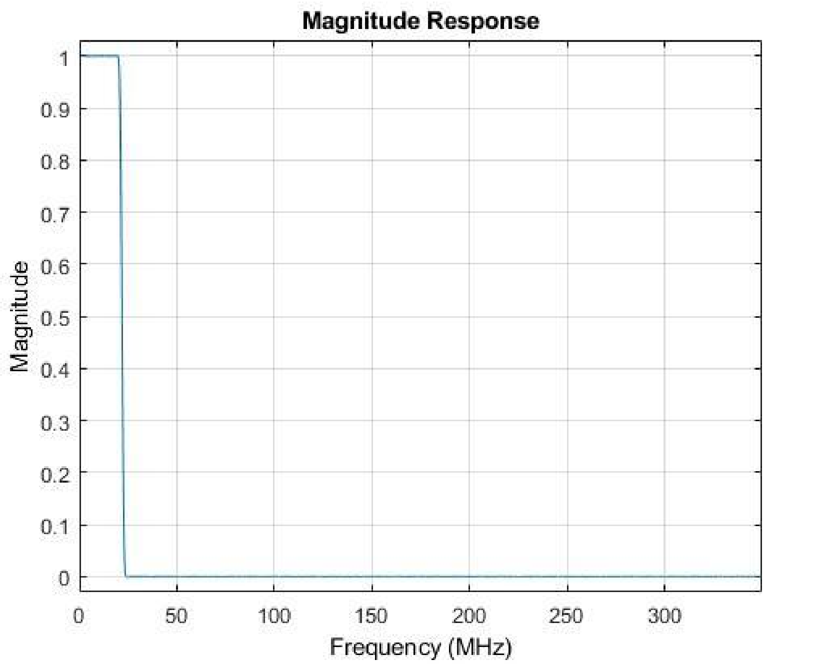
Filter Design Requirements:

DataRate = 20 MHz

SampleRate = 700 MHz

If we want to generate an ideal FIR filter with 0.01 dB peak-to-peak ripple, 80 dB stopband attenuation and 4MHz transition width, the filter needs to be a minimum 726th-order filter having 727 coefficients. This is huge to build in the circuit level / FPGA level in terms of resources.

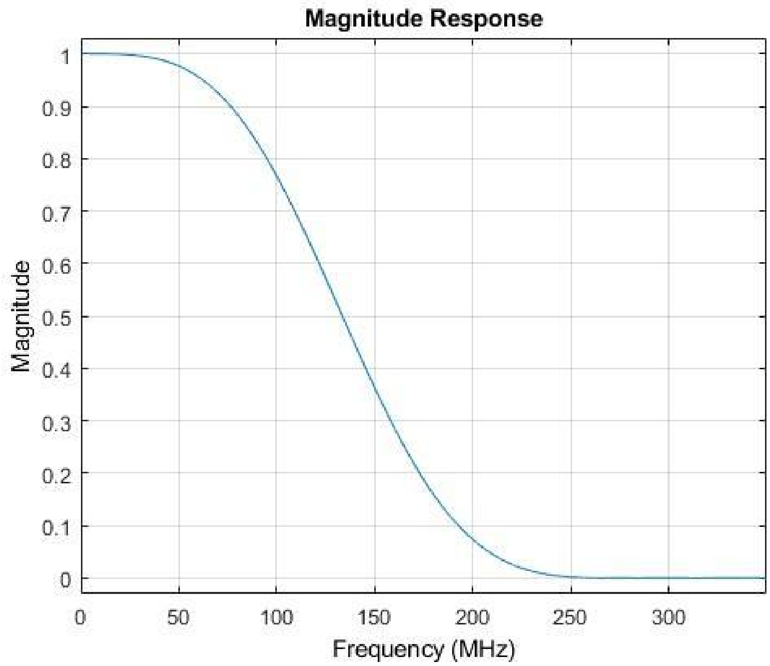


The code is attached with this report as "FIR_Filter_Design.m" and coefficients are attached as " coefficient_values_minorder.mat"

9th Order FIR Filter:

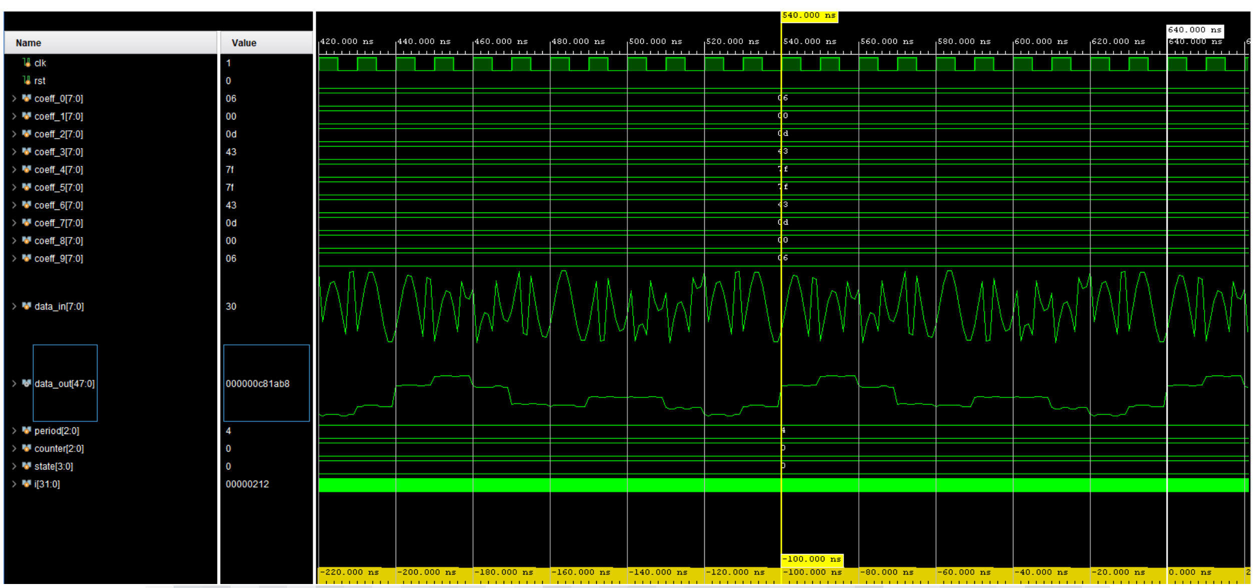
So, we have decided to go for lower Filter order. So, considering 9th order filter, we got the coefficients as

6 0 13 67 127 127 67 13 0 6



Though the frequency domain response is not as good as ideal filter, we tried to implement it on Verilog and simulate it using a test bench. In test bench, we actually mixed a 10 MHz sinusoidal signal with another 100 MHz sinusoidal signal and got a response which have the frequency of 10 MHz. This verifies that our filter is working in Verilog.

Here is the response that we got simulating our test bench with filter in Verilog,



Hrudaya got another set of 9th-order FIR Filter coefficients using her method. The coefficients are,

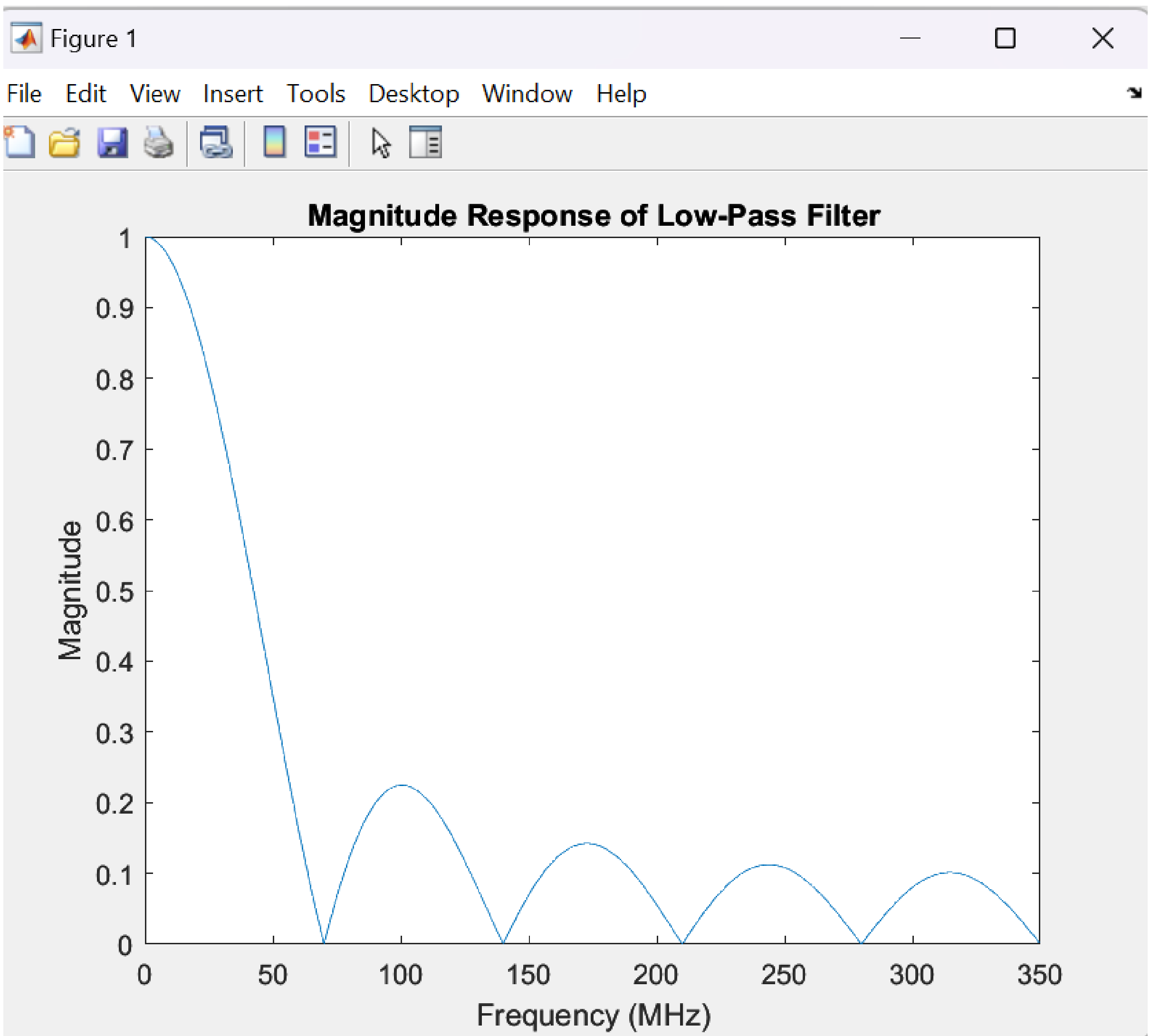
116 121 124 126 127 127 126 124 121 116

First, using the Kaiser window as per the required frequencies, 330 coefficients were obtained. The goal was to reduce these coefficients. The following two steps were followed. To decrease the coefficients, the values were rounded off to two decimals, resulting in 74 coefficients.

Instead of using 74 values, the intention was to use even fewer values for testing the output. Therefore, the precision was adjusted to 10. This adjustment yielded a low pass filter. The number of non-zero coefficient values was 10, all of which were above one.

The presence of zero coefficients indicates that the corresponding terms in the filter impulse response have no effect on the output signal. Omitting these zero coefficients will not change the overall characteristics of the filter.

The second step was to quantize the values.



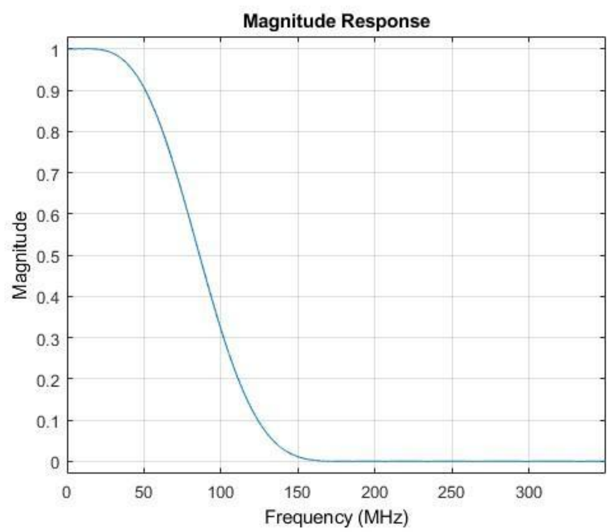
The magnitude response of the mentioned coefficients exhibits increased ripples in the stop band.

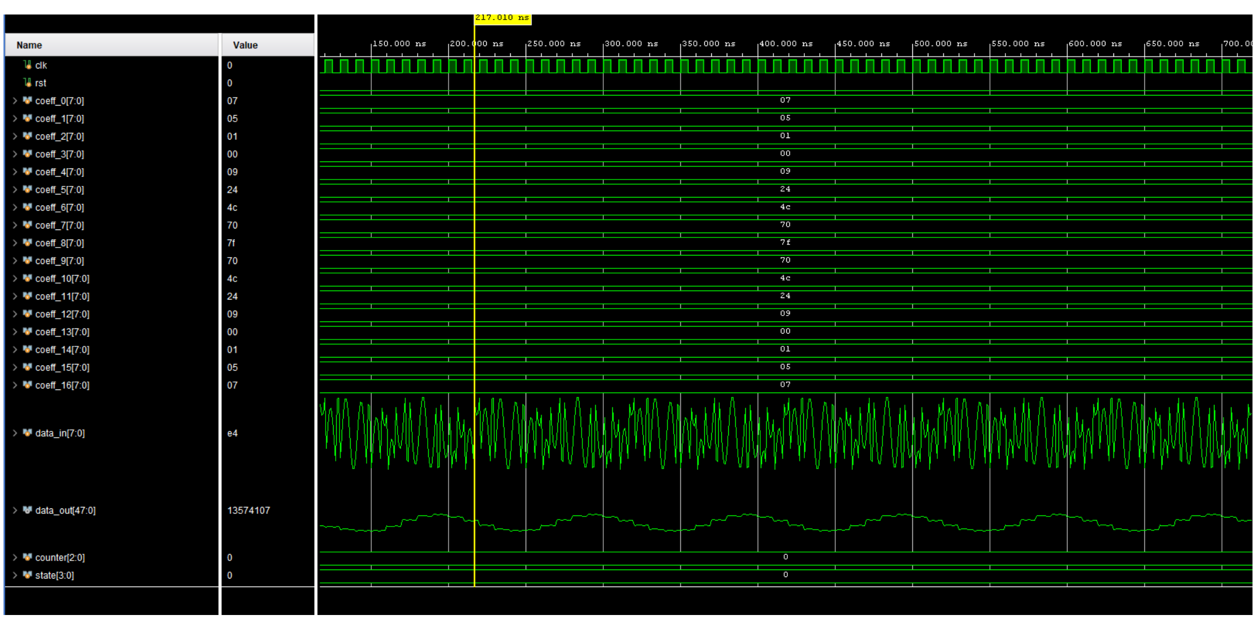
We applied the same input signal for these coefficients and got this response which also shows the same 10MHz frequency. The response is shown below,

16^{th} Order FIR Filter:

So, we decided to compare our result with a higher order filter and considered a 16^{th} order filter. The coefficient of this filter is,

7 5 1 0 9 36 76 112 127 112 76 36 9 0 1 5 7





The frequency domain response is better than the 9^{th} order filter. We checked the filter in Verilog for the same input as previous ones and got the time domain response which has periodicity of 10 MHz. The response is also better than the 9^{th} order filter response in terms of shape.

Analysis:

If the minimum order for filter cannot be met due to resource constraints, increasing the filter order as possible may improve the performance of the implemented filter. Another observation is that quantizing the lower order coefficient values gives better response than quantizing the non-zero values of higher order filter coefficients.

Resource Utilization:

Parameters	**9^{th} Order FIR Filter**	**16^{th} Order FIR Filter**
FF	72	128
DSP	19	33
IO	138	194
BUFG	1	1
Worst Negative Slack (WNS)	1.477 ns	1.957 ns
Power	0.23 W	0.266 W