D1 need to focus on coeffs for the 2 transmitters. THey're both the same btw

need a stopband Attenuation of 40 dB lower than DC; MER is a relative requirement, do as well as you can

L is Nsps; the FR of srrc and rc only differs from transition band;

To calculate theoretical magnitude responses, we need to do it in a piecewise operation like how their FR is defined

ideal SRRC filter has an infinite length IR

generating an FIR from an IRR is known as windowing (like in 461)

h_finite[n] = h_infinite[n]*w[n]

to build a real filter, we always perform truncation. Time frequency duality, which tell us the convolution in one domain corresponds to multiplication in the other domain.

Time domain we bring a input signal into the filter and the output is the convolution of the input and filter's impulse response

In frequency domain, the FR is multiple by spectrum of input signal to get the touptu

in the time domain we perform windowing operation which is just point by point multiplication; this corresponds to convolution in frequency domain

In the frequency domain we convolve ideal FR with the window's FR

what is Frequecny response of our window function and how does it influence overall FR of the finite filter? some of the windows are rect, kaiser, hamming, etc... and these all have distinct features that affect the magnitude response

windwoing is multipying 2 sequences in the time domain.

MER Jan 26

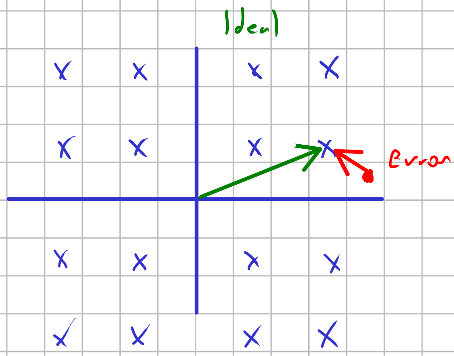
How can we use the cascade of 2 filters to analyze their MER? Consider the equivalent system

Equivalent system

4-Ask symbols $\rightarrow \uparrow 4 \rightarrow h_{ps}[n] \rightarrow h_{MF}[n] \rightarrow \downarrow 4 \rightarrow$ Recv Symbols

SRRC SRRC

$$h_{ps} * h_{MF} = RC$$



Take 4-ASK symbols which correspond to 1 dimension of our 16 QAM system. UPsample by 4 (inject 3 zeros). h[n] is IMPULSE RESPONSE. Down sample block comes out at symbol rate

in 16-QAM system, we expect a perfectly symmetrically 16 point constellation and the received symbols align with one of the 16 points. IRL this is unlikely tho. MER quantifies the perforamce of our system based on our output

Lets define 2 vectors: an ideal vector and error vector; the MER is the ratio of the average signal vector magnitude. MER tends to differ from symbol to symbol so we need to take an average of these signals.

Sources of error: AWGN (but not consider for D1), quantization noise (from implementing filters and their coeffs BUT if we use large enough signal formats this should go away), ISI occurs when length of IR of filters and cascade is longer than the spacing between symbols.

From the above upsampler, we see the spacing is 4 samples between each symbol where each symbol is spaced out by 3 zeros. If the IR of filters is longer than 4 samples (which they are) the response of this system from 1 impulse to the next symbol will not have died out when the next symbol is taken in. Thus theres a potential overlap in these response which can contribute to final receive symbol in our 16 QAMconstellation

We can controll the ISI using the Nyquist Criteon for zero ISI: If we have periodic zero crossings in the combined response in the combined filter (cascade of the 2 SRRC filters) AND if the period of the zero crossings is equal to the period of the spacing between symbols; these 2 will give us no ISI

If the periodic zero crossings match with the period of the incoming symbols, then it occur where the impulses are centered at each zero crossing

Raised cosine has these zero corssings which avoid ISI

BUT since we truncate the filters, their combined response should not be a true RC filter, could lead to ISI.

MER example

hps & hmf are both 4 samples long -> heq[n] is (10-1) = 9 samples long

assume heq[n] = [h0, h1, h2, ..., h0]

input is x[m], upsampler output is x_u[n] (m symbol rate, n sampling rate), output of hmf is xf[n] and system output is xd[n]

complete the following table:

xf[n] is the convolution of xu[n] and heq[n]

upsampler inserts n-1 zeros between each symbol; downsampler only accepts k*n samples (k is int)

From thue table, we see we've constructed a simpler system

4-Ask Symbols

$x[m] \longrightarrow h_{eq,ds}[m] = [h_1, h_5, h_9] \rightarrow x_{ds}[m]$

n	m	x[m]	x_u[n]	x_f[n] (xu[n] * heq[n])	x_ds[m]
1	1	X[1]	X[1]	x[1]h1	x[1]h1
2			0	x[1]h2	
3			0	x[1]h3	
4			0	x[1]h4	
5	2	X[2]	X[2]	x[2]h1+x[1]h5	x[2]h1+x[1]h5
6			0	x[2]h2+x[1]h6	
7			0	x[2]h3+x[1]h7	
8			0	x[2]h4+x[1]h8	
9	3	X[3]	X[3]	x[3]h1+x[2]h5+x[1]h9	x[3]h1+x[2]h5+x[1]h9
10			0	x[3]h2+x[2]h6	
11			0	x[3]h3+x[2]h7	
12			0	x[3]h4+x[2]h8	
13	4	X[4]	X[4]	x[4]h1+x[3]h5+x[2]h9	x[4]h1+x[3]h5+x[2]h9
					x[m]h1 + x[m-1]h5+x[m-2]h9

In order to have no ISI, we need to have periodic zero crossings in the filter such that the downsample version of the filteris going to hav 1 sample that is a peak and the rest are zeros

we see our equivalent systm, we only need h1, h5, h9

For a RC filter, it has a peak in the center then drops off

The center coeff in this case is h5, the largest peak is whatever is multipled by h5; we have 2 extra terms in the output: xmh1an xm-2h9

x[m-1]h5 is the ideal expected term and the extra terms are ISI terms. If our system is a perfect RC system, it would have periodic zero crossings such that h1 would be 0 and h9 would be 0 to remove the ISI terms and we only want the h5 term.

If we don't have the right coeffs or long enough response to duplicate the response, we may have contributes to the ISI based on every 4th term not being zero due to our upsampler and downsampler rate

ISI is generated when every 4th coeff isnt 0; lets see the MER for the system

compute MER under assumption of no AWGN or quantization noise; solely based on ISI

$$\underbrace{x[m]h_1}_{\text{ISI}} + \underbrace{x[m-1]h_5}_{\text{ideal expected}} + \underbrace{x[m-2]h_9}_{\text{ISI}}$$

$$MER = \frac{\text{avg}(\text{ideal mag}^2)}{\text{avg}(\text{error mag}^2)}.$$

avg of large number of received symbols

$$= \frac{\text{avg}\{x[m-1]^2 h_5^2\}}{\text{avg}\{x[m]^2 h_1^2 + \underbrace{2x[m]x[m-2]h_1h_9}_{\text{Avg of this is Zero}} + x[m-2]^2 h_9^2\}}$$

$$= \frac{E_s h_5^2}{E_s(h_1^2 + h_9^2)}$$

since theres + and - values for x[m] and x[m-2] (3a, a, -a, -3a)

Es = average signal energy

if you take 2 symbols and multiple them together, we could end up with this product being positive or negative; after injecting a large sequence of this, we would expect that these terms would negate each other

$$= \frac{h_5^2}{h_1^2 + h_9^2} = MER$$

$$h_{eq}[n] = [\underline{h_1} \quad h_2 \quad h_3 \quad h_4 \quad h_5 \quad h_6 \quad h_7 \quad h_8 \quad \underline{h_9}]$$

if the equilvalent filter meets nyquest zero condition, every 4th coeff besides h5 should be zero thus we get our h5^2/0 = infinity and means we have no error

ex: h5 = 1, h1 = 0.1, h9 = 0.1

MER = 1^2/(.1^2+.1^2) = 1/.02 = 50

in most cases we want the MER in dB since this is how we work with comms systems

MER dB = 10log10(50) = ~17 dB

we started out with a PS filter and match filter and convolve them into heq, based on this system with upsampling and downsampling on the ends;

the output will only depend on every 4th coeff, where we compare the magnitudes of each of these taps and can compute the MER for the filter

can play with beta, upsampling/downsampling rate (not advise to change O/W affects MER), truncation (IIR to FIR) to design a filter to meet your requirements

Best filter to use is based on MER

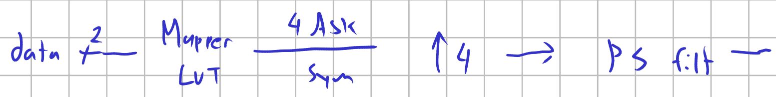
Multiplier-Free Pulse shaping filter (Jan 28)

FPGAS have a limited number of embedded hardwork circuits such as multipliers (want to conserve as much as possible)

Cyclone 4 FPGA can peform 18 by 18 multiplication

how to implement filter using no/few multipliers?

key point: whatever structure we use MUST perform calculation shown in difference equation!

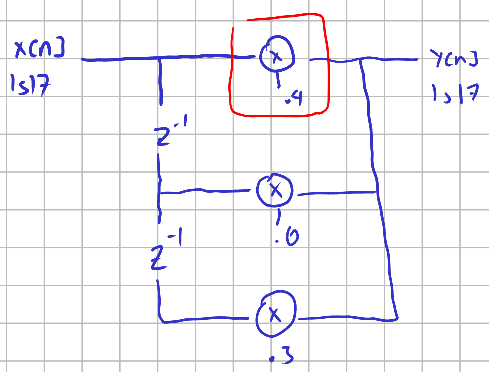


random data comes in, 2 bits comes into a mapper which is LUT; that LUT outputs sequence of 4 ASK symbols; symbols passed into upsampler then into PS filter which outputs a signal

we want to replicate this PS filter without any multipliers

ex: consider a 3-tap filter: h[n] = [.4 .6 .3]

difference equation: y[n] = .4x[n] + .6x[n-1] + .3x[n-2]
simplest implentation: Direct form



If this filter is our PS filter, its a wasteful structure

remember the multipler in the FPGA can compute a 36 bit output

mult uses addition and bitshifts

18 36 adders could be used

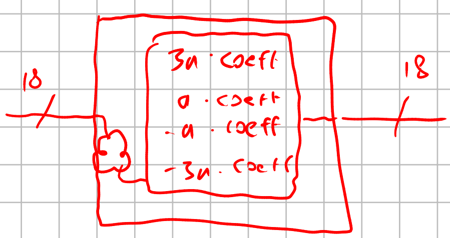
multipler is designed regardless of its input where it has 2^18 possible values for one input

For the red block, we see that one of our inputs is a constant value of 0.4 BUT the other operand has 4 possible values since the input of the PS filter comes out from a mapper. Mapper picks one out of four possible points in the constellation

Output from mapper is either -3a, -a, a, 3a

we're using a powerful circuit just to perform a multiplication with 4 possible outputs: 1.2a, .4a, -.4a, -1.2a

since we know theres only 4 possible values, we can replace our multiplier with a LUT



Before we do this, we need to consider a few things:

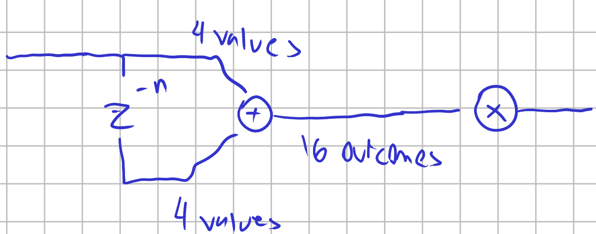
what value should a be in the ASK mapper?
to verify our design works, we need to verify its impulse response; note that this LUT can't accept a 0 output when doing an IR test **THIS IS WHAT WILL BE IN THE LAB EXAM**

consider if we did use this LUT idea; we have 2 LUT tables cascaded which is kinda inefficient.

Each of these delay registers are storing a 18 bit value BUT we know that theres only 4 possible inputs so we can conserve resources here as well

There might be some way to redesign this system

consider the symmetry filter we used:



This implementation would require a more complicated LUT