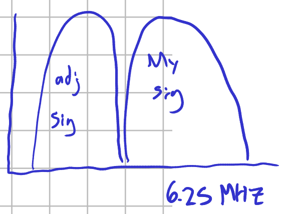


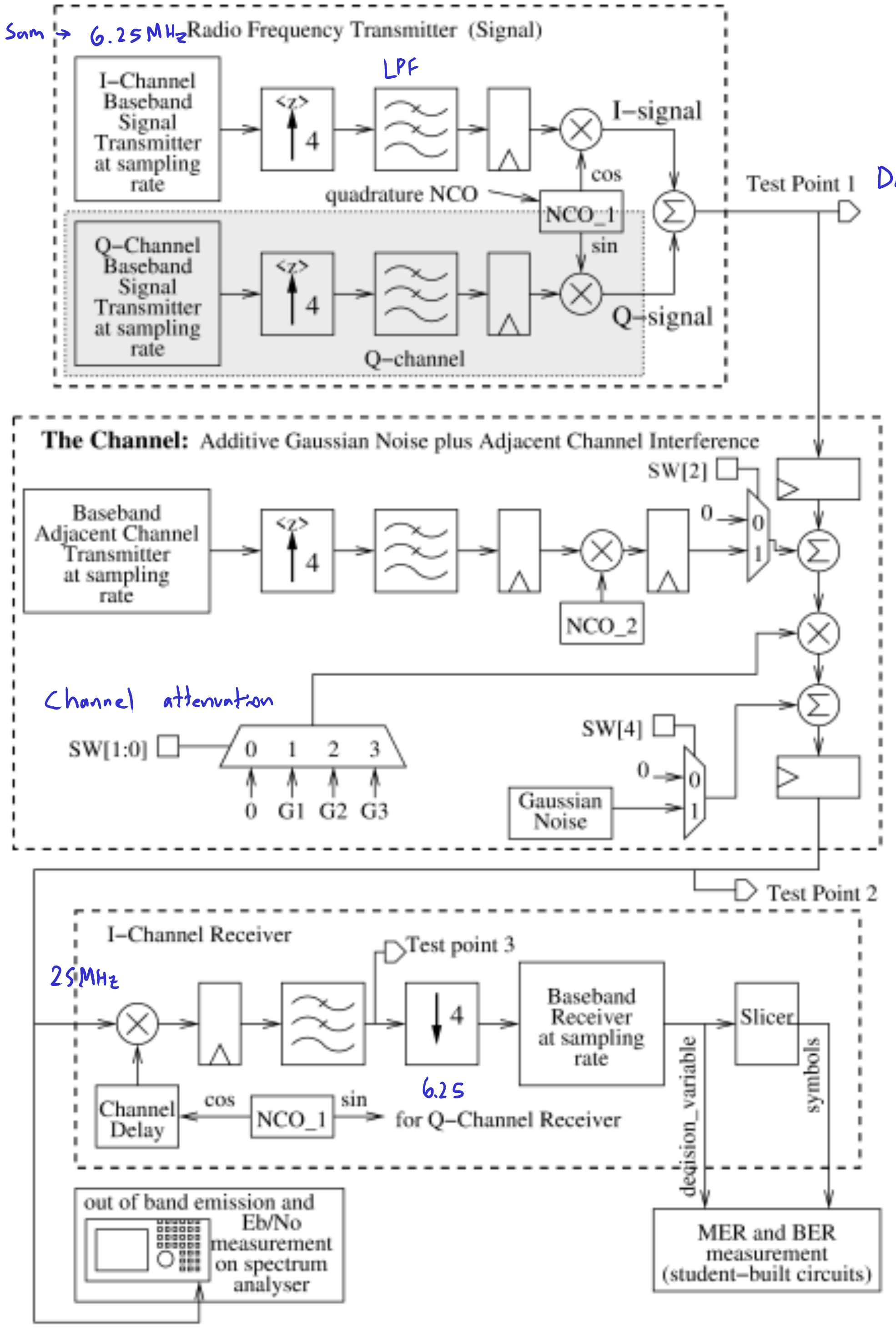


Figure 2.4: Block diagram showing the up/down sampling and up/down conversion.

I&Q Channel blocks @ input encompass the data gen: LFSR, mapper, & pulse shaping filter
LPF to filter spec copies. Fc is carrier freq

adjacebt channel to model real transmission. filt should filter and limit dmg of rcv sig

upsampler convs samplin rate to DAC rate
Down sampler convs DAC rate to sampl rate
upconverter mixes signal to Fc
downconvert mixes signal BY Fc to bring to baseband

Gaussin noise blk has fixed power

BERisused wifMER. verify BER to theo

i channel uses cos, Q uses sin

filters should still meet OOB requirements from d3

p.110 has PSD at bot

3) shouldmeet the requirements listed

BER meas on f.2.5.

Key design requirments:

-Fc = 6.25 MHz
-PPS & MF from D3 should have an MER of 39 dB with conjuction w/samplers & converters
-Need to meet all OOB requirements in D3

-If using Testpoint 2, you need to turn off Baseband channel & gauss noise for OOB req

-Less than 14 Multipliers total which refers to mults in samplers & converters for BOTH I & Q channel

-Need to choose gains in channel model which are to be selected and should give us specific SNR @ test point 2: 7.88, 10.52, and 12.2 dB from G1, G2, and G3 respectively

-BER from adjusting G1, G2, and G3 should be 10^-2, 10^-3, & 10^-4 respectively and should have AWGN enabled

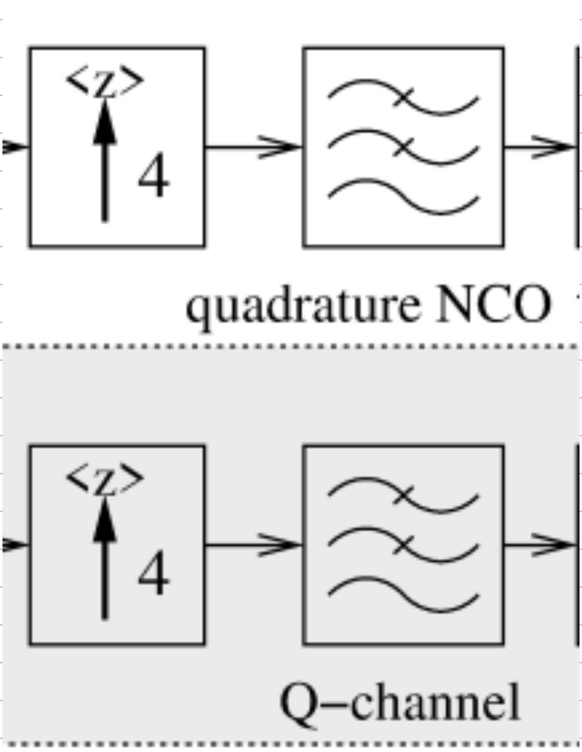
-when AWGN is disabled, BER=~0; not gonna be 0 since theres noise BUT should be close

Progress thru deliverable as listed in Intro pt2:

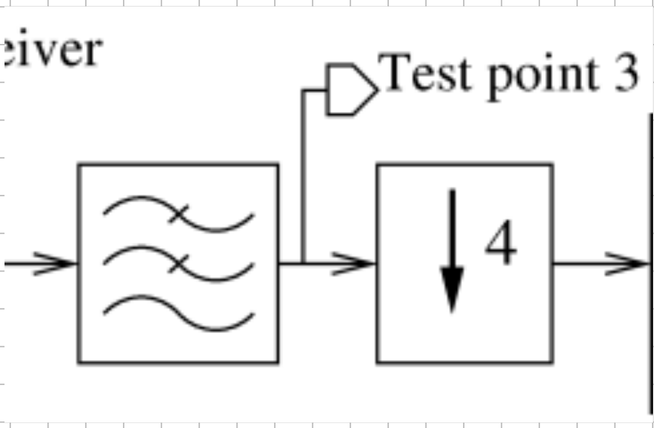
1) Upsampler & downsampler
2) Converters
3) Channel model circuit
4) BER meas circuit

Sampling filter design Pt.1

upsamp



downsamp



Upsampling + Downsampling

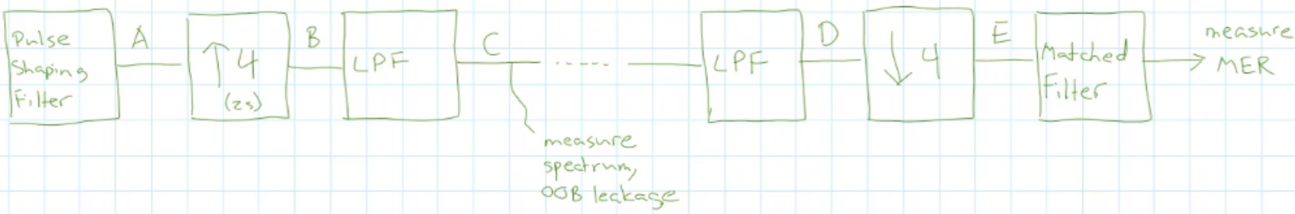
requirements : MER ≥ 39 dB (1 dB relaxation wrt D3)

: meet OOB requirements from D3

: minimize cost (≤ 14 mults total)

Key question : How does the upsampling + downsampling impact the MER + OOB leakage?

- Consider a simplified system (omit channel model + up/downconversion) :



LPF to remove aliasing; this assumes that PS & GS give 40 dB MER

what does the spectrum @ A,B,C,D, and E look like?

go next; note its important to do this on ur own

Spectrum at A :

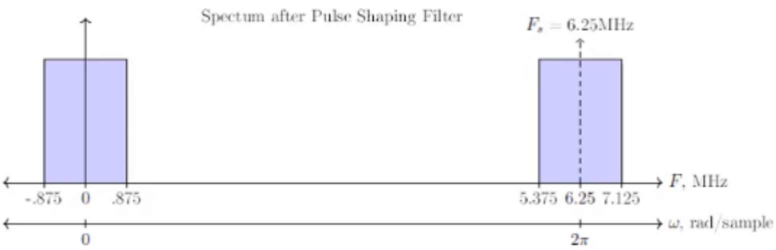
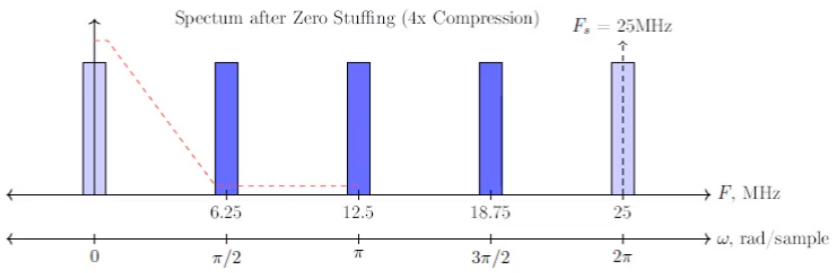


image of baseband signal @ Fs*k (Fs=6.25 MHz)

Spectrum at B:

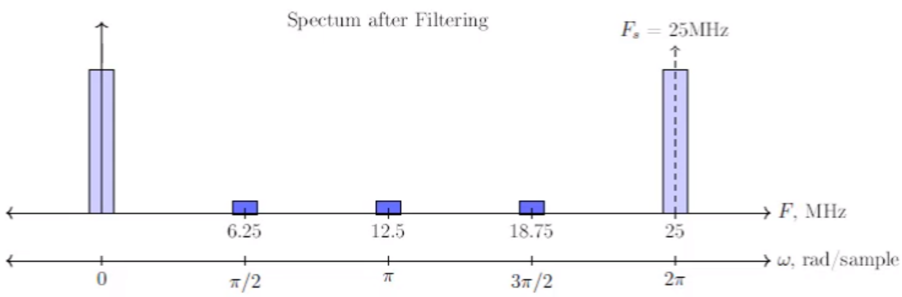


upsampling results in the spectrum being compressed by 4 & Fs is scaled by 4

DAC rolloff would scale the images in blue; these blue images violate our protocol,
we need to apply a filter to remove this images

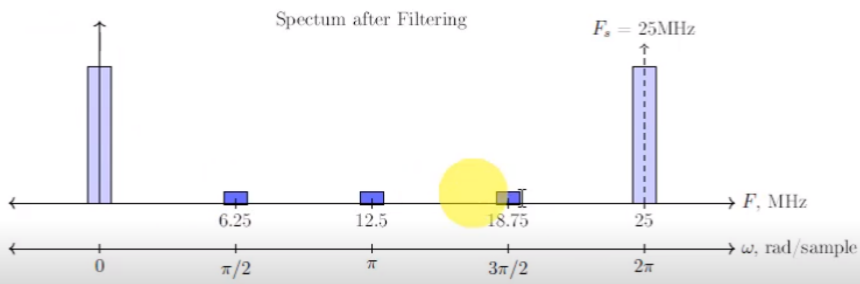
since spectrum is periodic about 2pi, need to ensure that images within 2pi are filtered as desired so that this desired response is replicated thruout

Spectrum at C:

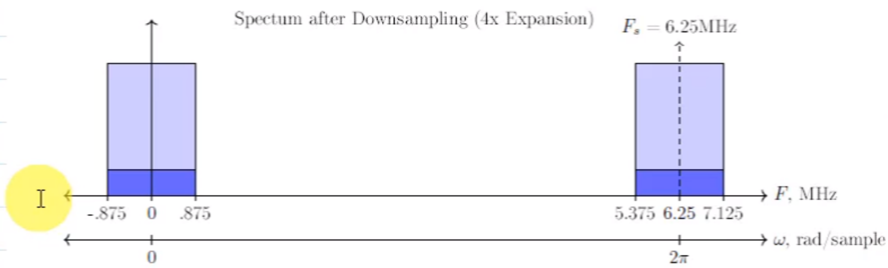


images are attenuate as much as possible by LPF

Spectrum at C:, D:



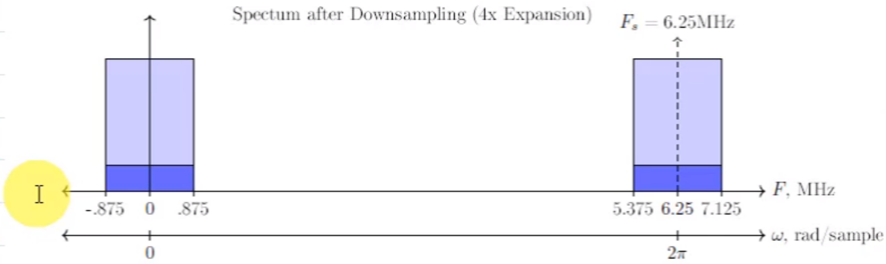
Spectrum at E:



w/Downsampling, we stretch our frequency by 4 where the images @ baseband align with our original Fs BUT notice that some of the images we tried to suppress now overlap with our desired spectrum from downsampling (images between 0 & 2pi) which acts as noise in our baseband channel

to have good performance, we need to check the power of these images

Spectrum at E:



Need to know what is the level diff between our signal of interest and new noise.

If the power in the images is too high, it will decrement our MER so we want to lower their power

the dark blue spectra @ baseband includes all 3 images between 0 & 2pi overlapping with our desired baseband signal

Upconversion & Downconversion

To design the filter:
we need to consider the coefficents based on:

-passband corner frequency (dont want our filter to roll off to soon to impact our channel. We want our corner frequency to be greater than our channels upper edge which is 0.875 MHz

-stopband corner frequency is related to lower edge of the next image centered @ 6.25 MHz, due to the nature of the mage, its width is 6.25 +/-0.875 MHz in analog. You can convert to digital domain by taking that frequency and dividng by Fs & mult by 2pi.

Thus our Fstop must be lower than the 1st images, lower edge

-Passband ripple adds distortion to our signal and causes our signal in the channel to not be flat need to make sure the ripple doesnt degrade MER by 1 dB

-Stopband attenuation: want to balance power in images with power in baseband to help us meet the 39 dB MER requirement

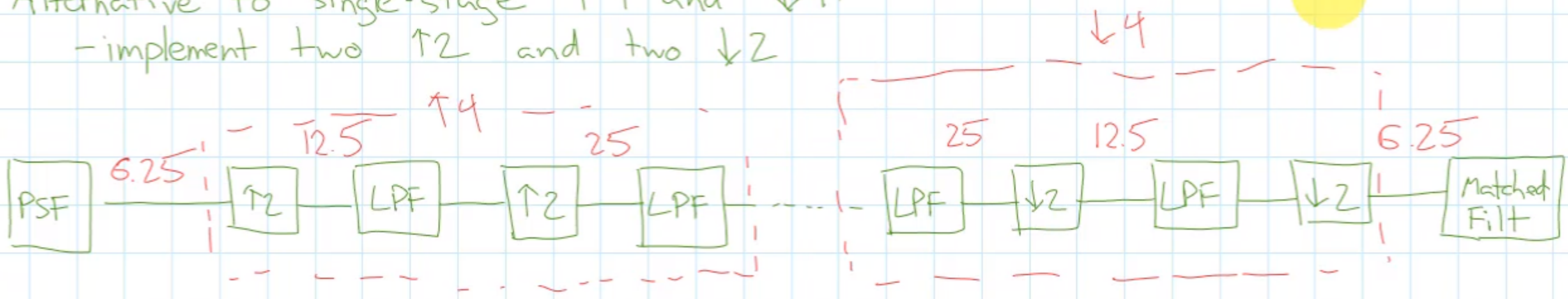
need 2 upsamplers & 2 downsamplers with a total of 14 Mults. Consider zero mult, symmetry, and polyphase implementation.

Remember for polyphase, we remove out coeffs due to the sampling

BUT consider this: (see next pg)

Alternative to single-stage ↑4 and ↓4:

- implement two ↑2 and two ↓2



- key advantage: filters can be half-band filters (every 2nd coeff is 0)

- looks more complicated than ↑4 and ↓4, but <u>may</u> be more economical

Instead consider two samplers by a factor of 2, where we bring the sam rate to 12.5 MHz, pass it thru an LPF and then sample again by 2 and pass it thru another LPF

notice that we have a cascade of 2 samplers and 2 LPF for an upsampler/downsampler by 4

if we design this circuit like this, we can take advangate of half-bands property where the 2nd coeff is 0

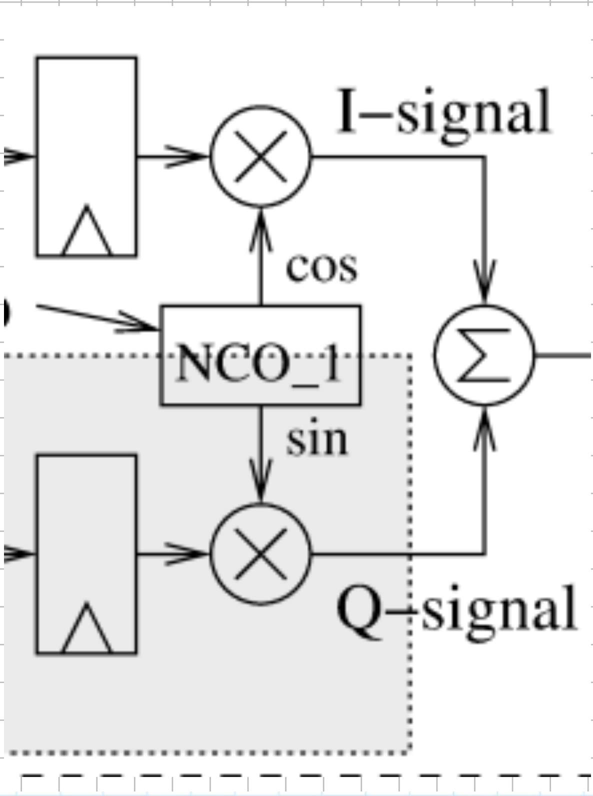
COmplicated BUT more cost efficient. For computing coefficients, consider filterDesigner in Matlab.

Top tap can also show the coefficnets, phase response, etc

always symmetry in halfband filter between passband & stopband response. Passband corner frequency determines stopband corner frequency

firsthalfband operates Fpass @ 12.5; notice in example the order is 6 and every 2nd coeff is 0 which is nice for zero isi criterion and is odd symmetric

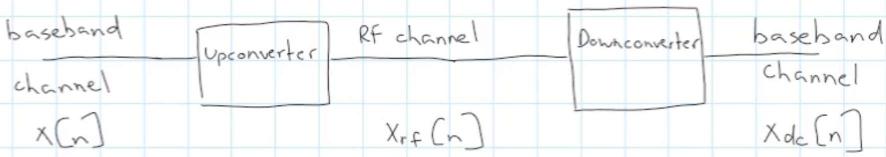
To verify upconversion process, you should see your signal @ 6.25 MHz & a slightly smaller lobe @ 12.5 MHz



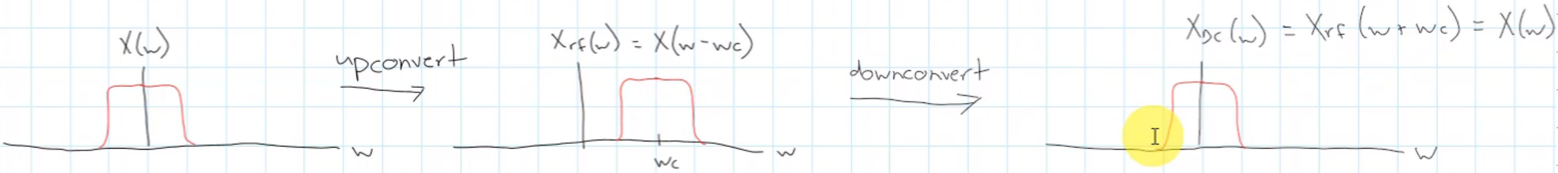
Upconversion block

shifts baseband channel to some chosen Fc (and vice-versa)

Purpose: shift baseband channel to some chosen center frequency (and vice-versa)



Ideally,



Basic Fourier Transform Theory:

$$X(\omega) \longleftrightarrow x[n]$$

Modulation property $X(\omega - \omega_c) \longleftrightarrow x[n]\, e^{j\omega_c n}$

shift spectrum right by ω_c $\longleftrightarrow$ multiply by complex sinusoid w/ frequency ω_c

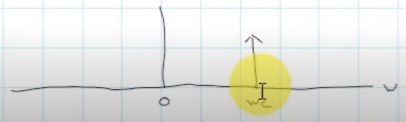
If we take a signal in time domain and apply a FT, we can observe its spectrum x(omega)

to shift the spectrum, we need to multiply by the time domain sequence, e^j*omega[n]

- define $c[n] = e^{j w_c n}$ $\rightarrow$ $x[n]\,e^{j w_c n} = x[n]\,c[n]$ ... mult in time domain $\rightarrow$ convolve in freq domain

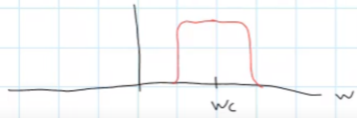
$X(\omega)$

$C(\omega) = \delta(\omega - \omega_c)$



$\longrightarrow$

$X_{rf}(\omega) = X(\omega) * C(\omega) = X(\omega - \omega_c)$



Let c[n] be the carrier signal. We can see the spectrum we output is related to the convolution of the 2 signals to the left

When u select a finite amount of bits, you'll get an approximation of the sine wave which gives you an approx sine sig to mix your input by

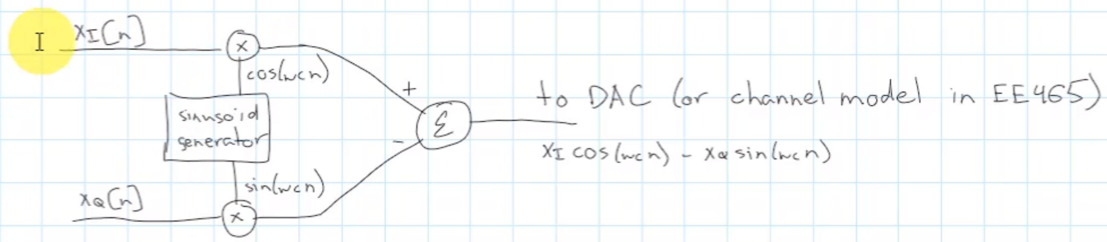
so by mixing with an imperfect sine, there will be the spectrum @ wc and some noise as a result of the convolution with the imperfect sine

QAM system: $x[n]$ is complex $\rightarrow$ $x[n] = x_I[n] + j x_Q[n]$ (independent transmitter paths)

$\rightarrow$ Upconversion: $x_{rf}[n] = x[n]\,\underline{e^{j w_c n}}$ $\rightsquigarrow$ Euler's identity $e^{j w_c n} = \cos(w_c n) + j\sin(w_c n)$

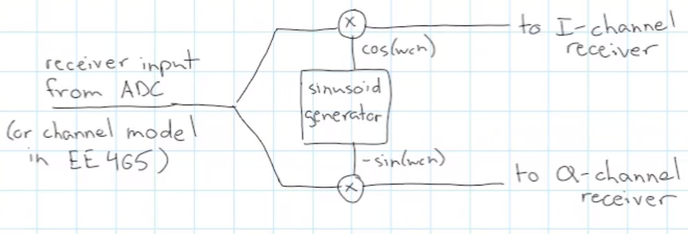
$$x_{rf}[n] = \big(x_I[n] + j x_Q[n]\big)\big(\cos(w_c n) + j\sin(w_c n)\big)$$
$$= \underbrace{x_I[n]\cos(w_c n) - x_Q[n]\sin(w_c n)}_{\text{real}} + \underbrace{j x_Q[n]\cos(w_c n) + j x_I[n]\sin(w_c n)}_{\text{imag}}$$

* only real part of $x_{rf}[n]$ can be sent to DAC + sent on the cable ...



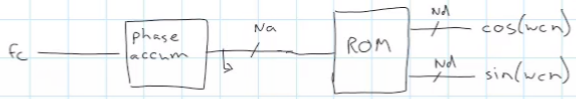
Downconversion: input from cable / ADC is real, say $x_c[n]$

$$x_{dc}[n] = x_c[n]\,e^{-j w_c n} = x_c[n]\big[\cos(w_c n) - j\sin(w_c n)\big] = x_c[n]\cos(w_c n) - j x_c[n]\sin(w_c n)$$



Since FPGA doesnt do imaginary numbers, wedo the above

For arbitrary up/downconversion, need variable sinusoid generators (NCOs)



- choices for Na, Nd control sinusoid SNR + spurious emissions

This block should upconvert to any frequency as well as downconvert to any frequency

fc is carrier frequency which is passed to phase accumulator which generates an address into a rom which stores a period of the sine

variable sine requires num of address bits NA. Superious emissions refers to the red noises in the frequency due to imperfect sine multiplication

We can bypass this complexity due to the design choice of D4 and simplify this process

$$F_c = 6.25\text{MHz} \longrightarrow w_c = \frac{6.25\text{MHz}}{25\text{MHz}} \times 2\pi = \pi/2 \text{ rad/sample}$$

carrier freq is 6.25 MHz, to get digital frequency, we divide by sampling frequency then multiply by 2pi to get the carrier frequency @ pi/2 radians/sample

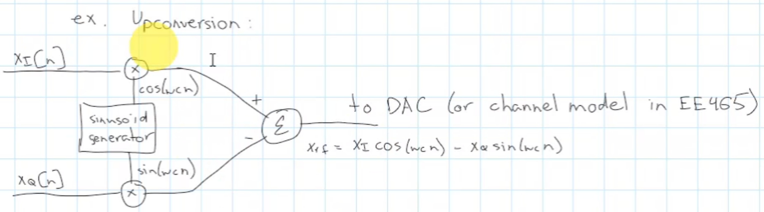
∴ the sequences to multiply by are:

$$\cos(\pi/2\, n) = [1 \quad 0 \quad -1 \quad 0 \quad 1 \quad 0 \quad \cdots]$$
$$\sin(\pi/2\, n) = [0 \quad 1 \quad 0 \quad -1 \quad 0 \quad 1 \quad \cdots]$$

since our sampling is @ pi/2 rads per sample, our NCO will always be +/-1 & 0 thus we need no multipliers for the upconversion

* no actual multiplications are needed!

ex. Upconversion:

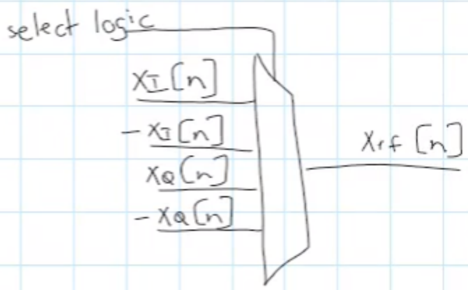


n	0	1	2	3	4	5 ...
$\cos(\pi/2\, n)$	1	0	-1	0	1	0
$\sin(\pi/2\, n)$	0	1	0	-1	0	1
$x_I \cos(\pi/2\, n)$	$x_I[0]$	0	$-x_I[2]$	0	$x_I[4]$	0
$x_Q \sin(\pi/2\, n)$	0	$x_Q[1]$	0	$-x_Q[3]$	0	$x_Q[5]$
x_{rf}	$x_I[0]$	$-x_Q[1]$	$-x_I[2]$	$x_Q[3]$	$x_I[4]$	$-x_Q[5]$

Notice that the output of Xrf alternates between -/+ x_i & x_Q

* each sample out of the upconverter is either x_I, $-x_I$, x_Q, or $-x_Q$

→ idea: implement with multiplexing logic (rather than NCOs + mixers)

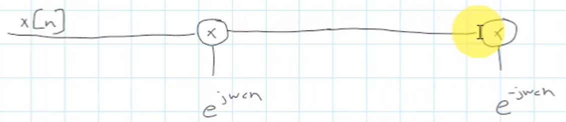


- similar idea can be applied to downconversion

we can replicate the mixing of a NCO using a MUX where we need to determine the sel logic to output the correct sequence

note that this concept applies to upconv but similar idea can be applied to down

Overall up/downconversion:



recall from EE 456: coherent receiver
(requires phases of oscillators in Tx and Rx to be identical)

- each sample gets multiplied by the conjugate of what it was multiplied by in the transmitter:

$$x[n]e^{j\omega_c n}e^{-j\omega_c n} = x[n]$$

- what if they don't match?
(due to different startup times or delay through channel)

remember the concept of coherent rcv, where the phase of the osciallators in TX and RX are synchornized to be identitcal

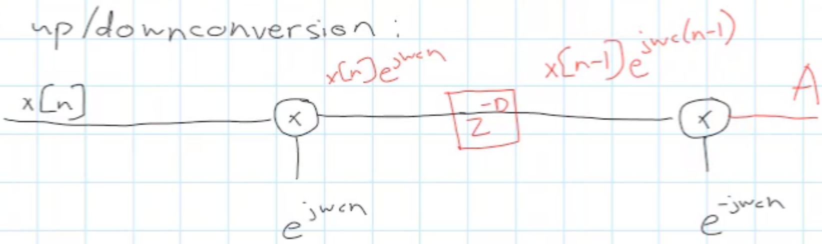
each sample is mixed with its conjuagate to return the signal @ baseband

may not match due to startup times, synchronization issues, etc

lucky we build both TX and RCV in the FPGA so we can sync both; however between the TX and RCV we have a channel model which may delay and impact our coherent recv

assume we have a single delay, D=1

Overall up/downconversion:



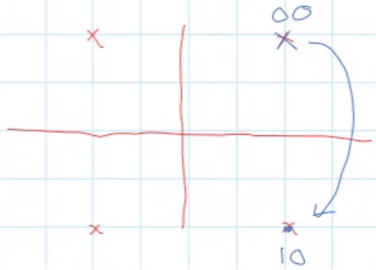
recall fro
(req
-each s
multipl
-what
(du

$x[n-1]\,\cancel{e^{j\omega_c(n-1)}}\,\cancel{e^{-j\omega_c n}}$

at A: $x[n-1]\,e^{-j\omega_c}$ $\pi/2$

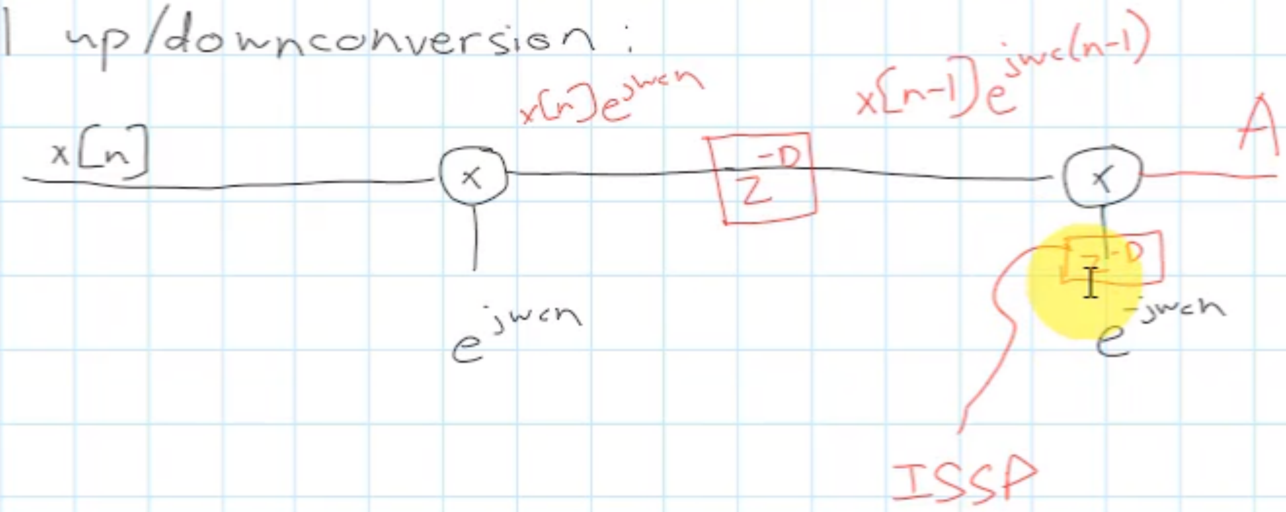
ex. chann

$x[n-1]\,\underbrace{e^{-j\pi/2}}$ → rotate by $-\pi/2$ radians



this delay can cause our output to be incorrectly rotated by pi/2 rads; wc is the carrier frequency in digital

Overall up/downconversion:



advice: add some sort of delay block to offset the delay in the channel so that A is just x[n-1] which shifts phase of sinusoid

connect the delay @ RCV to align your outputs