Mixer Notes

Piyush Sud

9/30/2024

1 High Level Design

For the mixer, we need:

- a large bandwidth (at least 200 MHz)
- low noise

The LT5560 works for this application. It has:

- a large bandwidth (4 GHz)
- low noise (9.3 dB @ 900 MHz)

Design Notes:

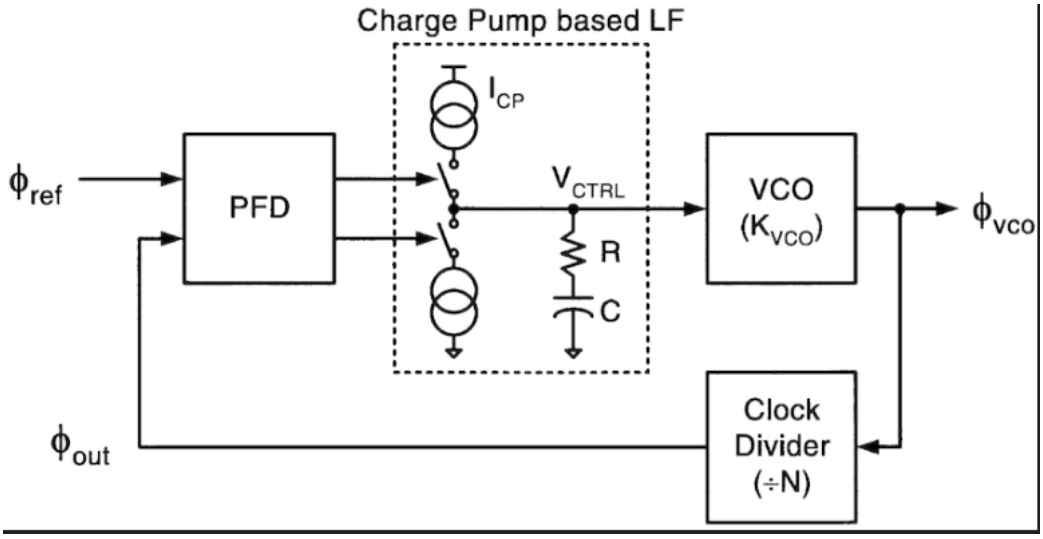
- It seems like Automatic Frequency Control (AFC) is no longer used since crystals provide a very stable frequency, as seen on both wikipedia and this link: https://www.quora.com/What-is-frequency-drift-in-an-FM-receiver. Most LOs today use digitally synthesized signals connected to crystals. These are known as "frequency synthesizers" - they are able to generate many different stable frequencies using a single reference frequency from a crystal.
- PLLs also don't drift because any error is compensated for in the PLL.
- It seems like we can just use a PLL with a VCO built in. This seems like a good option: https://www.digikey.com/en/products/detail/analog-devices-inc/ADF4360-9BCPZRL7/2043337
- Actually, it seems like this is an integer-N PLL, but we need a fractional N PLL.
- Even if we find another PLL with an integrated VCO, it seems we'll still need an external loop filter, so might as well stick with the original with an external VCO

2 Detailed Design

- This seems like a really good PLL: https://www.digikey.com/en/products/detail/skyworks-solutions-inc/SI5351A-B-GTR/4069612. Unlike the previous one, it uses a crystal, so the frequency error is close to 0.
- However, it seems like it can only divide by multiples of 2, so it wouldn't work for our purposes since we need to change the LO frequency from 77 MHz to 98 MHz.
- This might be a better option: https://www.digikey.com/en/products/detail/analog-devices-inc/ADF4169CCPZ/5441070
- In a high-side mixer, the FM modulated signal will essentially be inverted. This is because $f_{IF} = \mid f_{RF} - f_{LO} \mid$, so if f_{RF} increases then f_{RF} becomes closer to f_{LO}, so f_{IF} decreases. This doesn't really matter (not completely sure about this), as long as we're aware that it's mirrored. https://www.edaboard.com/threads/high-side-injection-and-low-side-injection-in-mixer.125131/
- Let's use low-side mixing so that the signal is not inverted. This means that fLO = fRF - fIF = fRF - 10.7MHz.
- Does a clock generator work (square wave) or do I need a sine wave for the LO?
- According to this, it seems like a square wave can cause higher order harmonics to be aliased down into the frequency range of interest. https://www.reddit.com/r/rfelectronics/comments/12wqxvw/rf_mixer_square_wave_lo/. Therefore, let's use a sine wave LO.
- If we're using the ADF4169CCPZ above, we'll need to write to the registers to control the input, which means we'll need a microcontroller to do that.
- It doesn't explicitly mention it, but looks like it uses I2C since it has a clock and data line. If not, we can just manually code the protocol instead of using an I2C library.

- We can use the same 0.1 uF cap we used as a blocking cap for the HF amp as a decoupling cap here since it has a very high self-resonant frequency of 6 GHz.
- The INT multiplier for the reference frequency is from 23 to 4095. If we want a LO signal of 77 MHz to 97 MHz, then we can potentially use a reference frequency around 1 MHz.
- Let's use this CMOS crystal oscillator: https://www.digikey.com/en/products/detail/analog-devices-inc-maxim-integrated/MAX7375AXR105-T/1520107

3 Understanding PLLs



- Assuming that the input is a square wave and the PFD is a digital circuit, a simple PFD can be implemented using an XOR gate that is high for the portion of time when the reference signal and the output signal are different. If the two signals are 1 degree out of phase, this will result in a pulse with a width of 1/180 the period of the signals.
- The charge pump performs integration, which can correct for both phase and freq. errors, since freq. is the derivative of phase.

- The charge pump for the ADF4169 outputs a current, which is converted to a voltage by the loop filter. https://electronics.stackexchange.com/questions/301056 filter-of-pll

- Since a LPF also performs integration, a charge pump can technically replace a LPF with a capacitor.

- This seems like a good VCO: `https://www.digikey.com/en/products/detail/analog-devices-inc-maxim-integrated/MAX2623EUA-T/1938007`

4 Loop Filter Design

- Following this link, second order loop filter design: `https://www.renesas.cn/cn/zh/document/apn/pll-loop-filter-design-and-fine-tuning`

- Desired loop bandwidth must satisfy Fpd/Fc >> 20, where Fpd is the freq. to the phase detector = the frequency of our ref signal, which is 1 MHz. This means that Fc ¡¡ 1 MHz/20 = 50 kHz. Let's set the loop bandwidth to 10 kHz.

- Rs = $\frac{2*pi*f_c*N}{/} I_{cp} * K_{Vco}$. N varies between 77 and 97, so let's choose 87 as the nominal value. According to the datasheet, it is best to start with 2.5 mA for the charge pump current. The VCO gain is given by 6.2 MHz/V. Therefore, $Rs = \frac{2*3.1415*10000*87}{2.5*10^-3*6.2*10^6} = 352.658709677$ ohms.

- Cs = $\frac{alpha}{2*pi*f_c*R_c}$, where alpha is the ratio of the loop bandwidth to the zero frequency. It's recommended that alpha ¿ 3, so let's do alpha = 5. That gives $Cs = \frac{5}{2*3.1415*10000*352.6587} = 225.656757$ nF.

- Cp = $\frac{Cs}{alpha*Beta}$, where beta is the ratio between pole freq. and bandwidth. It's recommended that beta ¿ 3, so let's choose beta = 5. That gives $Cp = \frac{225.656757*10^-9}{5*5} = 9.02627028$ nF.

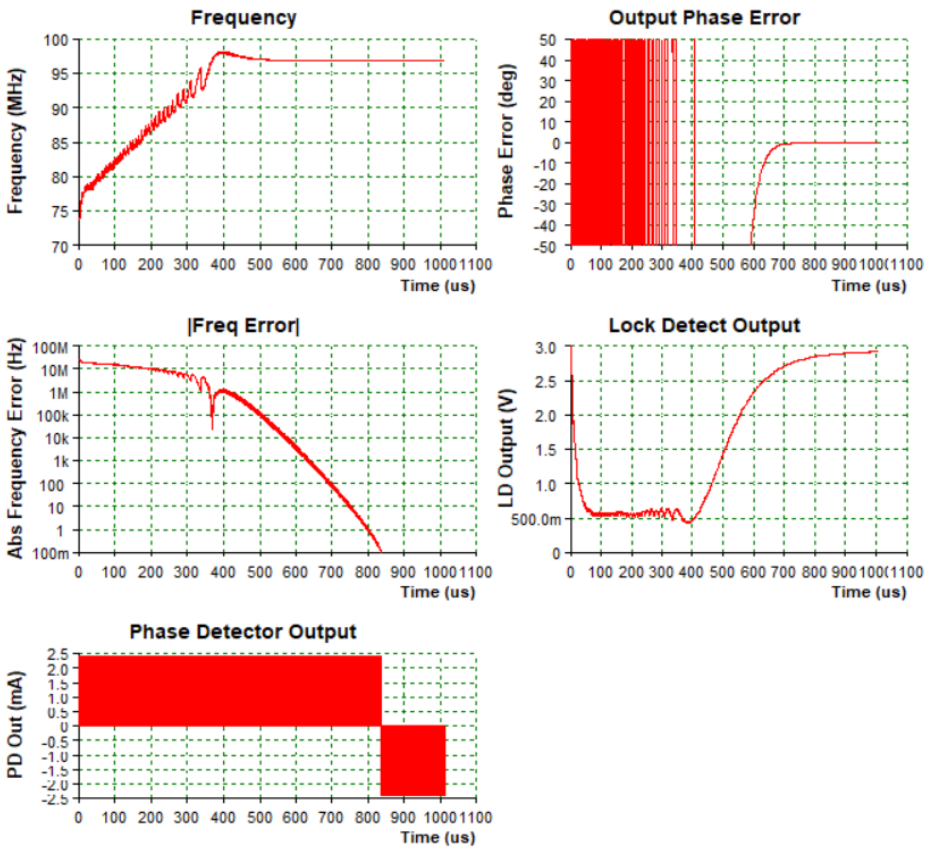
- The maximum phase margin is given by $arctan(\frac{b-1}{2*sqrt(b)})$, where $b = 1 + \frac{Cs}{Cp}$. In our case, $b = 1 + \frac{225.656757}{9.02627028} = 26$, so the phase margin = 67.80839366845 degrees > 50 degrees − > the PLL is stable.

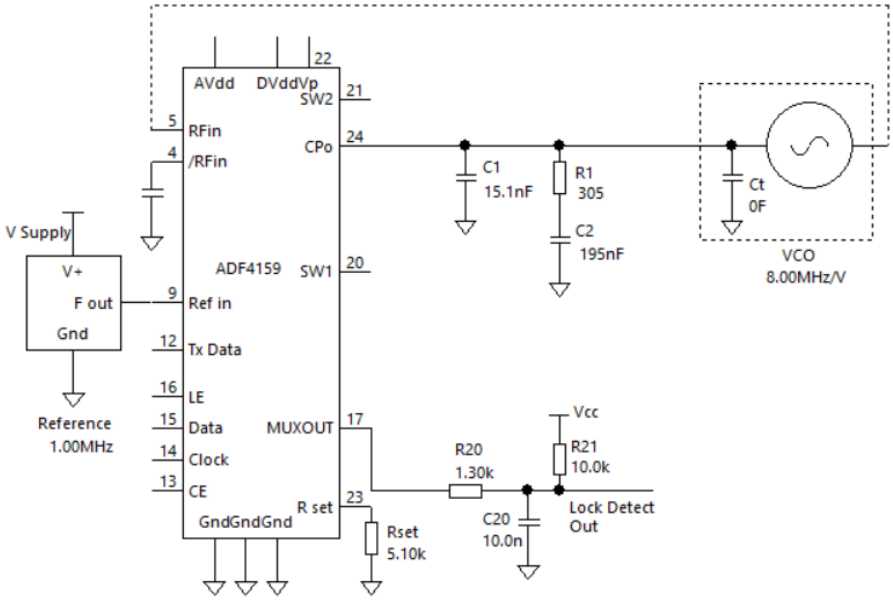
5 Simulating the Mixer

Right now, my main goal is to make sure the circuit works on a basic level. For this, let's just simulate the circuit using an ideal mixer in LTSpice. Later we can optimize for distortion. There's no point simulating the mixer IC since we know that will work (with some nonidealities which we don't care about since we're just testing basic functionality).

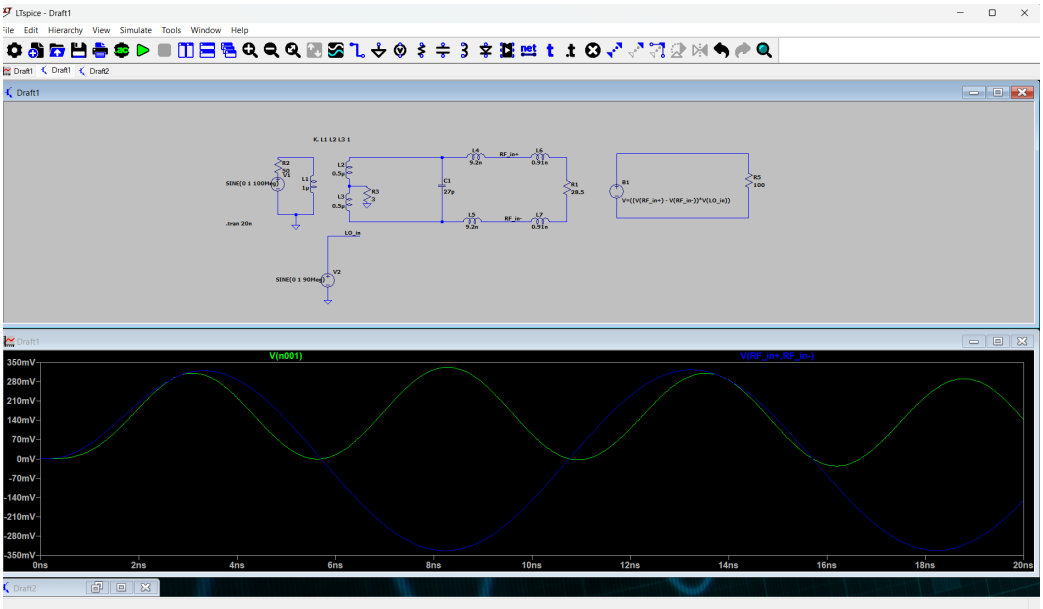
For the LO Simulation, let's use ADISimPLL to confirm the LO locks within a reasonable time and is stable. It seems like we were using hand calcs before using renesas instead of ADISimPLL. But ADISimPLL will be more accurate.

Simulated using ADISimPLL to find loop filter values, lock time = 500 us, PM = 60 degrees, loop BW = 10 kHz, CP current = 2.4 mA.


Frequency
Frequency (MHz)
100
95
90
85
80
75
70
0 100 200 300 400 500 600 700 800 900 1000 1100
Time (us)
Output Phase Error
Phase Error (deg)
50
40
30
20
10
0
-10
-20
-30
-40
-50
0 100 200 300 400 500 600 700 800 900 1000 1100
Time (us)
|Freq Error|
Abs Frequency Error (Hz)
100M
10M
1M
100k
10k
1k
100
10
1
100m
0 100 200 300 400 500 600 700 800 900 1000 1100
Time (us)
Lock Detect Output
LD Output (V)
3.0
2.5
2.0
1.5
1.0
500.0m
0
0 100 200 300 400 500 600 700 800 900 1000 1100
Time (us)
Phase Detector Output
PD Out (mA)
2.5
2.0
1.5
1.0
0.5
0
-0.5
-1.0
-1.5
-2.0
-2.5
0 100 200 300 400 500 600 700 800 900 1000 1100
Time (us)



Notes:
1. LFCSP pin numbers shown
2. AVdd Analog Power supply
3. DVdd Digital Power Supply
4. Vp Charge Pump power supply
5. AVdd = DVdd, Vp >= DVdd,AVdd
6. Consult manufacturer's data sheet for full details


LTspice - Draft1
File Edit Hierarchy View Simulate Tools Window Help
Draft1
Draft2
K_L1 L2 L3 1
R2
V1
SINE(0 1 100Meg)
L1
1µ
L2
0.5p
R3
2
L3
0.5p
L4
9.2n
RF_in+
L6
0.91n
C1
27p
R1
28.5
L5
9.2n
RF_in-
L7
0.91n
B1
V=(V(RF_in+) - V(RF_in-))*V(LO_in)
R5
100
.tran 20n
LO_in
V2
SINE(0 1 90Meg)
V(n001)
V(RF_in+,RF_in-)
350mV
280mV
210mV
140mV
70mV
0mV
-70mV
-140mV
-210mV
-280mV
-350mV
0ns
2ns
4ns
6ns
8ns
10ns
12ns
14ns
16ns
18ns
20ns