Integrated Radio Electronics

Laboratory 1: Low-Noise Amplifier

Rikard Gannedahl, Iman Ghotbi, Henrik Sjöland

January 2022

1. Introduction

The Low-Noise Amplifier (LNA) is typically the first active stage in a receiver. Its job is to suppress the subsequent stages' noise contribution by adding as much gain as possible, while adding as little of its own noise as possible. It should also provide good matching to the off-chip environment.

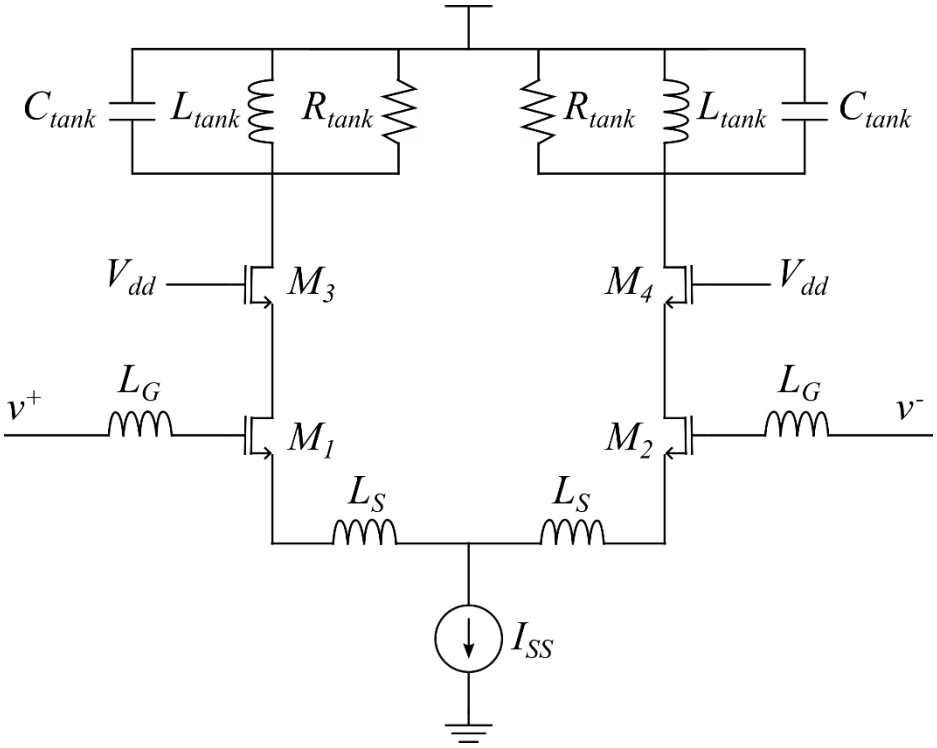

Figure 1: The inductively degenerated LNA.

In this lab, we will design and simulate a differential inductively degenerated common-source amplifier as our LNA, Figure 1. This is one of the most popular LNA architectures due to its excellent noise performance. The frequency of operation will be 5 GHz.

The LNA has a lot of different parameters to tweak, and it can be hard to know where to start. A common approach is to set an upper limit for the power consumption and try to optimize for that. In this lab, we will limit the power consumption to 6 mW. Since *VDD* = 1.2V, this means that I_{SS} = 5 mA, i.e. 2.5 mA per branch.

One of the important tasks for the LNA is to provide an impedance match to the outside environment, i.e., the antenna, to enable maximum power transfer and avoid reflections. The

antenna impedance is typically purely resistive, meaning that the input impedance should also be purely resistive. The input impedance of this LNA is:

$$Z_{in} = \frac{1}{sC_{gs}} + s(L_S + L_G) + \frac{g_m L_S}{C_{gs}} \approx \frac{1}{sC_{gs}} + s(L_S + L_G) + \omega_T L_S$$

Interestingly, the impedance has a resistive part that depends on the degeneration inductor and the transition frequency of the input transistors M_1 and M_2. To make the impedance purely resistive, we need the $1/sC_{gs}$ and the $s(L_S + L_G)$ terms to cancel at our desired frequency ω_0, while $\omega_T L_S$ matches the antenna resistance. In other words, we need $\omega_T L_S = R_S$, where R_S is the resistance of the antenna, and $\frac{1}{C_{gs}(L_S+L_G)} = \omega_0^2$.

Another important aspect is of course the noise. The most critical sources of noise are M_1 and M_2, which will cause both thermal drain noise and induced gate noise. A useful parameter when analyzing the noise is the quality factor Q of the input impedance, defined as:

$$Q = \frac{1}{(R_S + g_m \frac{L_S}{C_{gs}})\omega_0 C_{gs}} = \frac{1}{2R_S\omega_0 C_{gs}}$$

As it turns out, higher Q leads to lower drain noise, but higher induced noise. In the course book, Lee uses this fact to find an optimum Q which causes the lowest noise figure. It is shown that this Q is a constant, which in turn leads to an equation for the optimum transistor width (eq. (12.58) in the book). However, his approach does not consider other aspects of the LNA, such as linearity, and does not perform very well for low bias currents. In addition to this, it typically leads to very large inductors. Another approach[1] is to add an extra capacitance C_{ex} in parallel with C_{gs} of the input transistors, so that the total capacitance between gate and source is $C_t = C_{ex} + C_{gs}$. The quality factor then becomes:

$$Q = \frac{1}{2R_S\omega_0 C_t}$$

This gives the designer an extra degree of freedom by detaching the width of the transistor to the Q, making Q a variable that can be set to fulfill the specifications of the design. After a quite lengthy derivation it is shown that the optimum width will be:

$$W_{opt} = \frac{A_b}{2Q^2}\sqrt{\frac{5}{6}}\frac{1}{\frac{4}{3}\omega_0 R_S C_{ox} L},$$

where A_b is a constant called bulk charge factor (typically around 1.2-1.4), $C_{ox}$is the gate oxide capacitance, and L the length of the transistor (should be as small as possible, i.e. 65nm in our case). The ratio between C_{gs} and C_t is then shown to be:

$$P_{opt} = \frac{C_{gs}}{C_t} = \frac{A_b}{2Q}\sqrt{\frac{5}{6}}$$

Setting Q = 3 (a quite typical value), A_b = 1.4, C_{ox} = 14 fF/µm^2, L = 0.06µm, and R_s = 50Ω gives:

$$W = 47\mu\text{m}$$

[1] P. Andreani and H. Sjöland, "Noise optimization of an inductively degenerated CMOS low noise amplifier," in *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, Sept. 2001

$$P = 0.21 \Rightarrow C_{ex} = 3.76C_{gs}$$

A DC simulation shows that, for a transistor with the dimension W/L = 47µm/0.06µm and I_{DS} = 2.5mA, C_{gs} = 14fF (intrinsic capacitance) and $C_{gs,ol}$ = 14fF (overlap capacitance). $C_{gs,ol}$ will be part of C_{ex}, so the capacitance we need to add will be: $C_{ex}' = C_{ex} - C_{gs,ol} = 3.76C_{gs} - C_{gs,ol}$ = 38fF.

We can now return to impedance matching. Because of the added capacitance, we must slightly modify our expression from before. The real part of the impedance is now $\frac{g_m L_S}{C_t}$ and to cancel the imaginary part we need $\frac{1}{C_t(L_S+L_G)} = \omega_0^2$. The above-mentioned DC simulation showed g_m = 23 mS. For R_S = 50 Ω, this gives:

$$L_S = R_S \frac{C_t}{g_m} = R_S \frac{C_{ex}+C_{gs}}{g_m} \approx 140\text{pH}$$

To make the imaginary part 0 we now need:

$$L_G = \frac{1}{\omega_0^2 C_t} - L_S \approx 15\text{nH}$$

For the cascode devices, we use the same W/L ratio, but double both the width and length. Otherwise, we can get an unstable amplifier due to short-channel effects.

The only unknown values now are for the resonance tank. Here, we need to make some guesses. C_{tank} will be the capacitance from the next stage (typically a mixer), plus the parasitic capacitance from the inductor, plus any kind of capacitance that we choose to explicitly add. Let's guess that this will be roughly 200 fF. This gives an inductor value of:

$$L_{tank} = \frac{1}{\omega_0^2 C_{tank}} = 5.1\text{nH}$$

If we don't explicitly add a resistor at the output, R_{tank} will only be due to the finite Q-value of L_{tank}. A typical value for Q at these frequencies is 10, which gives a R_{tank} of:

$$R_{tank} = \omega_0 L_{tank} Q = 1.6\text{k}\Omega$$

An AC simulation shows that L_{tank} = 3.8 nH causes resonance at 5 GHz (this lower value is because of the added C_{gd} and C_{db} capacitances from M_3 and M_4), which in turn gives R_{tank} = 1.2kΩ.

The initial values to be used in the simulations are summarized below:

W	47 µm
L	60 nm
$W_{cascode}$	2W
$L_{cascode}$	2L
C_{ex}'	38 fF
L_S	156 pH
L_G	14.3 nH
L_{tank}	3.8 nH
C_{tank}	200 fF
R_{tank}	1.2 kΩ
I_{SS}	5 mA

2. Setting up the testbench

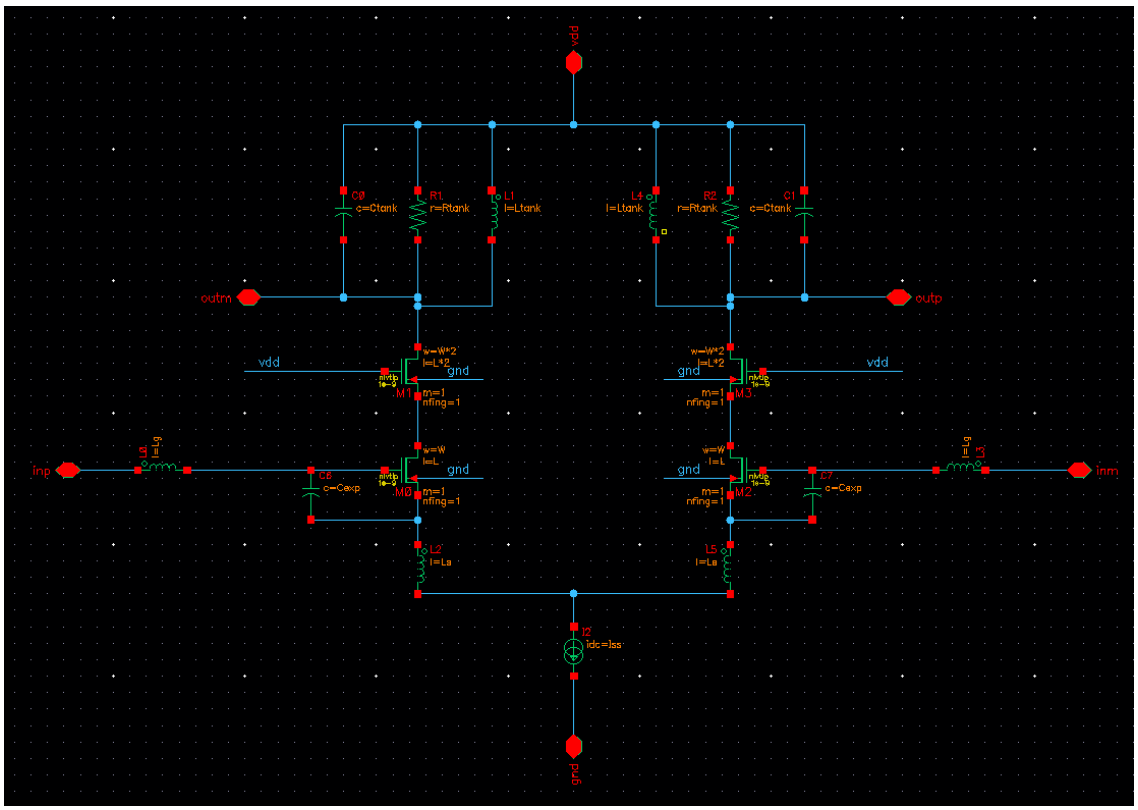

Figure 2: The LNA schematic in Cadence.

Firstly, create a library called RFIC_Labs and attach it to “cmos065”. Then, create a schematic called LNA according to Figure 2. Use ideal passive components and the *nlvtlp* transistor. Instead of giving exact values to all the components, use variables. Create a symbol of the LNA (“Create” -> “Cellview” -> “From Cellview”). Next, create a new cellview called LNA_tb and place components according to Figure 3. The components you most likely are unfamiliar with are “port”, “ideal_balun” and “vcvs” (voltage-controlled voltage source), all of which can be found in analogLib. Use all the default values, except change the resistances in the ports to 100 Ω. We do this because we have calculated for an input impedance of 50 Ω of each input, thus the differential input resistance should be 100 Ω[2]. The output port does not really do anything, but it is needed in some simulations. However, we do not want its resistance to load the output, which would cause the gain to degrade. Thus, we use a vcvs to isolate the port from the output.

[2] This is not always the case. Sometimes we design the differential input resistance to be 50Ω, i.e. 25Ω at each input.

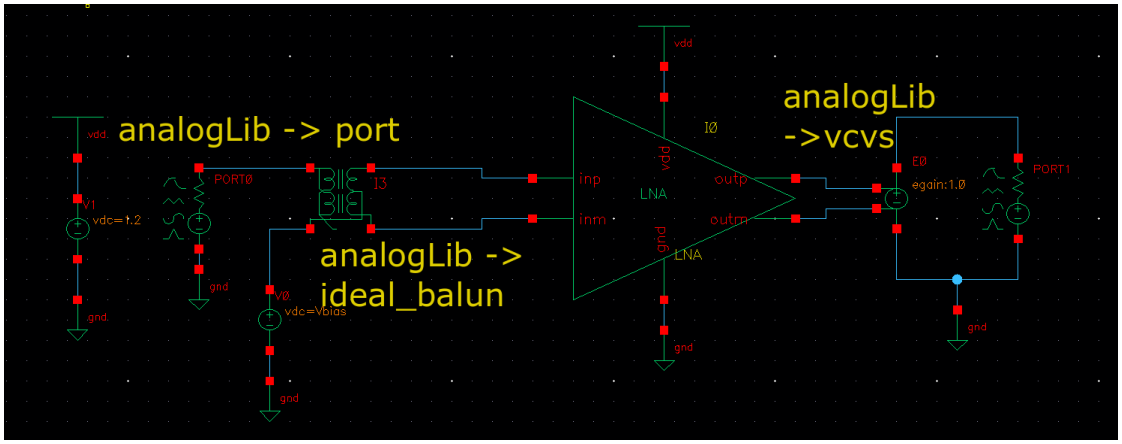

Figure 3: The LNA testbench.

Launch ADE L, import all variables (“Variables” -> “Copy from cellview”), and give them the values from the Introduction and set V_{bias} to 600 mV. Setup the corners. Then, go to “Analyses” -> “Choose” -> “DC” -> tick “Save DC operating point” -> “OK”. Bring up the LNA in a new tab by selecting the LNA in the LNA_tb, press “e”, select “New tab” and “Edit”, and click “OK”. Go back to ADE L, select “Results” -> “Print” -> “DC operating point” and select one of the input transistors and check that everything seems reasonable. Another very useful thing you can do after a DC simulation is to go to ADE and select “Results” -> “Annotate” -> “DC Node Voltages”. This will print all the DC voltages in the schematic, making it very easy to spot if something is strange.

If everything looks OK, proceed to the next step.

3. Simulating the input match

Now we will see how good our input match is, i.e. how close our input impedance is to 50 Ω (or 100 Ω differentially). The easiest way to do is to use an S-parameter analysis, or sp in Cadence. In particular, we are interested in the S11 at this stage, which is measurement of how much of the generated power at the port is reflected back to the port. Go to ADE -> “Analyses” -> “Choose” -> “sp”. At the “Ports”-section, click on the “Select”, the click on your input port and then on your output port, then press the Esc-key. At the “Frequency”-section, select the start frequency as 1 GHz, and the stop frequency as 10 GHz. Leave the sweep type as “Automatic”. When everything is done, it should look like Figure 4. Press “OK” and run the simulation. sp is a small-signal analysis, just like ac, so the simulation should be very fast.

Figure 4: S-parameter setup.

After the simulation is done, go to “Results” -> “Direct plot” -> “Main form”. A new window will open, containing plotting options for every simulation that was run, except for dc. Under Function, choose “SP”, select “Rectangular” and “db20”, and click on “S11”-button. A plot will appear, plotting S11 versus frequency. Good matching is generally considered to be below –10 dB. Do you have good matching at 5 GHz?

While S11 is very useful for getting an overview of the matching, it doesn't really tell you why the matching is not perfect. For that, we can use Z11 instead, i.e. what impedance is seen by first port. Go back to the "Main form"-window, but this time, select "ZP" instead of "SP". Then select "Rectangular" and "Real", then press on "Z11". Now, the resistance seen by PORT0 is plotted. Repeat again but select "Imag" instead of "Real". This time, the reactance seen by PORT0 is plotted. To get perfect matching, the resistance should be 100 Ω and the reactance 0 Ω at 5 GHz. If the reactance is positive at this point, your resonance frequency is too low, and you should decrease the value of some of your reactive elements. If your reactance is negative, the opposite applies. If your resistance is off, you can tweak L_S. In this case, your resistance should be quite good, but the resonance frequency should be too low. This is because we haven't accounted for C_{gd} in our calculations, which causes a quite significant capacitive load due to the Miller effect, despite the usage of a cascode. The easiest way to solve this is to lower L_G since that does not affect the resistive part. Note also that there is a small hump in the real part of Z11. This is due to the resonance in the tank, which leaks all the way to the input due to the short channel lengths used here.

A helpful tip when using the "Main form" window is the "Add to Outputs". If you tick this option, Cadence will automatically generate an output expression in ADE, so that the next time you run the simulation, the same plot will directly be plotted. If you use this, remember to untick the option the next time you use the "Main form"-window, otherwise things can become quite chaotic.

4. Simulating gain and noise

For simulating the gain, we can use the sp-analysis again. Open the "Main form"-window, select "SP", "Rectangular", "dB20" and click on "S21".

What is your gain and 3-dB bandwidth?

For noise simulations, we will use the *noise*-analysis. This is also a type of ac-analysis, so it will be very fast. In ADE, go to "Analyses" -> "Choose" -> "noise". Once again, we set a frequency span between 1 GHz and 10 GHz. At "Output Probe Instance", click "Select" and click on the output port. At "Input Probe Instance", click "Select" and click on the input port. Click "OK" and run the simulation. Open the "Main form"-window. Note time that there are two options at the top, "sp" and "noise". Select "noise", "Noise Figure", "V / sqrt(Hz)", "dB20" and "Plot". Now, noise figure versus frequency will be plotted. At 5 GHz, you should have excellent noise performance, with a noise figure well below 1 dB.

A helpful tool when running a noise simulation is "Noise Summary". This will tell you where the noise is coming from, which can help us make necessary changes. In ADE, go to "Results" -> "Print" -> "Noise Summary…", select "spot noise", change "Frequency Spot (Hz)" to 5G, click on "Include All Types", and change "top" from 3 to 10, see Figure 5. Click "OK". Here you can see what device is generating the noise, what type of noise it is, and how much noise it generates. "rn" is resistive thermal noise, "id" is transistor thermal noise, and "fn" is flicker noise. For good performance, the thermal noise from PORT0 should completely dominate, since this means that we are adding very little extra noise to the system.

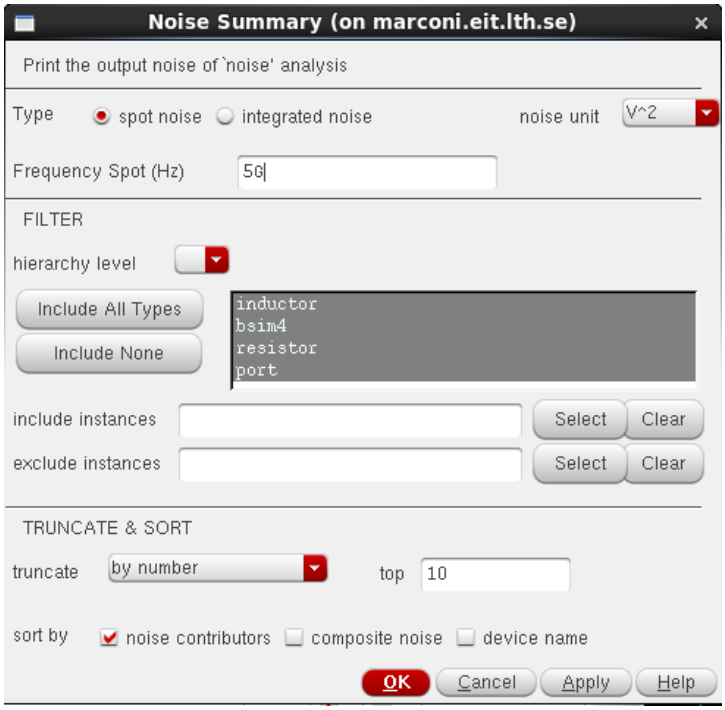

Figure 5: Noise summary setup.

In the noise summary, there is one major noise type missing: The induced gate noise. This is because the transistor models used in this PDK does not account for this kind of noise[3]. Therefore, we should add it manually. We can do this by adding a resistor and a voltage-controlled current source (vccs) in parallel with gate-source connection of the input transistors, see Figure 6. Now, we need to size the resistor and the transconductance *G* of the vccs correctly. The noise generated by the resistor is:

$$\overline{\iota_{nR}^2} = 4kTR\Delta f \cdot G^2$$

The induced gate noise is given by:

$$\overline{\iota_{ng}^2} \approx 4kT\frac{\delta\omega^2 C_{gs}^2}{5g_m}\Delta f$$

These equations should give the the same noise:

$$\overline{\iota_{nR}^2} = \overline{\iota_{ng}^2} \Rightarrow R \cdot G^2 = \frac{\delta\omega^2 C_{gs}^2}{5g_m}$$

The easiest way to implement this is to set R = 1Ω, and then calculate G.

Calculate G at 5 GHz, using δ = 2, and re-simulate the noise. Do you see any difference?

Another source of noise we have not accounted for yet is the noise coming from the series resistance of L_G. This noise can be very troublesome since it appears in the signal path before the active devices. The series resistance is given by:

$$R_G = \frac{\omega L_G}{Q}$$

[3] In more modern PDKs, this is typically modeled.

Luckily, this inductor can be implemented as a bondwire (a metal wire connecting a pad of the chip to the package/PCB), which have a high *Q*.

Calculate this resistance at ω_0, assuming a *Q*-value of 20, and repeat the noise simulation. You do not have to add an extra resistor, in the properties of the inductor there is a parameter called "Resistance" in which you can add the value of your series resistance.

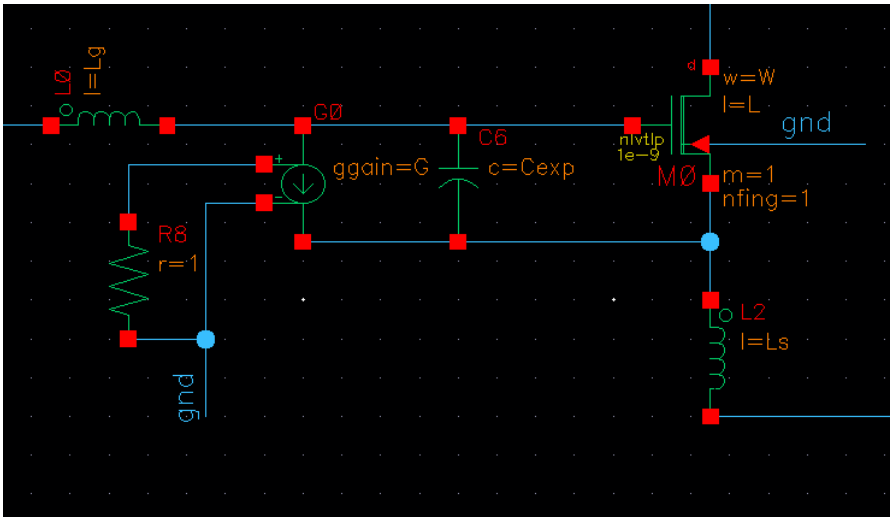

Figure 6: Resistor and vccs added to replicate induced gate noise.

5. Compression point simulation using PSS

We will now use one of the most powerful tools for RFIC design, the periodic steady state (PSS) analysis. PSS assumes (as the name implies) that your system is periodic with some fundamental frequency and harmonics of this fundamental frequency. PSS allows us to look at large-signal behaviors, such as linearity and oscillation in an oscillator, while being much faster and much easier to analyze than a transient simulation. It can also handle frequency translations, in for instance a mixer. In addition to this, we can combine the PSS simulation with small-signal analyses that uses the results from PSS as its operating point (similar to how an ac analysis uses a DC simulation as its operating point). We will see two examples of this later.

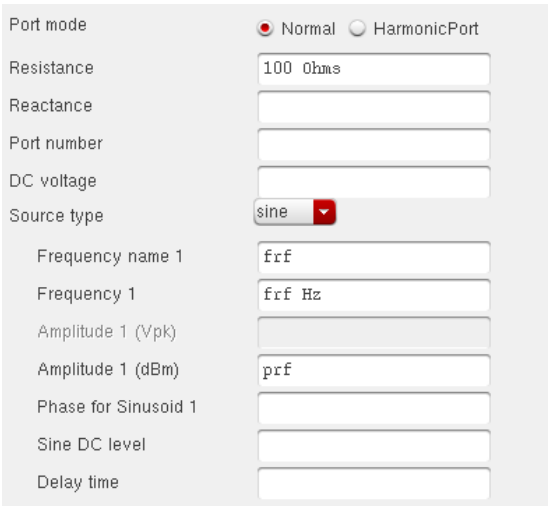

Figure 7: Port configuration for PSS analysis.

The compression point is a common measurement of the linearity of the circuit. It is the input power (in the case of input-referred compression point) at which the gain has dropped by 1 dB from a perfectly linear gain. To simulate it, open the properties of the input port. Configure it as shown in Figure 7. Import the new variables into ADE and set *frf* = 5G and *prf* = -60. Then, go to "Analyses" -> "Choose" -> PSS. Tick the box "Auto Calculate" (notice that it becomes 5GHz, since this is the only signal present), set "Number of harmonics" to 7 (this is how many overtones the PSS should consider), the "Accuracy Defaults" to conservative (the most accurate setting) and "Run transient?" to "No" (this can be necessary if the circuit has some settling behavior, for instance in an oscillator). Also tick the box "Sweep", set "Variable Name" to "prf" and sweep from -60 to 10 using a step size of 10. The result should look like Figure 8. Run the simulation and open up the "Main form"-window. Go to "pss", select "Compression Point", leave everything as default values, except set "Input Power Extrapolation Point (dBm)" -60 and select 5G as your "1st Order Harmonic). Finally, click on the output port and a P_{in} versus P_{out} plot should appear, see Figure 9. The white line corresponds to a perfectly linear system with a slope of 1 dBm/dBm and starts 1 dB below the real signal. Thus, when the lines cross, the gain has dropped by 1 dB.

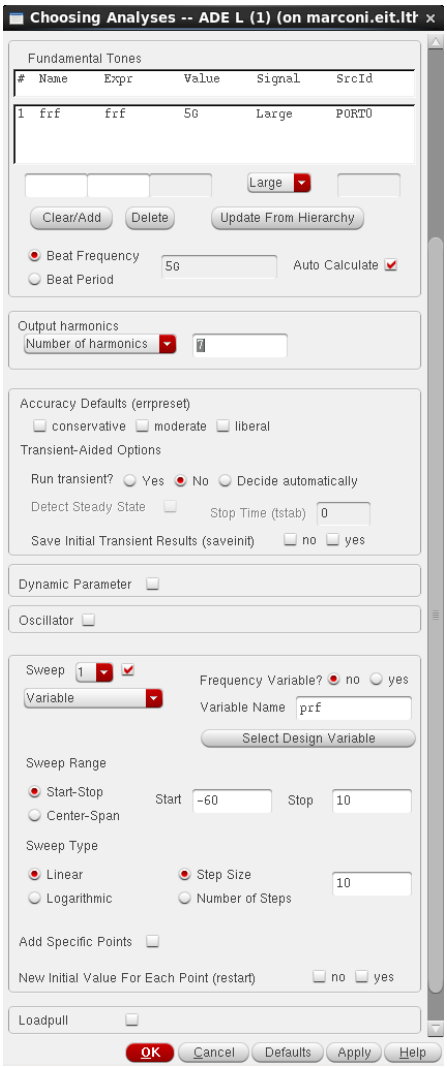

Figure 8: PSS setup for compression point simulation.

As you can see, this line is rather "choppy". To improve accuracy, we need smaller steps in the sweep. Repeat the simulation with higher resolution.

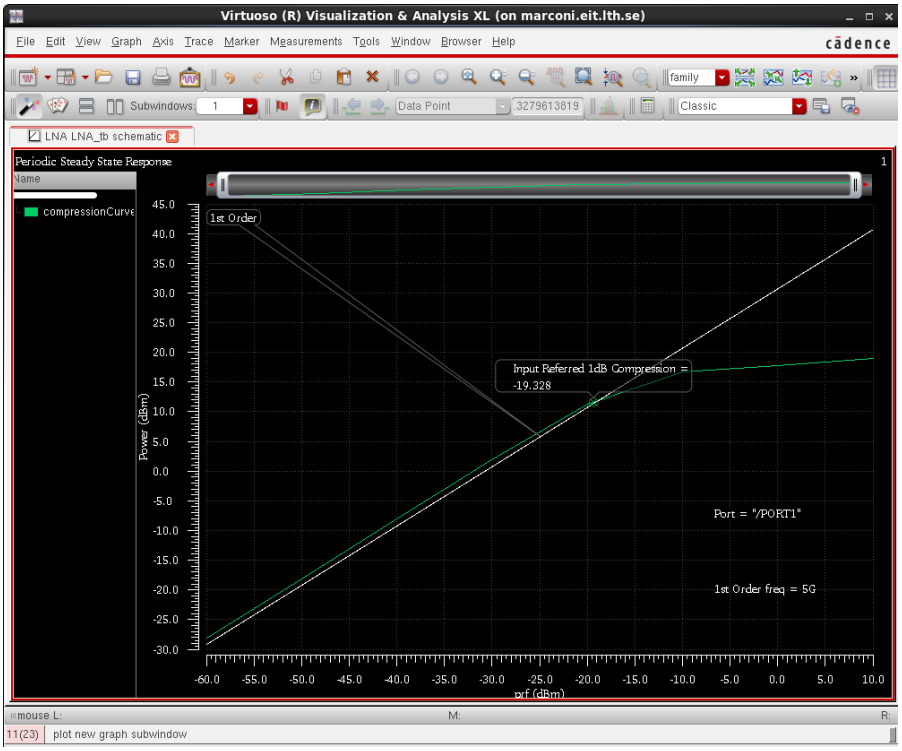

Figure 9: Result of compression point simulation.

6. IIP3 analysis using PSS and PAC

The input-referred third intercept point (IIP3) is the other common linearity measurement. Here, instead of only having one signal, we apply two signals at the input (a so-called two-tone test). If the two tones are located at f_1 and f_2, the nonlinearities in the LNA will create tones at $2f_1 - f_2$ and $2f_2 - f_1$, which will grow as the power of 3 with respect to the input power (or 3 dBm/dBm). If the two tones are close to each other, this can be very tricky to simulate using only PSS, since it would require a lot of harmonics (the fundamental tone would in this case be $f_1 - f_2$). The way to solve this is to use a periodic AC (PAC) analysis, which will use the results from PSS as its operating point.

Open the properties of PORT0 and set “PAC Magnitude (dBm)” to “prf”. If this field does not appear, tick the box “Display small signal params”. For f_1, we will use *frf* = 5GHz. f_2 we will generate using PAC[4].

Go to ADE -> ”Analyses” -> ”Choose” -> ”pac”. Change the “Sweeptype” to “relative”, “Relative Harmonic” to 1 and frequency to 20 MHz. The first harmonic of the PSS is 5 GHz, so by doing this we set the PAC frequency to f_2 = 5GHz + 20MHz = 5.02 GHz. Finally, set the “Maximum sideband” to 7 and click “OK”. Runt the simulation (make sure that PSS is also ticked). When it finishes, go to “Results”->”Direct plot”->”Main form”->”pac”. Select “IPN Curves”, change the “Circuit Input Power” to “Variable Sweep”, set “Input Extrapolation Point (dBm)” to -60 and make sure that the drop-down lists are set to “Input Referred IP3” and “3rd”. Now we just need to figure out which frequencies to choose. For the “1st Order Harmonic”, we should choose the frequency generated by the PAC, which

[4] If you are interested in why this works, you can read more here: https://designers-guide.org/analysis/intercept-point.pdf

is 5.02 GHz. The “3rd Order Harmonic” will be $2f_1 - f_2 = 10\text{GHz} - 5.02\text{ GHz} = 4.98\text{ GHz}$. Select these harmonics and click on the output port. A plot like Figure 10 should appear.

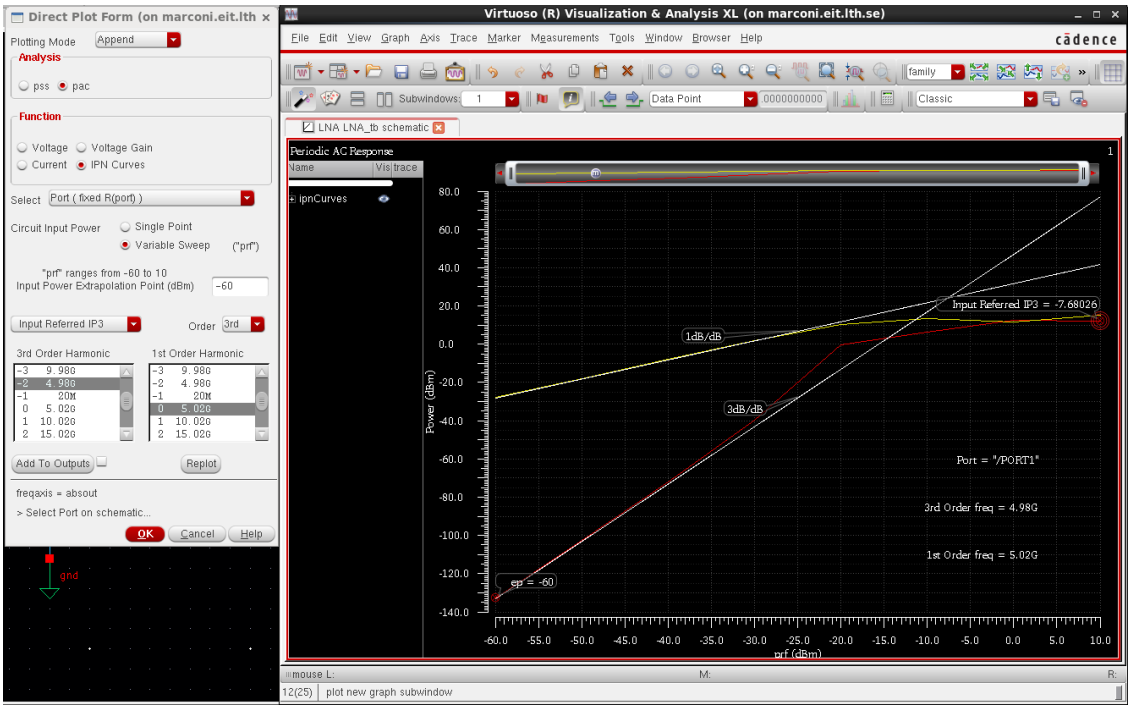

Figure 10: Result of IIP3 simulation.

7. Blocker tolerance

As a final analysis, we will simulate how a large signal affects the noise performance of the LNA. Let’s say, for instance, that we are trying to “listen” to a signal at 5.02 GHz, but at the same time a very large signal is present at 5.0 GHz. What happens then to the noise figure? To analyze this, we can use pnoise, which is a noise analysis done in the presence of large, periodic signals.

For this one, leave the PSS as it was for IIP3 (so that it sweeps prf). Go to “Analyses” -> “Choose” -> “pnoise” and change all the settings according to Figure 11. This will cause the pnoise to look at the noise at 5.02 GHz. Run the simulation and open the “Main form”-window. Go to “pnoise” -> “Noise Figure” -> “Plot”. Cadence will now plot the noise figure versus relative frequency with prf as a parameter, which is not very helpful. Righ-click on the x-axis, select “Swap Sweep Var…” and click “OK”. Now the noise figure at a relative offset of 20 MHz (i.e. 5.02 GHz) versus prf is plotted. At what input has the noise figure degraded by 1 dB? This is known as the blocker tolerance.

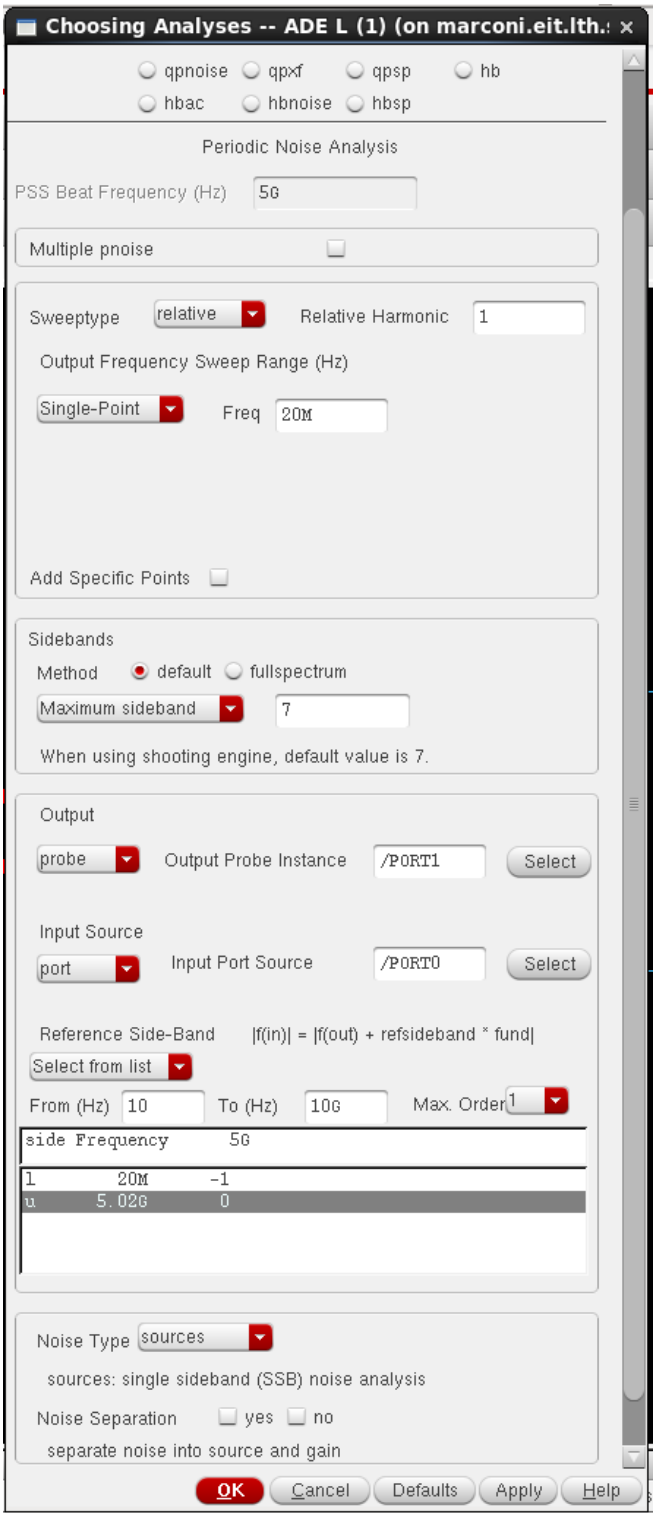

Figure 11: Setup for a blocker tolerance simulation.