TUNNING

- Tuning, consist on selecting the right value for the LO and the appropriated sampling rate.
- All the tuning parameters are setup on Simulink block.
- You should be carful with the LO difference between SDR boards.
- On B200, the sample rate is defined by the relation between clock and up/down sampling.
THE SYNC PROBLEM

- Channel effects: A) Propagation delay B) frequency shift
- If we could consider No delay, always we have difference between LOs.
If TX and RX are fixed position, we always have fixed phase offset.
FREQUENCY SHIFT

- If TX and RX moves, there will be a time variance phase shift that is equivalent to a frequency shift.
- Doppler effect.
HARDWARE EFFECTS

- LO change over time due to: temperature, manufacture process, ageing and others.
- Example: error of 5000ppm, \( f_1 = 100.5 \text{Mhz} \), \( f_2 = 99.5 \text{Mhz} \), \( F_c = 100 \text{Mhz} \)
• Demodulate with an LO that is not exactly the same....problems, so **Solutions:**

• Option 1: TX the carrier like DSB-TC (pros/con)

• Option 2: **Recovering the carrier from the modulated signal.**

• Modulated Signal (DSB-SC)

\[ m(t) = \frac{A}{2} \left[ \sin(2\pi(f_c + f_b)t) - \sin(2\pi(f_c - f_b)t) \right] \]

• Demodulated Signal

\[ x(t) = \cos(2\pi(f_c + f_{\Delta})t) \times \frac{A}{2} \left[ \sin(2\pi(f_c + f_b)t) - \sin(2\pi(f_c - f_b)t) \right] \]

\[ u(t) = \frac{A}{4} \sin(2\pi(2f_c + f_{\Delta} + f_b)t) - \frac{A}{4} \sin(2\pi(f_{\Delta} - f_b)t) - \frac{A}{4} \sin(2\pi(2f_c + f_{\Delta} - f_b)t) + \frac{A}{4} \sin(2\pi(f_{\Delta} + f_b)t) \]

\[ u(t) = \frac{A}{4} \left[ \sin(2\pi(f_b + f_{\Delta})t) + \sin(2\pi(f_b - f_{\Delta})t) \right] \]
The PLL is the fundamental component on every coherent receiver.

PLL:
- Phase Detector: the output is proportional to the phase difference between received signal and locally generated.
- Controlled Oscillator: it is a VCO for analog receiver and NCO for digital receiver.
- Loop filter: Filter acts upon the output of the Phase Detector to remove unwanted high frequency terms, and produce the signal that drives the VCO or NCO.
The phase error is proportional to the phase difference:

\[ \theta_e(t) = K_p(\theta_i(t) - \theta_o(t)) \]

- Implemented as a multiplier.
- If the difference is small, the mixing approach to the difference.

\[ y(t) = x(t) \times s(t) \]

\[ = \cos(2\pi f_i t + \theta_i(t)) \times -\sin(2\pi f_o t + \theta_o(t)) \]

\[ y(t) \approx \frac{1}{2} \sin(\theta_i(t) - \theta_o(t)) + \text{high frequency terms} \]

\[ y(t) \approx \frac{1}{2} \left( \theta_i(t) - \theta_o(t) \right) \]
The design of the loop filter is vital in defining the overall characteristics and behavior of the PLL.

The PLL type corresponds to the number of integrators in the loop, including the one that add the VCO/NCO.

<table>
<thead>
<tr>
<th>PLL Type</th>
<th>Integrators</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>VCO/NCO</td>
<td>Loop Filter</td>
<td>Total</td>
</tr>
<tr>
<td>Type 1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Type 2</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Type 3</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>
CONTROLLED OSCILLATOR

- The VCO control signal is the filtered phase difference $v(t)$.

- The estimated phase at instant "t" is:
  \[ \hat{\theta}(t) = k_0 \int_{0}^{t} v(t) \, dt \]

- $k_0$ is the sensibility of VCO.

- $c(t) = \cos\left(2\pi f_0 t + \hat{\theta}(t)\right)$

![Diagram showing the controlled oscillator and its output signals](image)
NCO: NUMERICALLY CONTROLLED OSCILLATORS

- The digital version of VCO.

\[ c[m] = \cos\left(2\pi f_o m T + \hat{\theta}[m]\right) \]

\[ \hat{\theta}[m] = K_o \sum_{m=0}^{M} v[m] \]

\[ \hat{\theta}[m] = \hat{\theta}[m-1] + \mu[m] \]

- Different frequencies are created from different step size.

\[ \mu = \frac{2\pi f_d}{f_s} \]
To follow a frequency change, an adjustment term is needed.

\[
\hat{\theta}[m] = \hat{\theta}[m-1] + \mu[m] \quad \mu[m] = \mu_q + \mu_a[m] \quad \mu_a[m] = K_o v[m]
\]
The simplest operation inside FPGA could be up/down conversion to intermediate frequency.

NCO are usually implemented on LUTs. Also called Digital Direct Synthesizer (DDS).

The LUT has $N=2^n$ size, where $n$ is the numbers of bits that accumulator generate.

The amplitude resolution of the signal depends on the number of outputs bits $L$, and the frequency resolution depends on LUT size, $n$. 
NCO ON FPGA

- Frequency and amplitude resolution

<table>
<thead>
<tr>
<th>address</th>
<th>amplitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>0000 0000</td>
</tr>
<tr>
<td>0001</td>
<td>0011 0001</td>
</tr>
<tr>
<td>0010</td>
<td>0101 1011</td>
</tr>
<tr>
<td>0011</td>
<td>0111 0110</td>
</tr>
<tr>
<td>0100</td>
<td>0111 1111</td>
</tr>
<tr>
<td>0101</td>
<td></td>
</tr>
<tr>
<td>0110</td>
<td></td>
</tr>
<tr>
<td>0111</td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td></td>
</tr>
<tr>
<td>1001</td>
<td></td>
</tr>
<tr>
<td>1010</td>
<td></td>
</tr>
<tr>
<td>1011</td>
<td></td>
</tr>
<tr>
<td>1100</td>
<td></td>
</tr>
<tr>
<td>1101</td>
<td></td>
</tr>
<tr>
<td>1110</td>
<td>1010 0101</td>
</tr>
<tr>
<td>1111</td>
<td>1100 1111</td>
</tr>
</tbody>
</table>

Memory Locations: $N = 2^4 = 16$
NCO ON FPGA

- Quantization effect, L
• Frequency control is done by the **step** of Accumulator

![Diagram](image_url)
The Accumulator step is determined by:

- N: number of entries on the LUT.
- Fs: sampling frequency
- Fd: desired frequency

Example 8 bits, N=256, fs=10Mhz, fd=2.5Mhz

If we need 2.4Mhz??

We need to add a fractional part to the ACC.

The step will have an integer and a fractional part [n:b]

\[ \mu = N \frac{f_d}{f_s} \]

\[ \mu = 256 \frac{2.5MHz}{10MHz} = 64 \]

\[ \mu = 256 \frac{2.4MHz}{10MHz} = 61.44 \ldots \]

\[ \frac{f_s}{N} = 2, 399, 902Hz \]

\[ \mu = 61.4375 [8:4] \]
NCO ON FPGA

• Frequency resolution: depends on steps differences.

• The fractional bits should be selected following the desired frequency resolution.

• Frequency resolution $\Delta f_a$:

$$\Delta f_a = \frac{f_s}{N} = \frac{f_s}{2^b N}$$
• Truncated error: if the step is $\mu=1.7$, then spurious appear due to truncate.
NCO ON FPGA

- TRUNCATED ERROR due to small N.
- Fs: 100kHz, fd: 24.3kHz, N:6, [n:b]=[6:16], L=32
NCO ON FPGA

- QUANTIZATION ERROR, small L

- fs: 100kHz, fd: 24.3kHz, L:8, [n:b]=[12:16]
NCO ON FPGA

- SFDR: Range free of spurious frequencies
- GSM requires 110Db of SFDR
NCO ON FPGA

- Increase LUT size helps, but **cost a lot**.
- Better solution: add a dither signal to **break the quantization error**.
- Usually the number of dither bits is equal to fractional bits, \( b = d \)
NCO ON FPGA

- DIRECT DIGITAL SYNTHESIZER
LOOP FILTERS TYPES

Type 1 Loop Filter

Type 2 Loop Filter

Type 3 Loop Filter

Phase Detector

Loop Filter

NCO
**DESIGN PARAMETERS**

- Time to achieve lock, depends on the step size.
- Steady state error, depends on the number of integrators and how the input signal change.
- Tracking capabilities, deepens on the PLL type.
• $\zeta < 1$ under-damped  \quad $\zeta > 1$ over damped  \quad $\zeta = 1$ critically damped

• The damping factor, or damping ratio, relates to the transient behavior of the PLL as it achieves phase lock.

• Typical value is 0.707
The bandwidth refers to the range of frequencies over which the PLL operates.

At lower BW, bigger transient time at bigger BW lower transient. Cons: more noise into the PLL.
COSTAS LOOP

• It is a type of PLL used on AM-DSB-SC demodulation. Also used on M-PSK demodulations.

• It is based on the sin vs cos orthogonality.

• Principal advantage is its double sensibility. \( \text{sen}(2(\theta_i - \theta_f)) \)

• Especially useful for Doppler effect correction.

\[
r(t) = A(t) \cos(2\pi f_c t + \phi)
\]

\[
\hat{I}(t) = A(t) \cos(\phi - \hat{\phi})
\]

\[
\hat{Q}(t) = A(t) \sin(\phi - \hat{\phi})
\]

\[
e(t) = A^2(t) \cos(\phi - \hat{\phi}) \sin(\phi - \hat{\phi})
\]

\[
= \frac{1}{2} A^2(t) \sin(2(\phi - \hat{\phi}))
\]
COSTAS LOOP FPGA IMPLEMENTATION

- Decision Direct PLL
COSTAS LOOP EXAMPLE

• 4QAM RX/TX