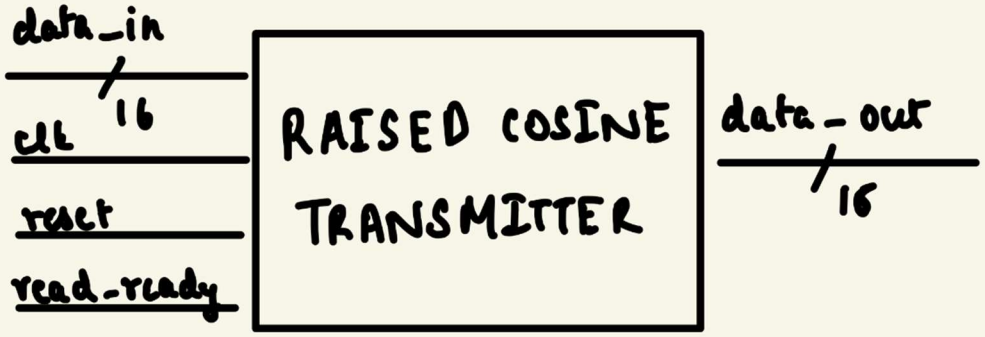
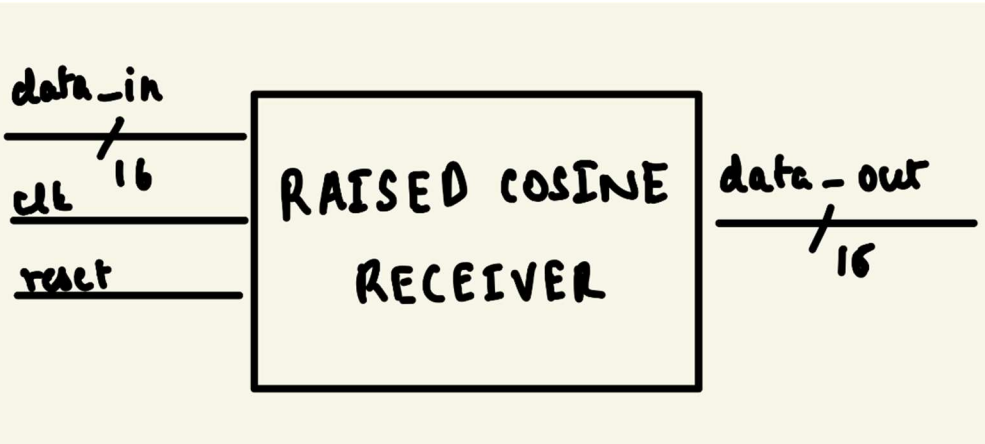
Raised Cosine Transmitter and Receiver

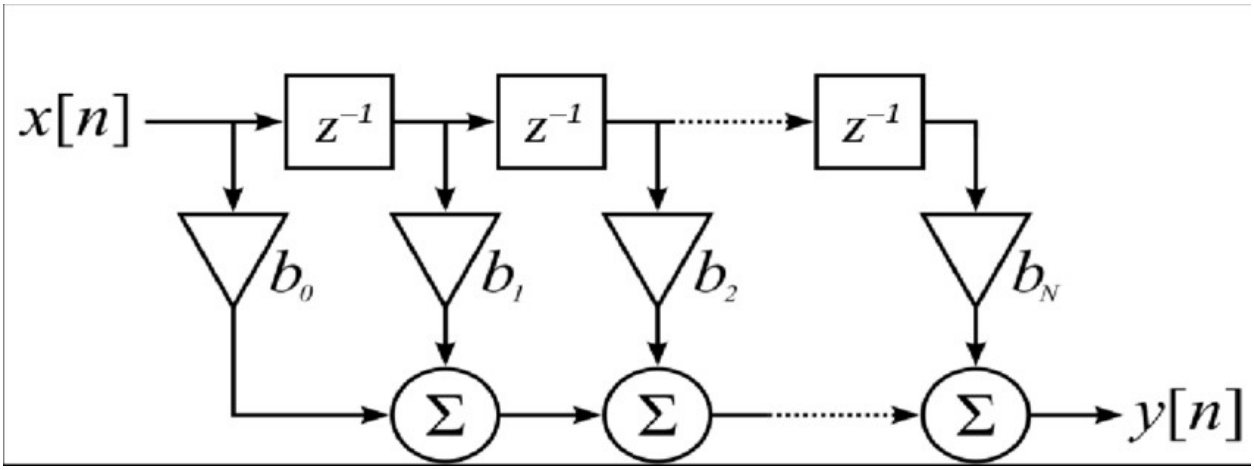
The raised cosine transmitter is used to transmit the symbols generated by the modulator into the channel through waveforms. Its receiver equivalent matches the waveform received by the channel into noisy symbols, which is fed into the demodulator. The input/output ports of the transmitter and receiver are shown below:





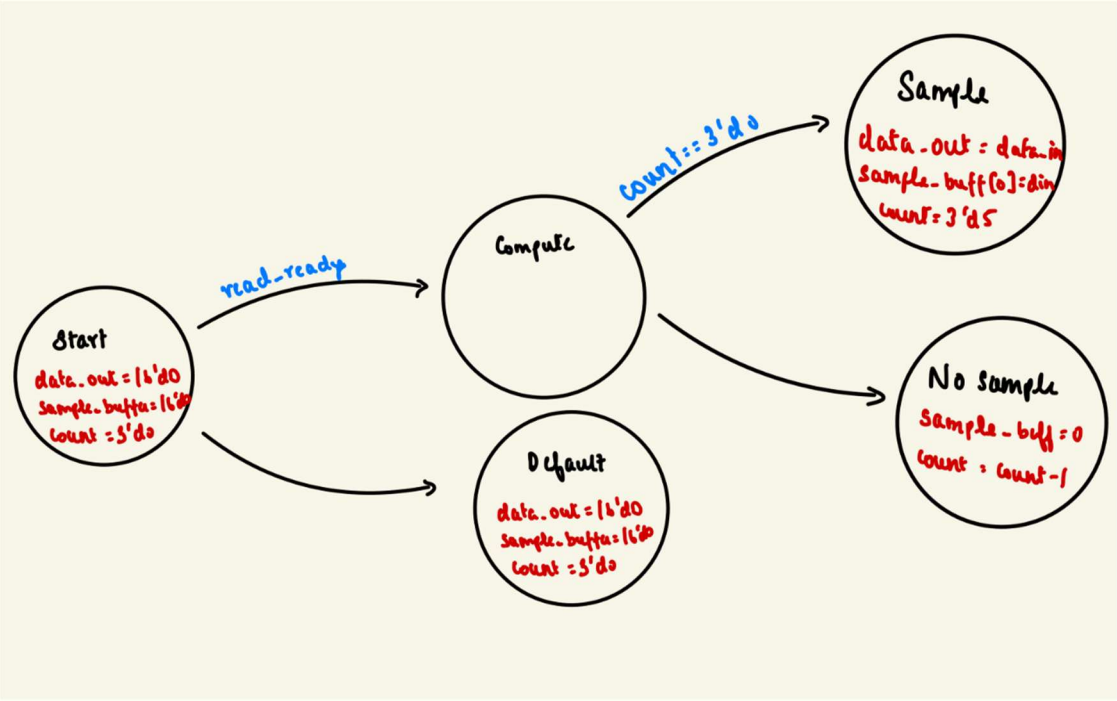
INPUT/OUTPUT PORT DESCRIPTION		
PORT	TRANSMITTER	RECEIVER
Inputs	- clk: Clock signal to synchronize the filter operations. - reset: Signal to reset the filter. - read_ready: Signal to indicate readiness to read input data. - data_in: 16-bit input data from the modulator.	- clk: Clock signal to synchronize the filter operations. - reset: Signal to reset the filter. - data_in: 16-bit input data from the channel.
Outputs	- data_out: 16-bit filtered output data to be transmitted to channel.	- data_out: 16-bit filtered output data.

The general design of both these filters were inspired by the fundamental design of the Finite Impulse Response, which has the following block diagram:



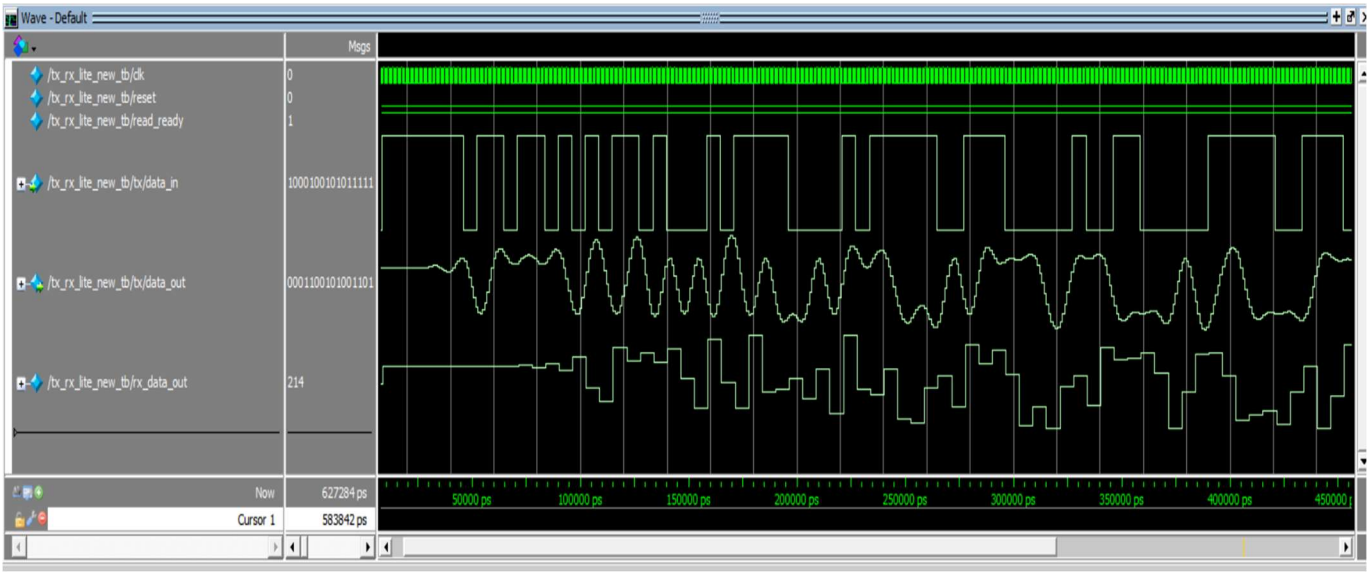
In this diagram, b0, b1, b2...., bn, are replaced with the coefficients of the raised cosine filter we get from our Simulink simulation. These coefficients values are multiplied by 2^(13) and rounded off to the nearest integer, and was performed with the help of excel. The reason for multiplying by 2^(13) is to match the quantization requirements of the channel, so we treat our coefficients as 16 bit signed numbers with 13 fractional bits. We perform the same for the receiver except we scale down our coefficients by our decimation factor (which is 6 in our case).

Apart from the FIR filter implementation, the transmitter follows a ready enable counter to maintain data integrity when receiving data from qpsk. This counter can be better understood by the bubble diagram:



Lastly, the output from the receiver is followed by a downsampler that effectively reduces the number of samples in the receiver waveform. Its significance is to try and recover the pulse like structure of the QPSK symbols by downsampling the received wave by the same factor it was oversampled with in the transmitter.

The test strategy taken towards the transmitter and receiver involves providing randomized QPSK values to the transmitter filter at 160kHz, simulating our real model. The expected behavior is that the transmitter will process this data correctly and the receiver will reconstruct the original signal after filtering and downsampling. The transmitter should output a filtered version of the input data and the receiver should output a reconstructed version of the transmitted data. Furthermore, we want to verify that that the output matches the expected filtered and reconstructed data values. On coding up the testbench and running the simulation of our model, we see the following waveforms:



The first analog wave represent the QPSK values sent into the transmitter. The second analogue wave represents the output of the transmitter and the last wave represents the output of the receiver when the input of the transmitter is fed into it. As we can see the transmitter effectively captures the pulses being sent in by the modulator and the receiver outputs a filtered and downsampled version of the transmitted wave, indicating that we have met our requirements for the raised cosine transmitter and receiver.