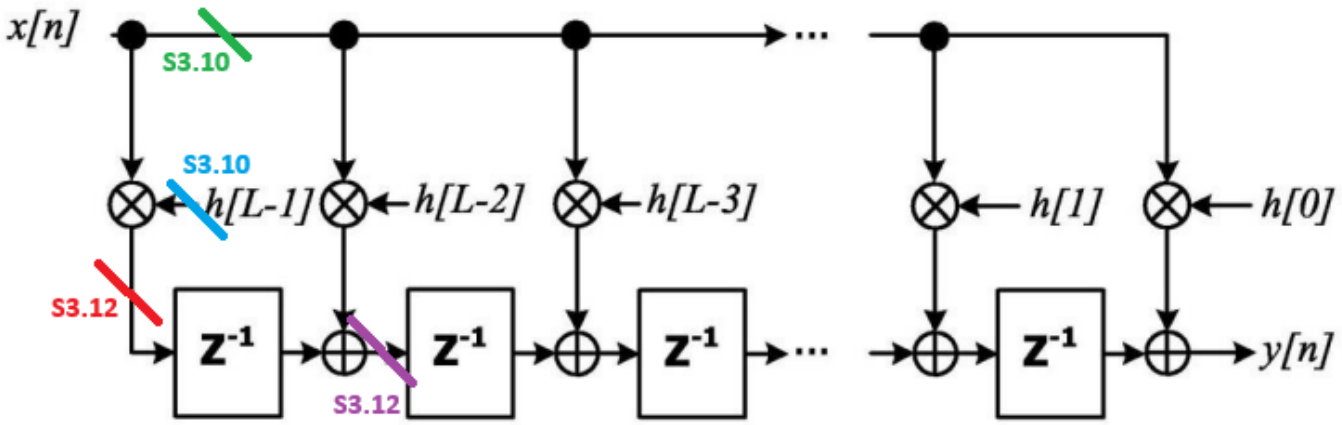
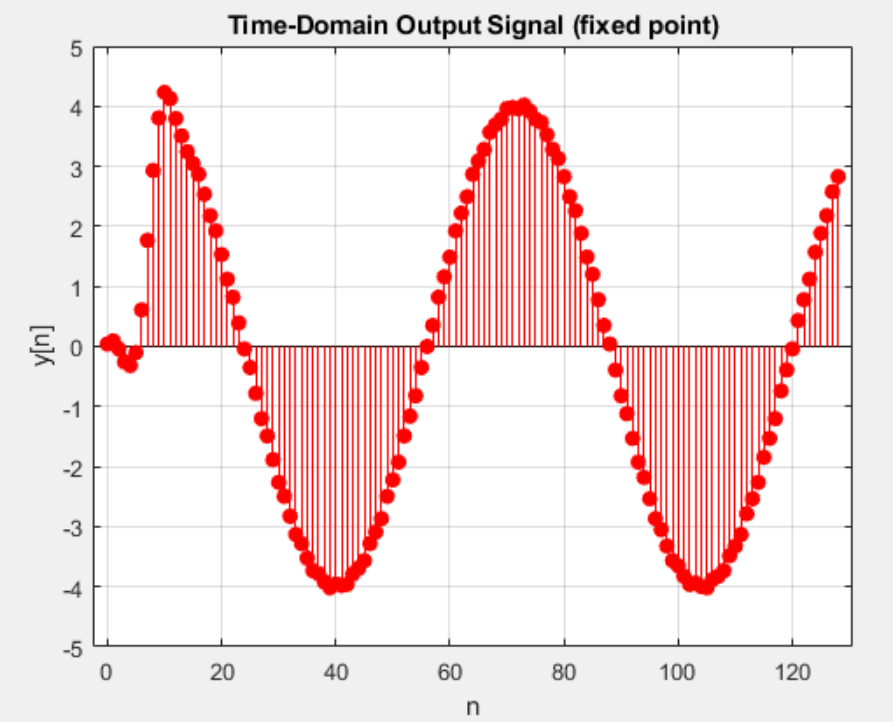


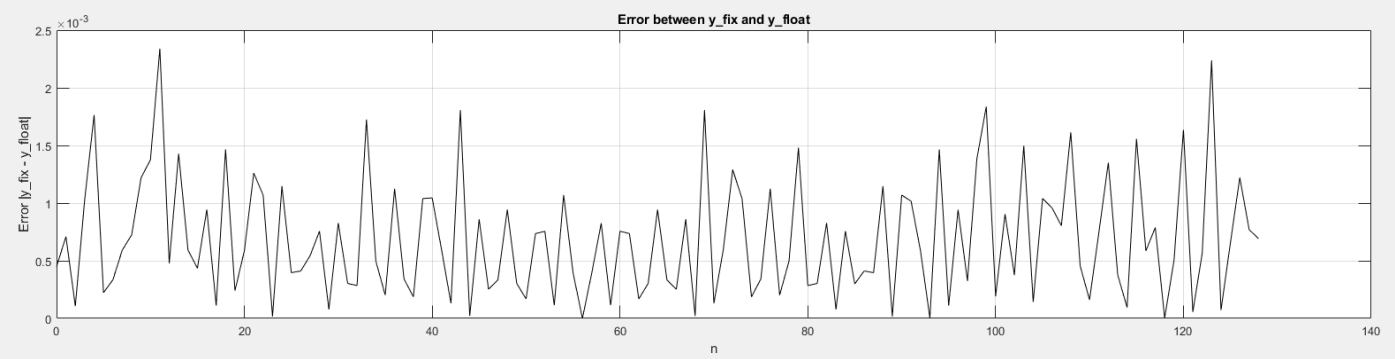
Fig. 2 FIR filter in transposed form.

4. (Step 3) Show the fixed-point output of time domain signal. (10%) Compare to the floating-point output and draw the error. (10%) Show the frequency response of the fixed-point direct-form FIR filter.(10%) Compare to the frequency response of floating-point results and draw the error of the magnitude response in passband (within 3dB bandwidth), expressed in dB (5%)

4.1 Show the fixed-point output of time domain signal:

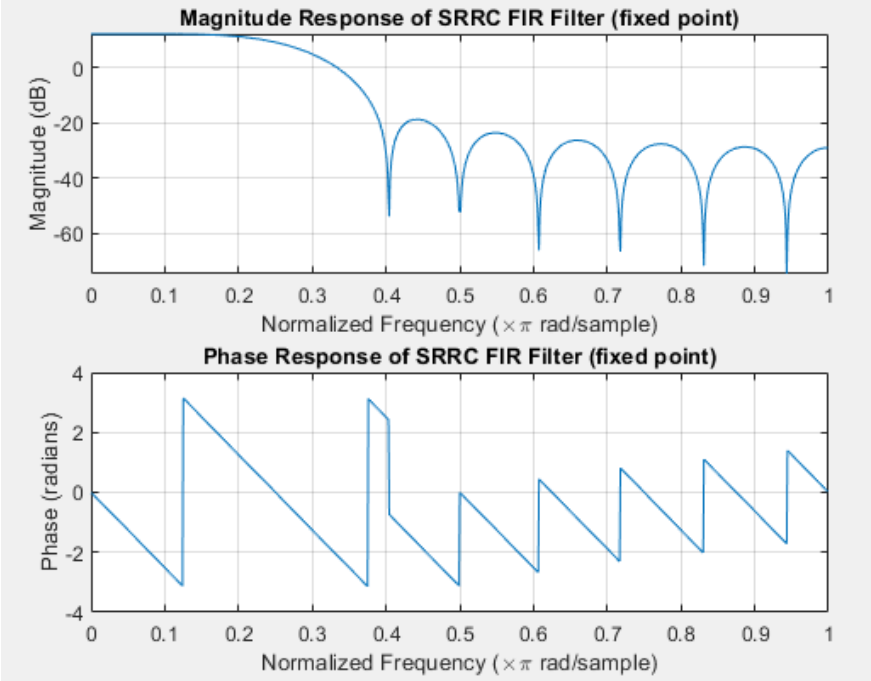


4.2 Compare to the floating-point output and draw the error:

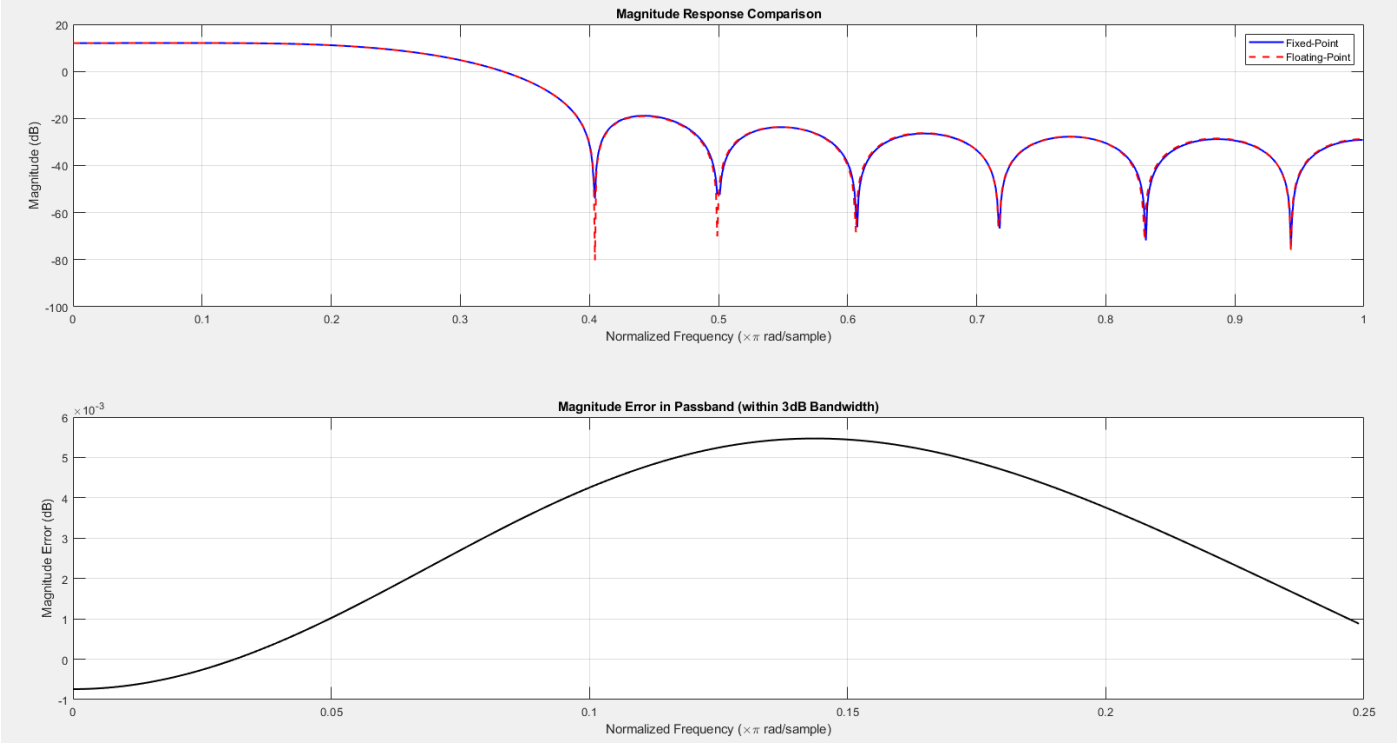


Rmse: 8.6363e-04

4.3 Show the frequency response of the fixed-point direct-form FIR filter.

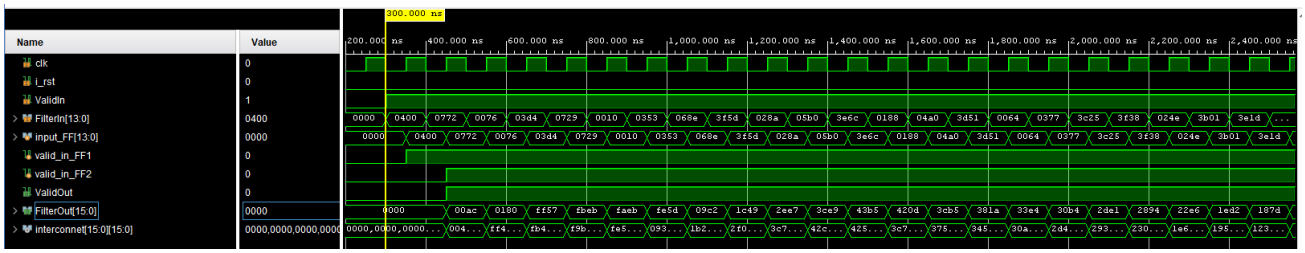


4.4 Compare to the frequency response of floating-point results and draw the error of the magnitude response in passband (within 3dB bandwidth), expressed in dB:

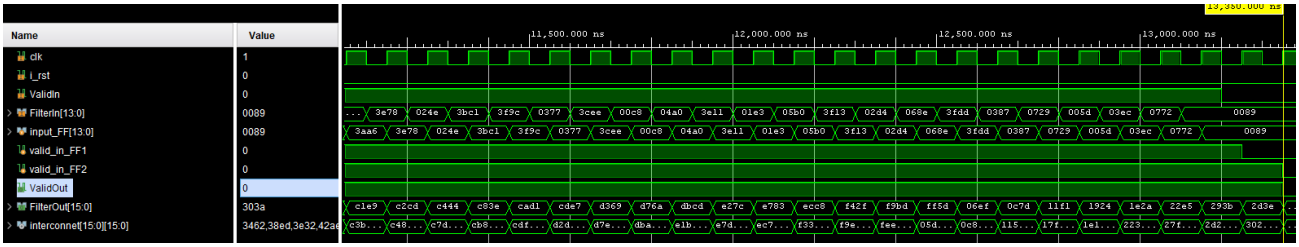


5. (Step 4) Print out the behavior timing diagram for the first 20 samples and the last 20 samples (10%). Show the errors of hardware outputs and Matlab floating-point results of transposed from FIR versus for 0 ≤ n ≤ 128 + $\boldsymbol{d}$ by figures, where $\boldsymbol{d}$ is the latency of your design and is not constrained, but you have to observe all the outputs until output becomes 0. (10%)

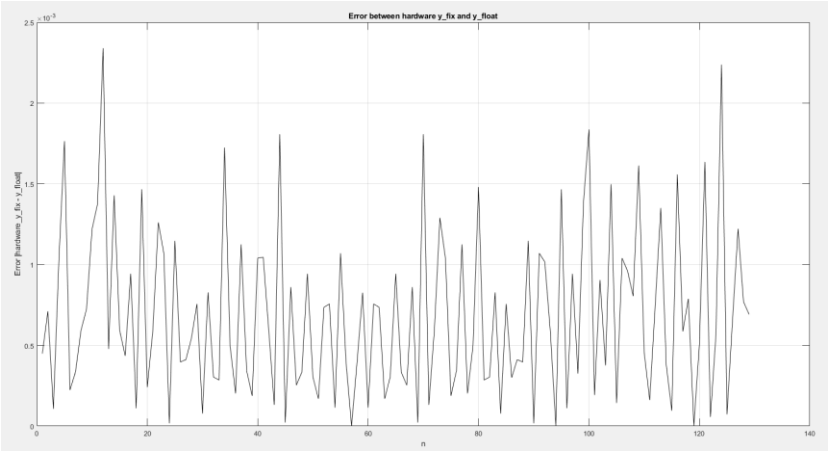
5.1 first 20 samples:



5.2 last 20 samples:



5.3 error between hardware output and matlab floating point output:



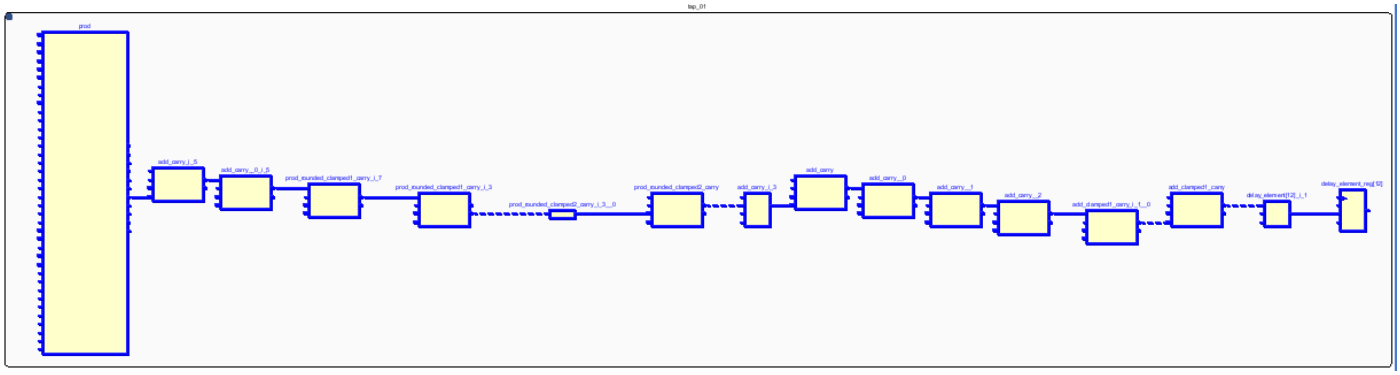
Rmse: 8.6363e-04

6. (Step 4) Paste the timing report about the critical path. Show the numbers of adders and multipliers in the critical path. (5%). List your timing constraint to achieve the highest operating frequency (5%). Print out the post-SYNTHESIS timing diagram for the first 20 samples and the last 20 samples (10%). and show the errors of hardware outputs and Matlab floating-point results of transposed from FIR versus for 0 ≤ n ≤ 128 + $\boldsymbol{d}$ (10%).

6.1 timing report about the critical path

Summary

Name	Path 1
Slack	0.720ns
Source	tap_01/prod/CLK (rising edge-triggered cell DSP48E1 clocked by clk {rise@0.000ns fall@6.500ns period=13.000ns})
Destination	tap_01/delay_element_reg[12]/D (rising edge-triggered cell FDRE clocked by clk {rise@0.000ns fall@6.500ns period=13.000ns})
Path Group	clk
Path Type	Setup (Max at Slow Process Corner)
Requirement	13.000ns (clk rise@13.000ns - clk rise@0.000ns)
Data Path Delay	12.129ns (logic 8.843ns (72.910%) route 3.286ns (27.090%))
Logic Levels	14 (CARRY4=11 LUT1=1 LUT3=1 LUT4=1)
Clock Path Skew	-0.145ns
Clock Un...rtainty	0.035ns



Critical path contain:

1 multiplier, 2 adder (one is for rounding), 4 comparator (for clamping), 2 3-to-1 mux (for clamping)

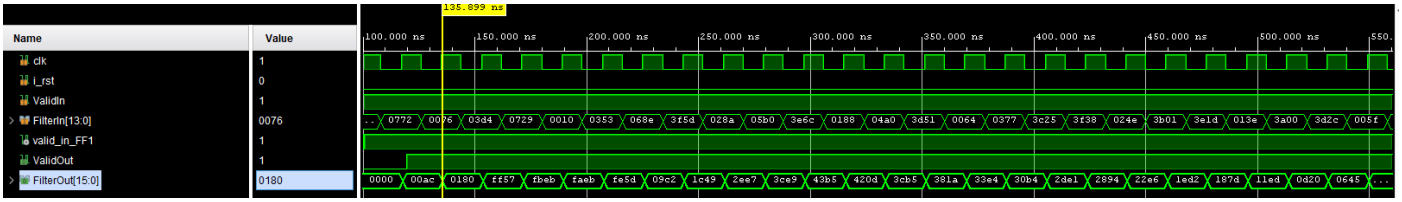
6.2 timing constraint:

My simulation clock period is 18.0. If set simulation clock period = 17.0, post-syn simulation fails.

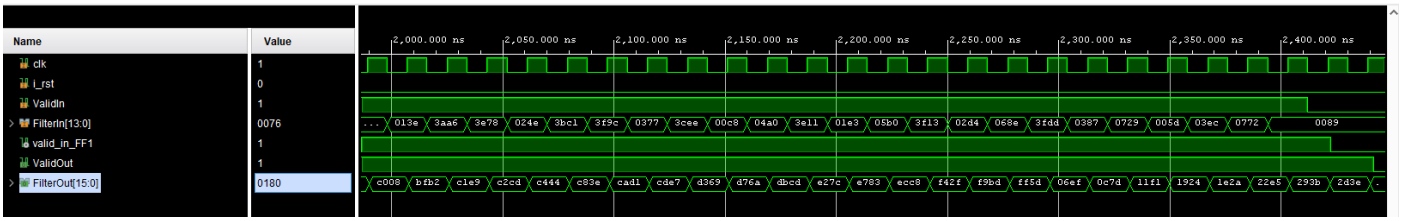
```
create_clock -period 13.000 -name clk -waveform {0.000 6.50} -add [get_ports clk]
set_input_delay -clock [get_clocks *] -add_delay 6.50 [get_ports -filter { NAME =~  "*" && DIRECTION == "IN" }]
set_output_delay -clock [get_clocks *] -add_delay 1.000 [get_ports -filter { NAME =~  "*" && DIRECTION == "OUT" }]

```

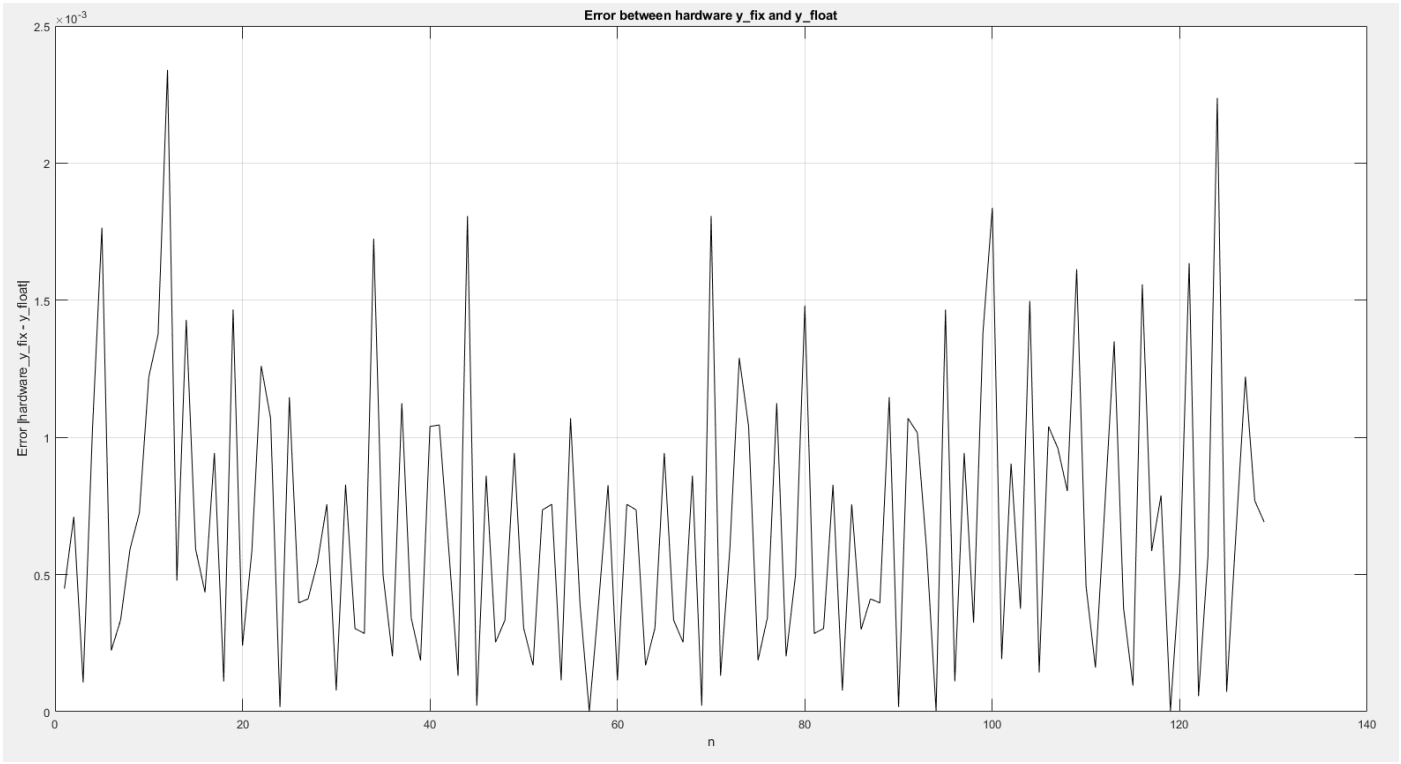
6.3 first 20 samples:



6.4 last 20 samples:



6.5 error between hardware output and matlab floating point output:



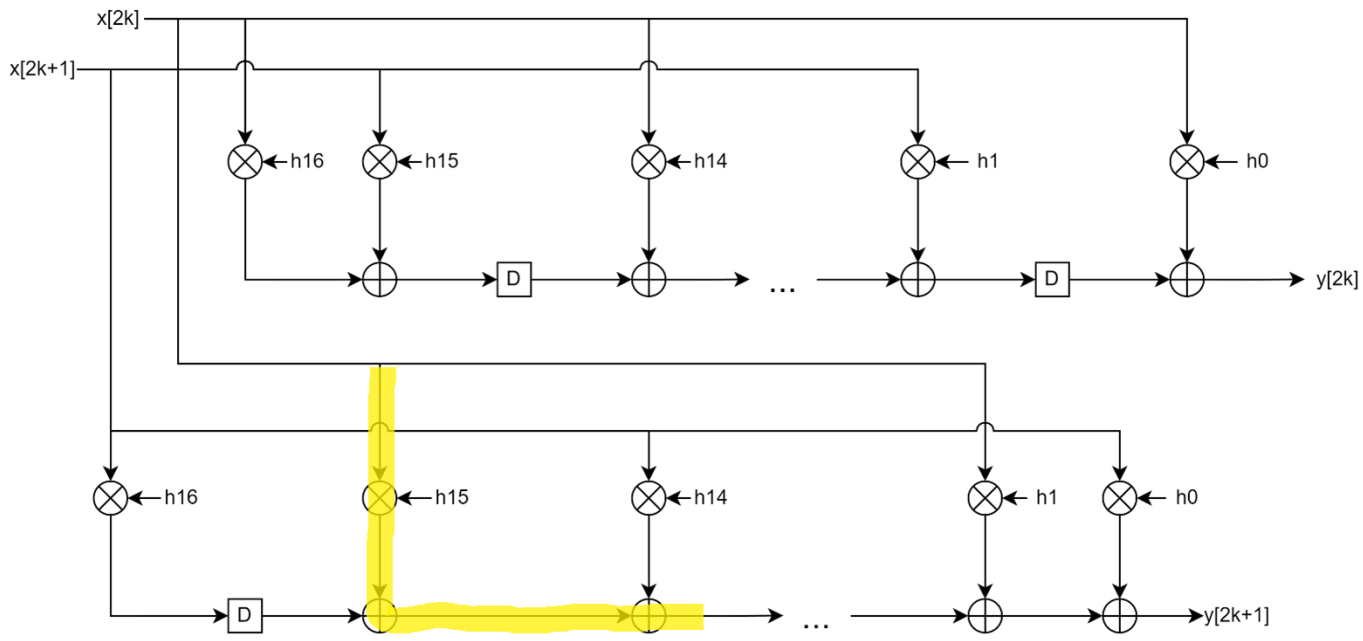
RMSE: 8.6363e-04

7. (Step 5) Show that $y[0]$ to y[128] can be generated within $Tout/2$ in the timing diagram. Explain the way that you use to accelerate the throughput and draw the block diagram and indicate your critical path. (15%). Show your synthesis report of max delay to verify your explanations. (5%)

7.1 block diagram:

I use a 2-level parallel technique to accelerate throughput, with the block diagram and critical path shown below.

The parallel FIR processes two inputs at once and generates two outputs simultaneously. Although the critical path is slightly longer compared to the original design (increasing from mul*1 + add*1 to mul*1 + add*2), it achieves approximately 2x throughput.



7.2 synthesis report:

The delay increases slightly because the critical path includes one more adder. However, since it generates two outputs at once, it achieves 2x throughput.

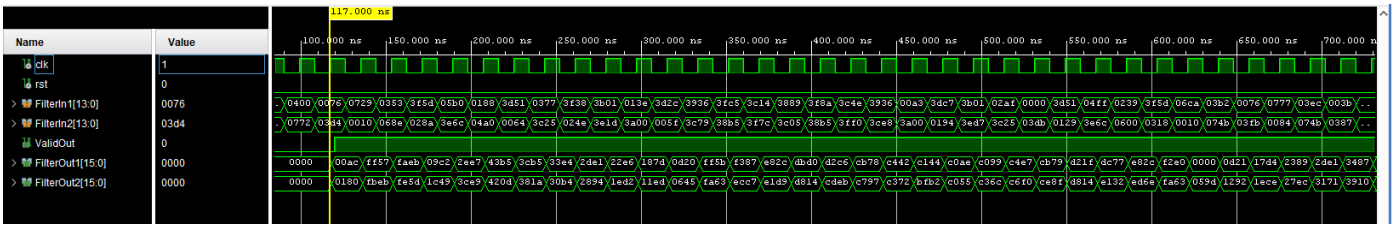
Summary

Name	Path 1
Slack	0.174ns
Source	tap_2k_07/prod_0/CLK (rising edge-triggered cell DSP48E1 clocked by clk {rise@0.000ns fall@7.000ns period=14.000ns})
Destination	output_FF2_reg[12]/D (rising edge-triggered cell FDRE clocked by clk {rise@0.000ns fall@7.000ns period=14.000ns})
Path Group	clk
Path Type	Setup (Max at Slow Process Corner)
Requirement	14.000ns (clk rise@14.000ns - clk rise@0.000ns)
Data Path Delay	13.675ns (logic 9.766ns (71.416%) route 3.909ns (28.584%))
Logic Levels	16 (CARRY4=12 LUT1=1 LUT3=1 LUT4=2)
Clock Path Skew	-0.145ns
Clock Un...rtainty	0.035ns

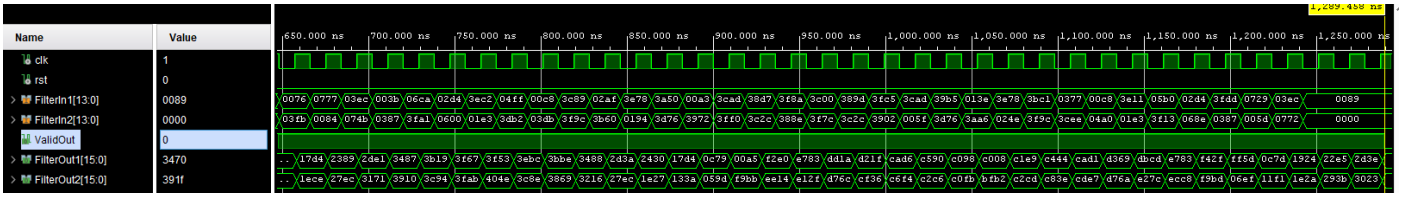
8. (Step 5) Show the post-SYNTHESIS simulation results for the first 20 samples and the last 20 samples and indicate why your throughput is doubled (10%). Show the errors of hardware outputs and Matlab floating-point results of transposed from FIR versus for 0 ≤ n ≤ 128 + $\boldsymbol{d}$ by figure. (10%) Show the area (resources) before and after speedup from the synthesis report. (10%)

8.1 first 20 samples:

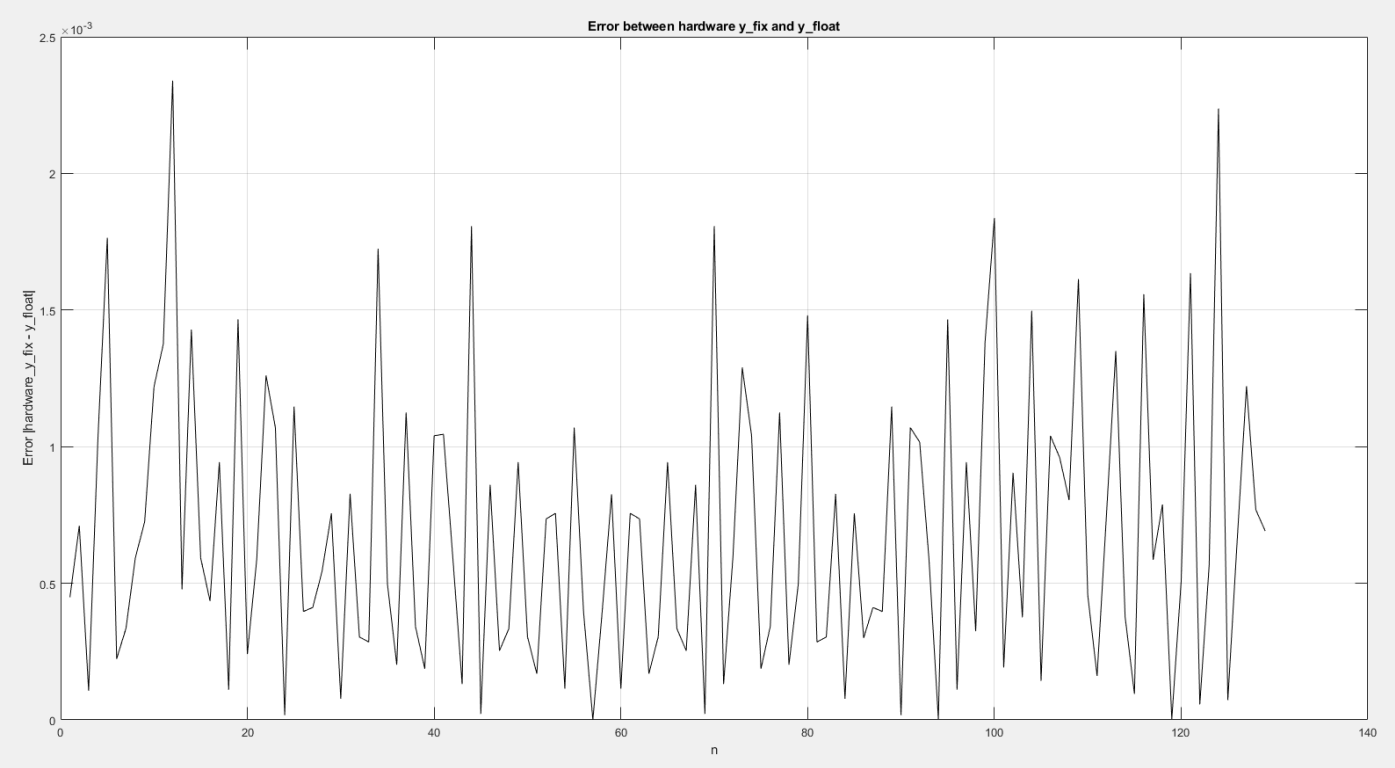
It generates two outputs at once, it achieves 2x throughput. (clk period is same as original one, which is 18ns)



8.2 last 20 samples:



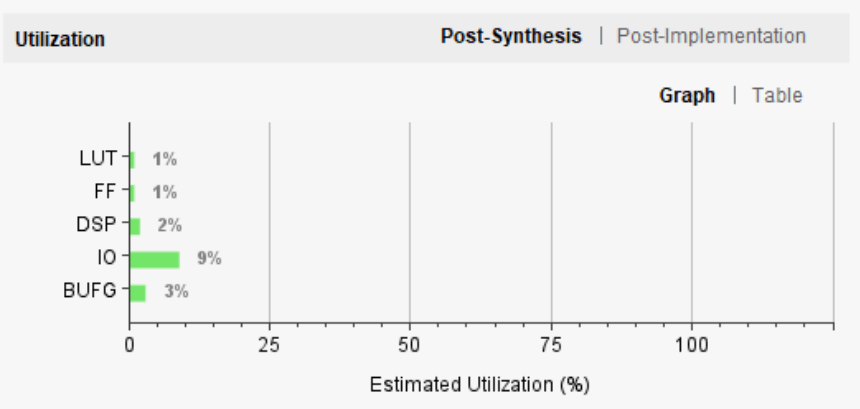
8.3 error between hardware output and matlab floating point output:



RMSE: 8.6363e-04

8.4 Show the area (resources) before and after speedup from the synthesis report.

Original FIR:



Parallel FIR: