TSMC PDK RF Flow Guide (IC61):

A Low-Noise Amplifier (LNA) Design Flow Example of TSMC CRN65LP Process Design Kits (PDK)

DISCLAIMER

The information contained herein is provided by TSMC on an “AS IS” basis without any warranty, and TSMC has no obligation to support or otherwise maintain the information. TSMC disclaims any representation that the information does not infringe any intellectual property rights or proprietary rights of any third parties. There are no other warranties given by TSMC, whether express, implied or statutory, including, without limitation, implied warranties of merchantability and fitness for a particular purpose.

STATEMENT OF USE

This information contains confidential and proprietary information of TSMC. No part of this information may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any human or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise, without the prior written permission of TSMC. This information was prepared for informational purpose and is for use by TSMC’s customers only. TSMC reserves the right to make changes in the information at any time and without notice.

Contents

Contents

Introduction

The major purpose of this user guide is to introduce the basic usage of a TSMC's PDK for t hose users who are completely new to TSMC PDK or never use TSMC's PDKs before as a reference. To ease the overall introduction, we use a simple LNA design as an example to go through the whole design flow: starting from the schematic capture and ending at the physical verification and post-layout simulation. We divide the whole flow into several phases below:

Schematic Capture

- Environment setup
- Creating a library, design, symbol and test fixture

Pre-layout Simulation

- Using Spectre simulator
- LNA performance

Layout Creation

- Schematic-driven-layout
- Components placement
- Manual routing

Physical Verification

- Assura flow

Post-layout Simulation

- Assura flow

Contents

Schematic Capture

After you have finished the installation of the TSMC's PDK, we will start to create a new design based on the installed PDK.

- **Environment setup**
- **Creating a library**
- **Creating a design**
- **Creating a symbol**
- **Creating a test fixture**

Environment setup

Before we start to create a new design, some environment setups should be done. First, we have to set the environment variable of “CDS_Netlisting_Mode” to “Analog”. This can be archived by the following UNIX command:

```
setenv CDS_Netlisting_Mode   “Analog”
```

Then, go to the demo directory and enter Cadence environment by:

```
%cd  <pdk_install_directory>/RF_flow
%virtuoso &
```

Note:

1) The installation procedures of the TSMC’s PDK can be found in the document of “TSMC PDK reference manual” released along with the corresponding PDK.
2) The <pdk_install_directory> is referred to the path where the TSMC’s PDK was installed.

Creating a library

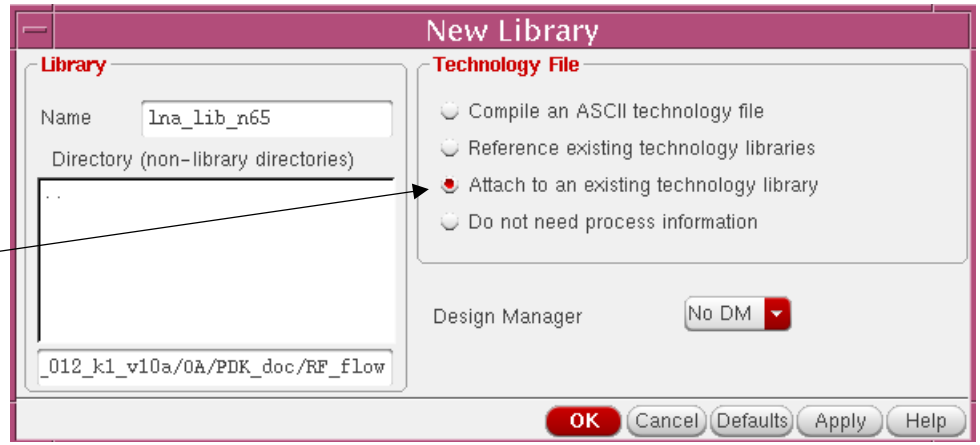
After completing the environment setup, we can start to create a new library.

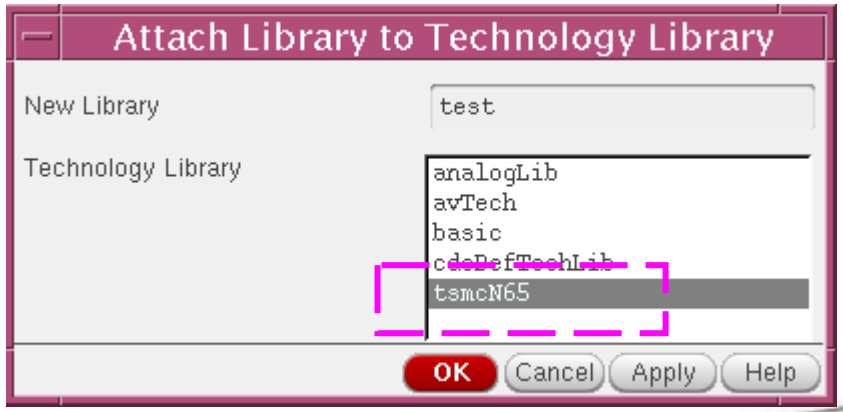
1) **In the CIW, select “File->New->Library**

2) **In the Attach Design Library to Technology File form, select “tsmcN90rf”, then click OK.**

Enter the new library name into the Name field.

Select “Attach to an existing techfile”





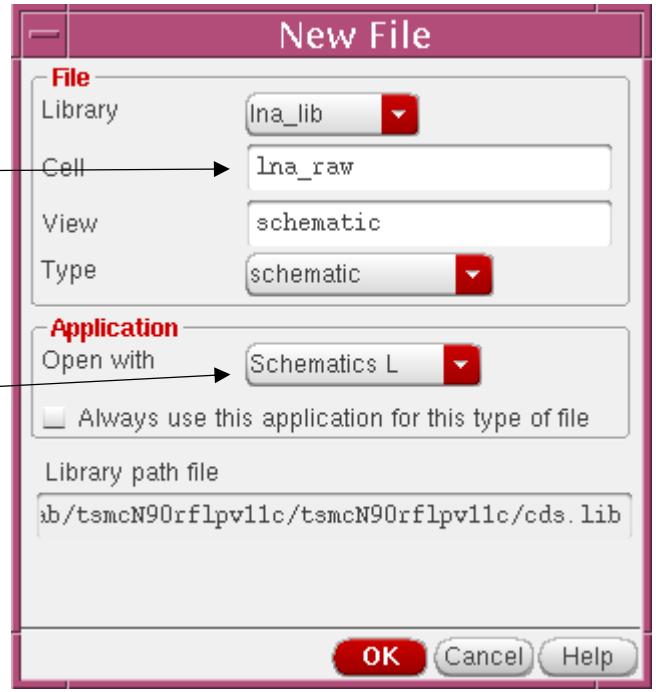
Creating a design

Creating a Schematic Cellview

1) In the CIW or Library Manager, select File->New->Cellview
2) Set up the Create New File as follows:
3) Click OK when done.

Enter the new cell name into the Name field.

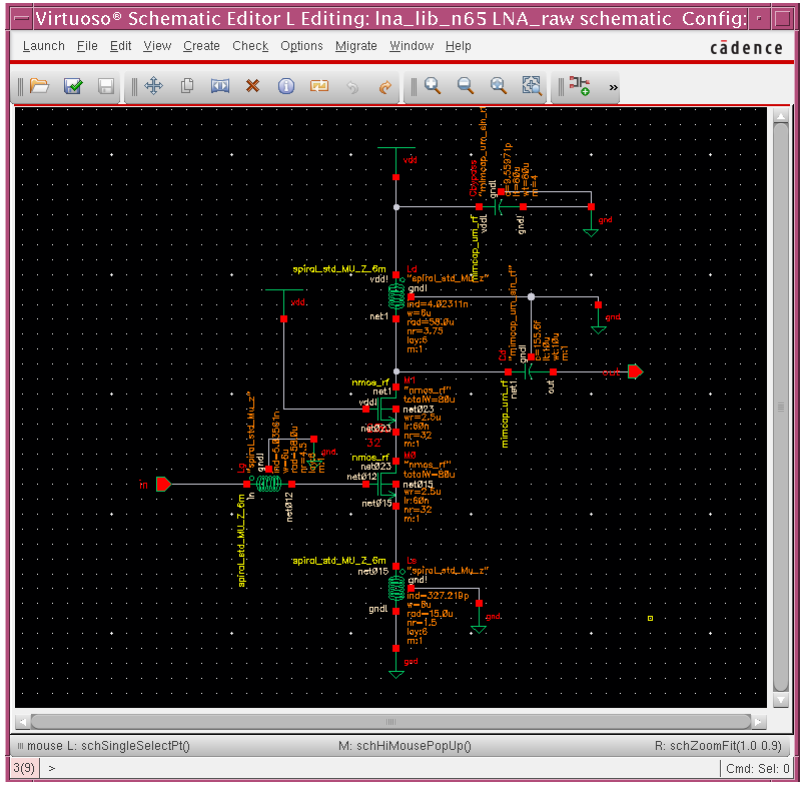
Select “Composer-Schematic”



Creating a design

Adding Components to a Schematic

Build the lna_raw schematic shown below:



1. In the *LNA_raw* schematic window, click the Instance fixed menu icon to display the Add Instance form.
2. Make sure that the View Name field in the form is set to symbol. You will update the Library Name, Cell Name, and the property values given in the table as you place each component.
3. After you complete the Add Instance form, move your cursor to the schematic window and click left to place a component.

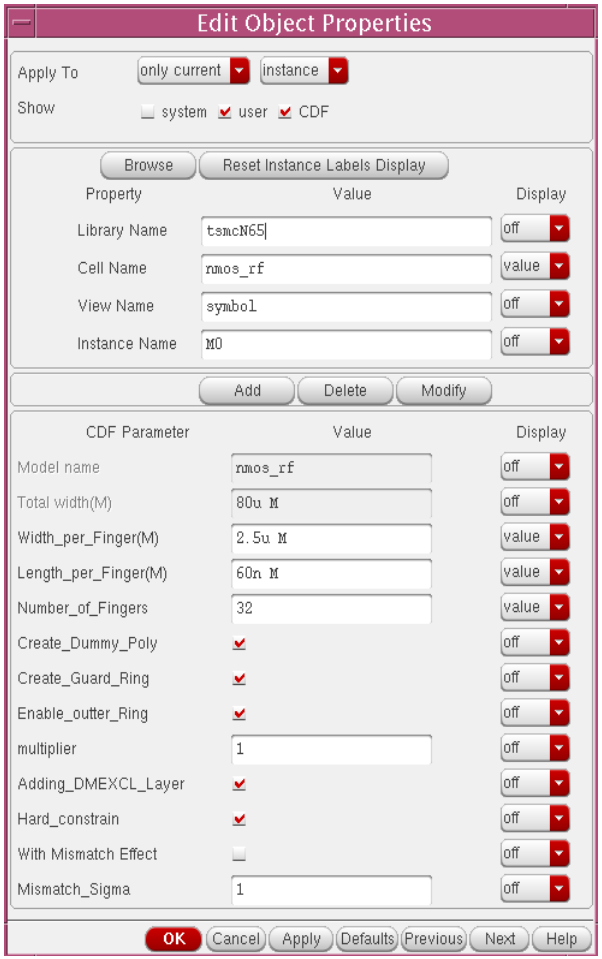
   Another way to fill in the Add Instance form is to click on the Browse button. This button opens up a Library Browser from which you can select components to place your left mouse button.

Creating a design

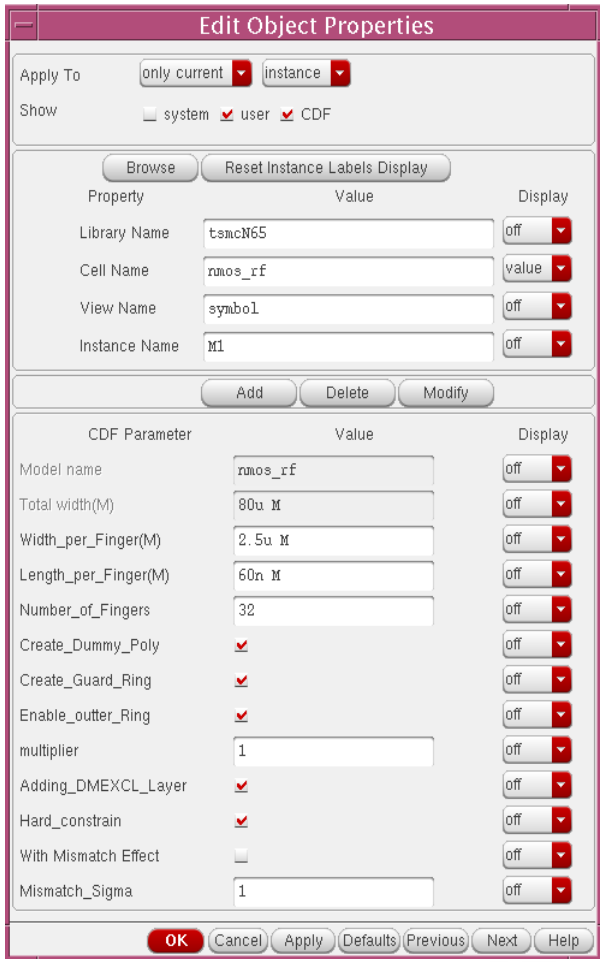
Component	Library Name	Cell Name	Model Name	Properties
M0	tsmcN65	nmos_rf	nmos_rf	Width_per_Finger=2.5um Length_per_Finger=60nm Number_of_Fingers=32
M1	tsmcN65	nmos_rf	nmos_rf	Width_per_Finger=2.5um Length_per_Finger=60nm Number_of_Fingers=32
Ls	tsmcN65	spiral_std_MU_Z	spiral_std_Mu_z	Inductor_Width=6um Inner_Radius=15um Number_of_Turns=1.5 Guard_Ring_Distances=50um
Lg	tsmcN65	spiral_std_MU_Z	spiral_std_Mu_z	Inductor_Width=6um Inner_Radius=58um Number_of_Turns=4.5 Guard_Ring_Distances=50um
Ld	tsmcN65	spiral_std_MU_Z	spiral_std_Mu_z	Inductor_Width=6um Inner_Radius=58um Number_of_Turns=3.75 Guard_Ring_Distances=50um
Cd	tsmcN65	mimcap_um_rf	mimcap_um_sin_rf	Length=10um Width=10um
Cbypass	tsmcN65	mimcap_um_rf	mimcap_um_sin_rf	Length=80um Width=80um Multiplier=4

Creating a design

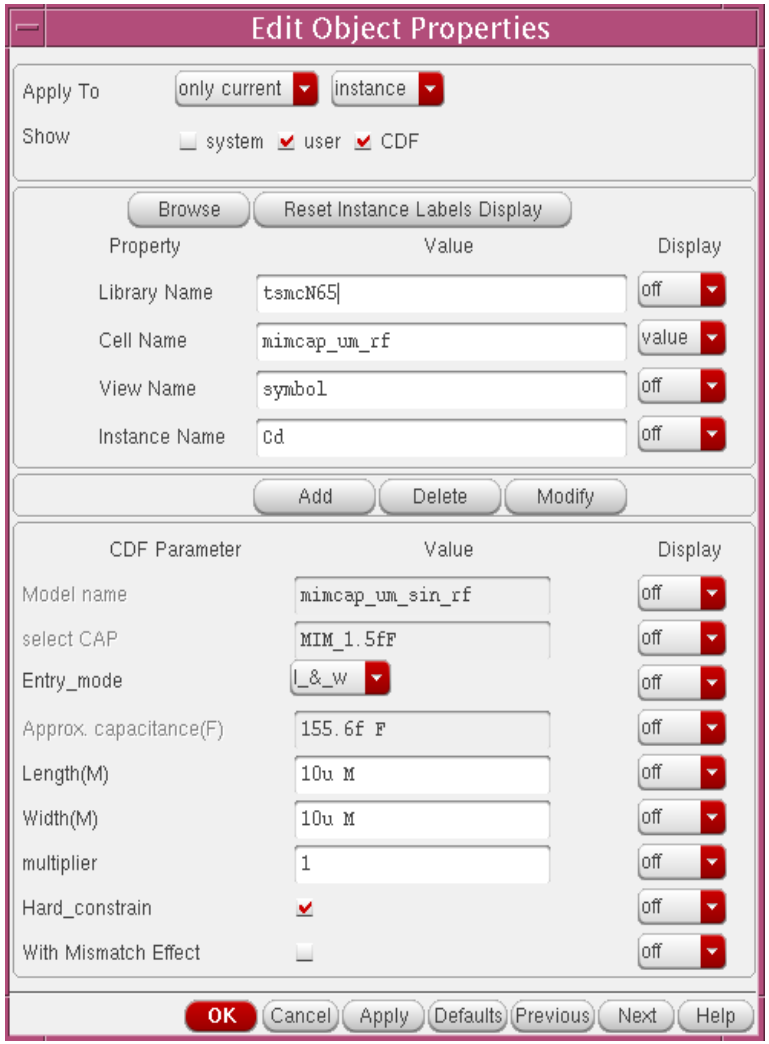
If you place a component with wrong parameter values, you can do the following steps to change the parameters. Edit->Properties->Objects .The properties of M0,M1,Cd,Cbypass,Ls,Lg and Ld should be made sure the same as follows:

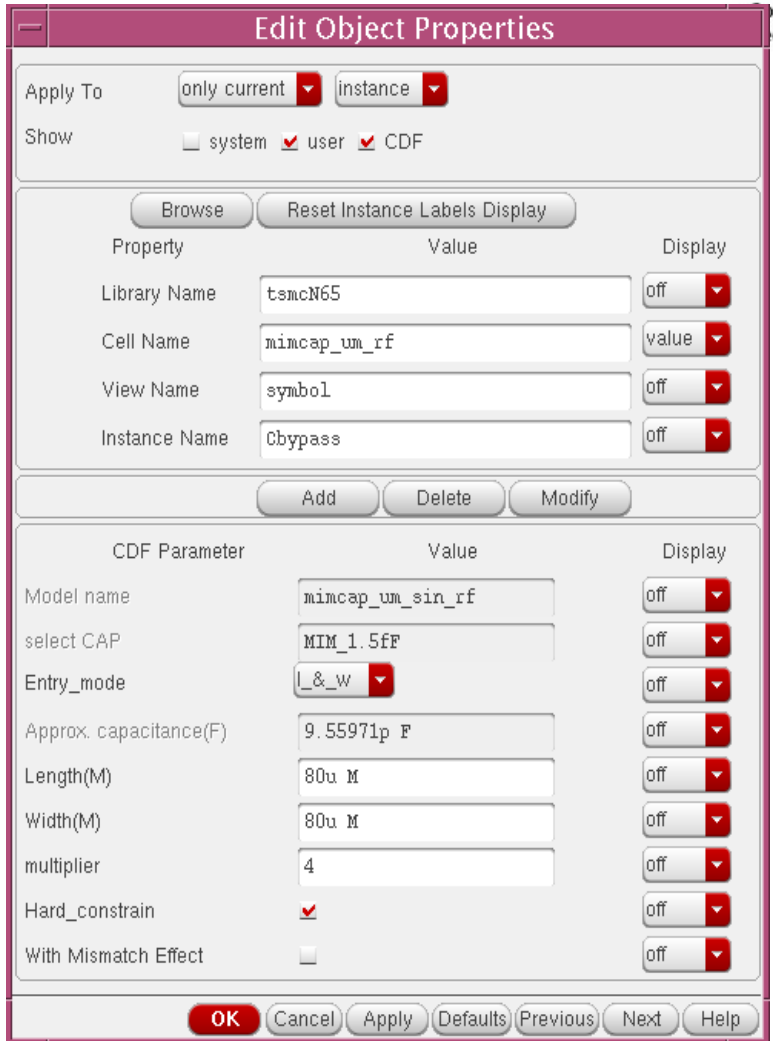


M0



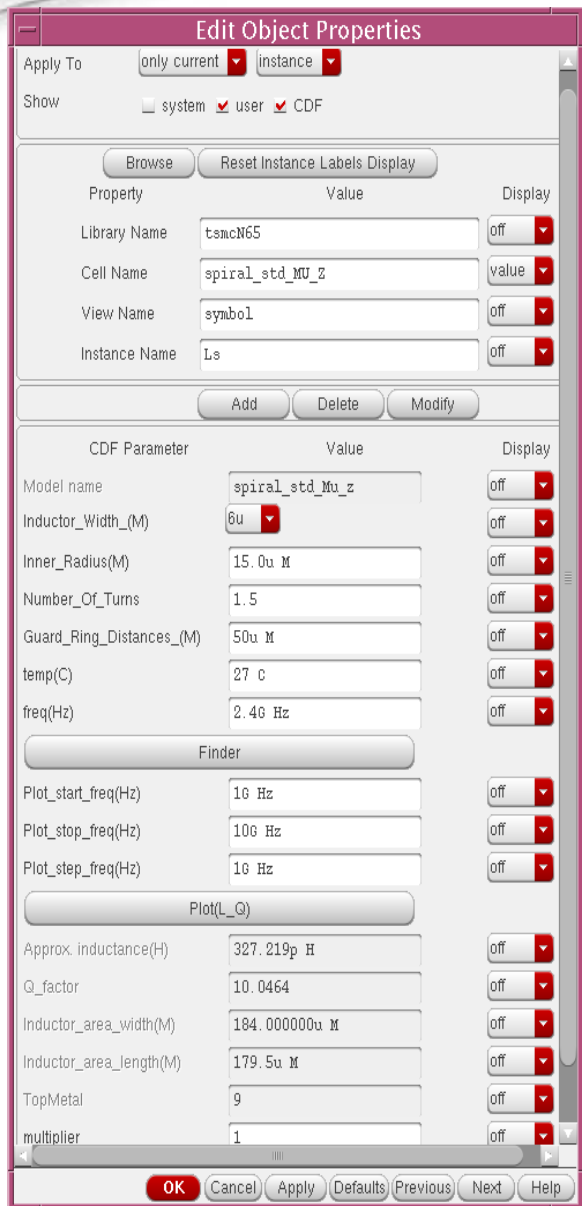
M1



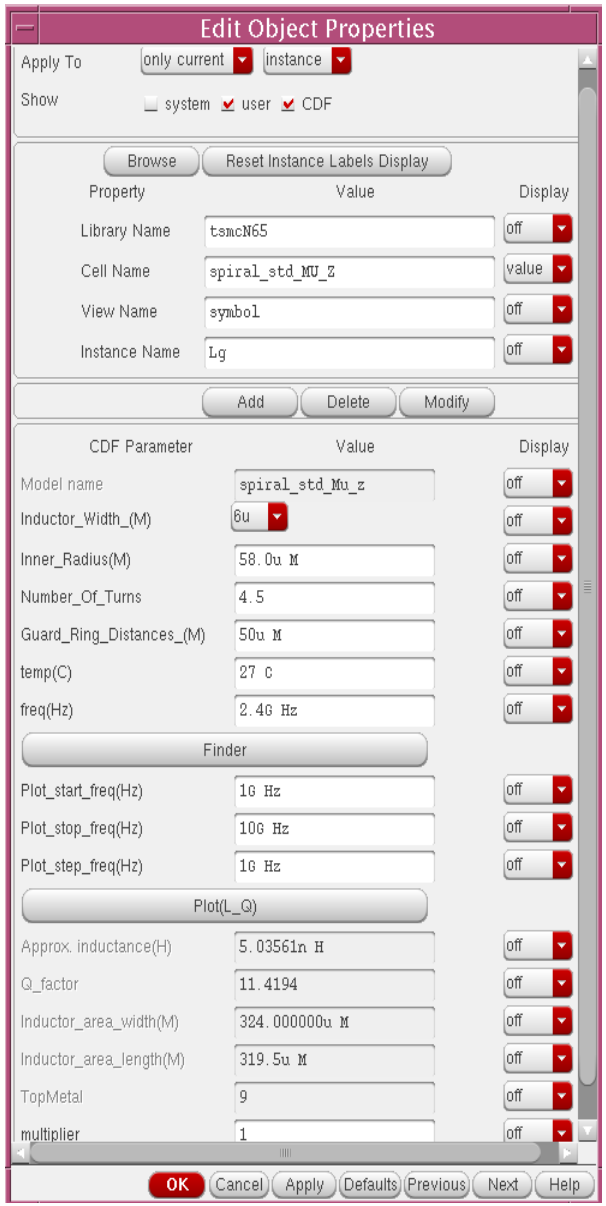


Cd

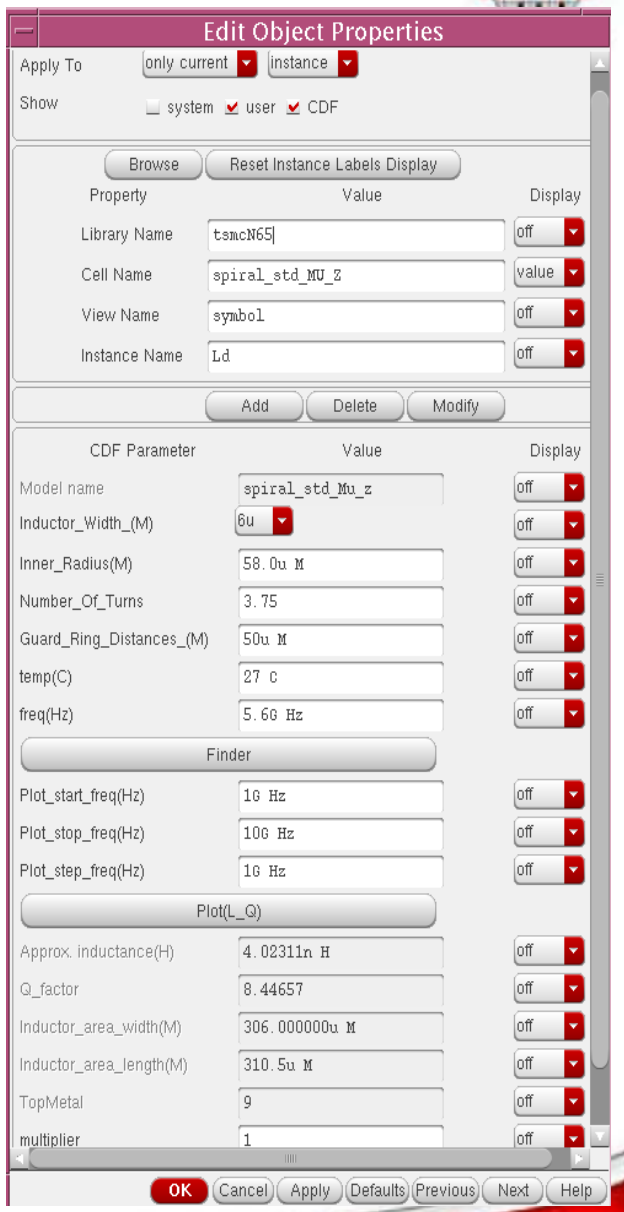
Cbypass



Ls



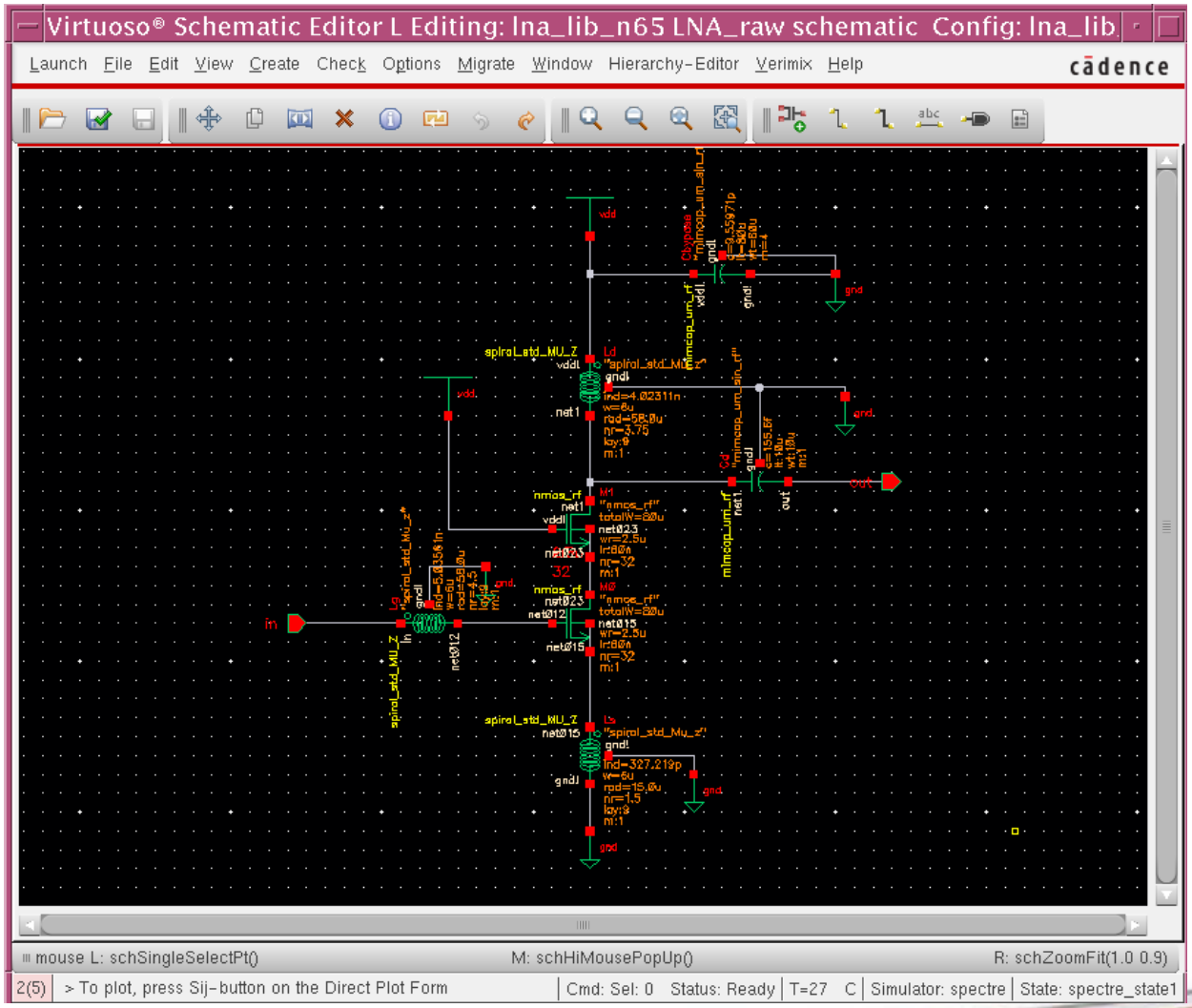
Lg



Ld

Creating a design

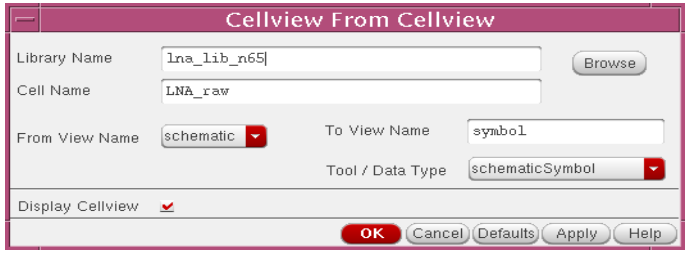
After placing this components mentioned above, finally add vdd and gnd from “analogLib” and do wires and pins. Your schematic should be like this.

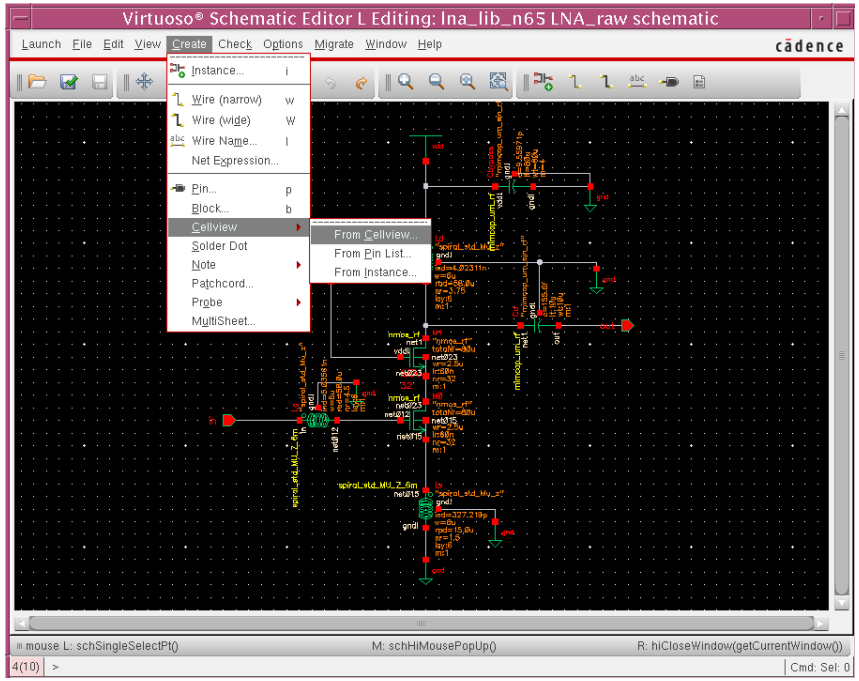


Creating a symbol

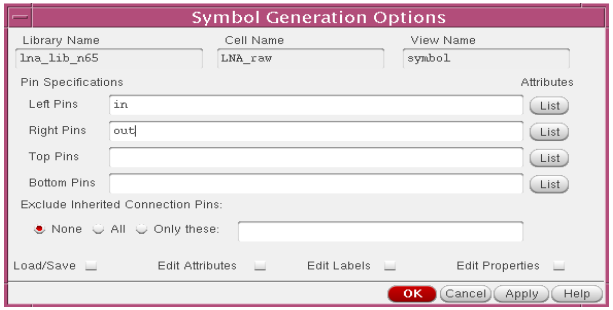
After completing the creation of schematic-capture, we need to create a corresponding symbol for the subsequent simulation steps.

1. **In the LNA_raw schematic window, select Create-> Cellview->From Cellview**
2. **Setup the Cellview From Cellview window as follows:**

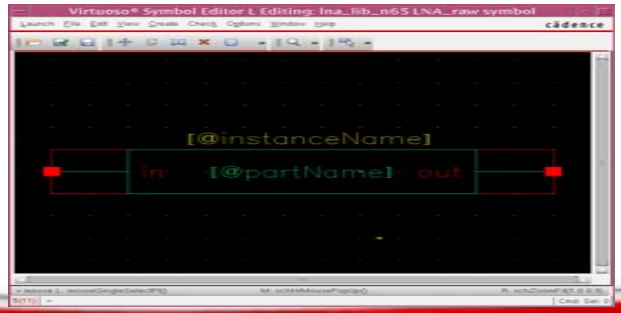




3. **Edit the Pins according to the following figure.**



4. **Click “OK” and then the symbol view is created.**

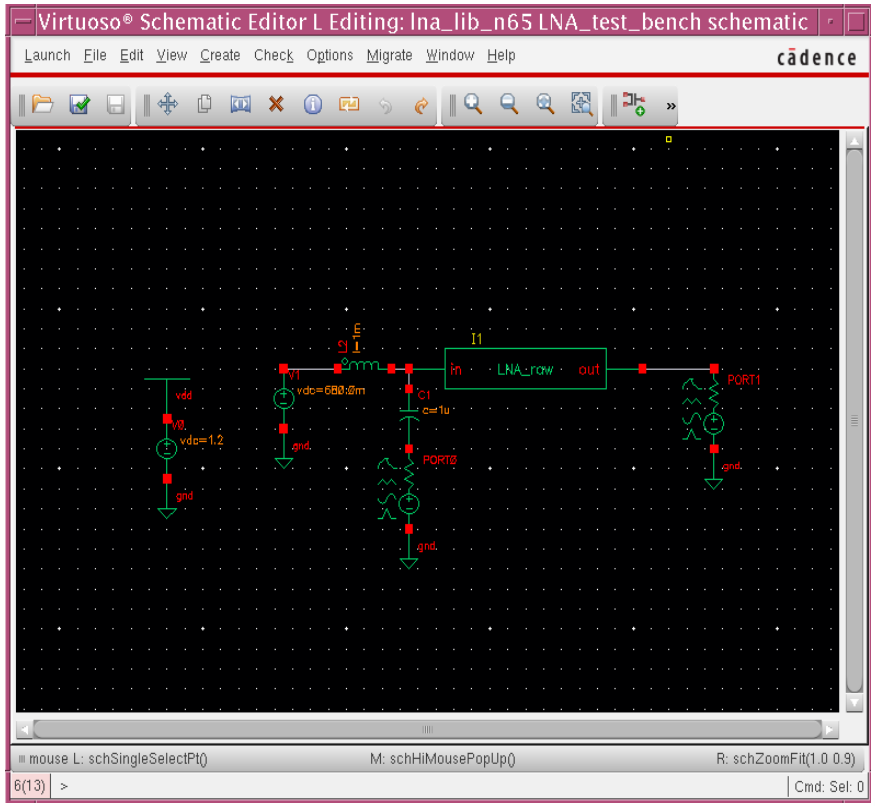


Creating a test fixture

The final step before we start the simulation is to create a test fixture for our design. The creation of test fixture is similar to the creation of a design. Furthermore, you also have to prepare the component table shown below to build the test fixture schematic. Generally, a test fixture will consist of the following components: a core design (the LNA_raw in our case), DC voltage source, ground, vdd, port, DC blocking capacitor and RF choke inductor. The test fixture that we used for our design namely "LNA_test_bench" is shown below.

Library Name	Cell Name	Properties/comments
analogLib	vdd	--
analogLib	gnd	--
analogLib	vdc	For vin: DC voltage=0.68v
analogLib	vdc	For vdd:DC voltage=1.2v
analogLib	Port	For PORT0: Resistance=50 ohm
analogLib	port	For PORT1: Resistance=50 ohm
analogLib	ind	For L2: L=1m H
analogLib	cap	For C1: C=1u F
lna_lib_n65	LNA_raw	Core design

Component table



LNA Test bench

Contents

Pre-layout Simulation

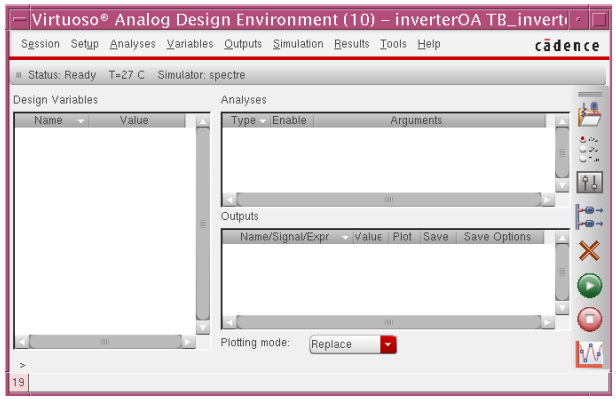
After completing the creation of test fixture schematic-capture, we need to run simulation to check its function and performance. In this chapter, we will use Spectre as the simulator.

- **Using Spectre simulator**
- **LNA Performance**
- **Corner simulation**

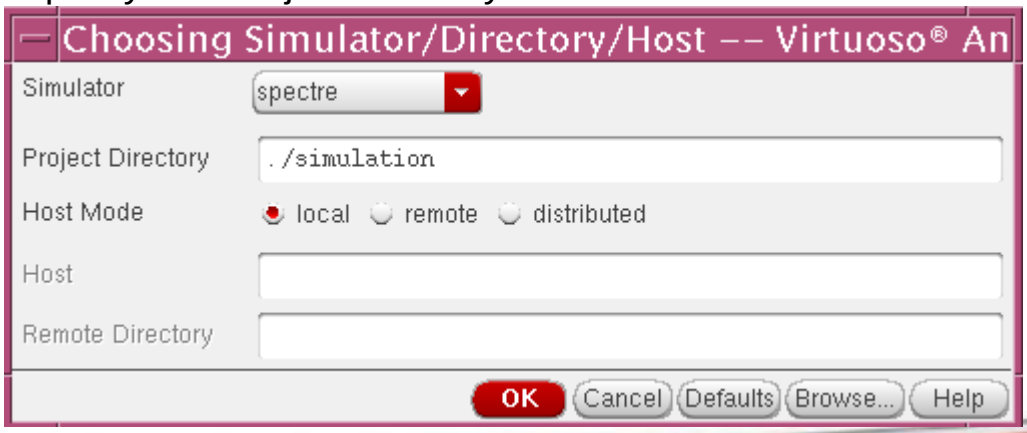
Using Spectre Simulator

In the section, we will start to run simulation. We use “Spectre” as our simulator in ADE (analog design environment). And the following is our steps.

1) In the schematic window, select “Launch->ADEL”. And then, the ADE window will pop up.



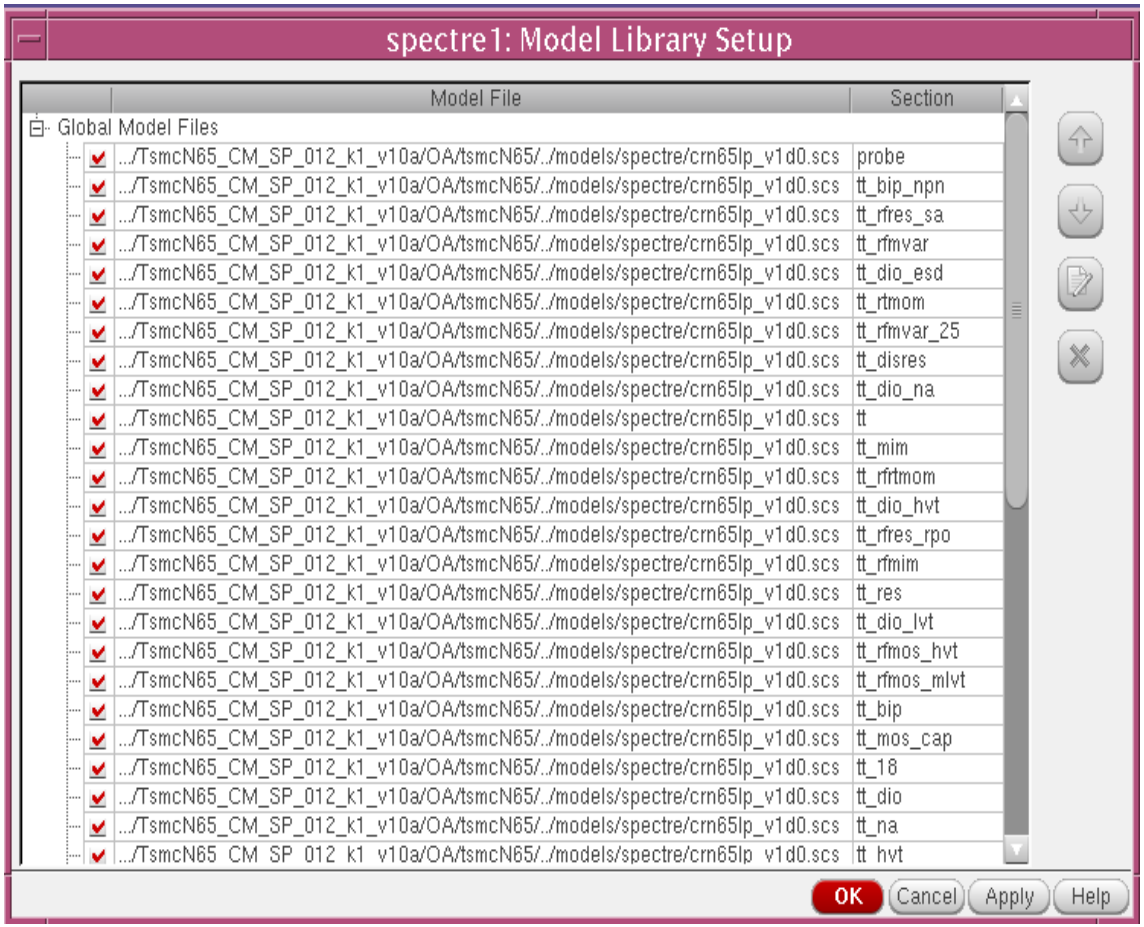
2) Select simulator
Setup->simulator/Director/Host and set simulator to “spectre” and specify the “Project Directory” to “./simulation”.



Using Spectre Simulator

3) Model library setup

In the ADEL menu bar, to pull down “Setup” and choose “Model Libraries”.



4) Select analysis type and fill in parameters for simulation

In the Analog Design Environment (ADE), there are many analysis options that you can choose. Since we want to **analysis the S-parameter** of our design, we choose the sp analysis for our design. Some designers may want to see the OP point and they can also include **the DC op point** analysis.

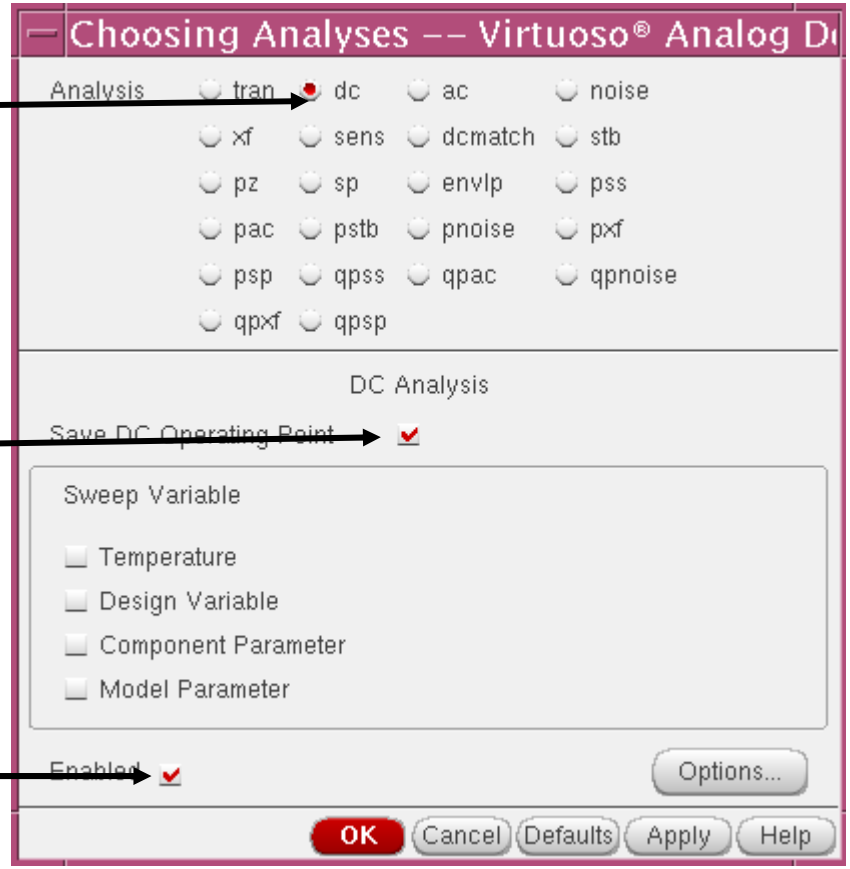
Using Spectre Simulator (cont.)

5) Setup of DC analysis

In the Analysis section, select *dc*

Turn on the “Save DC Operating point” field

Turn on the Enabled field



Using Spectre Simulator (cont.)

6) Setup of sp analysis

In the Analysis section, select ***sp.***

In the S-parameter section, click on the Select button, and select the ports of interest in schematic: **PORT0 and PORT1 are selected in this case.**

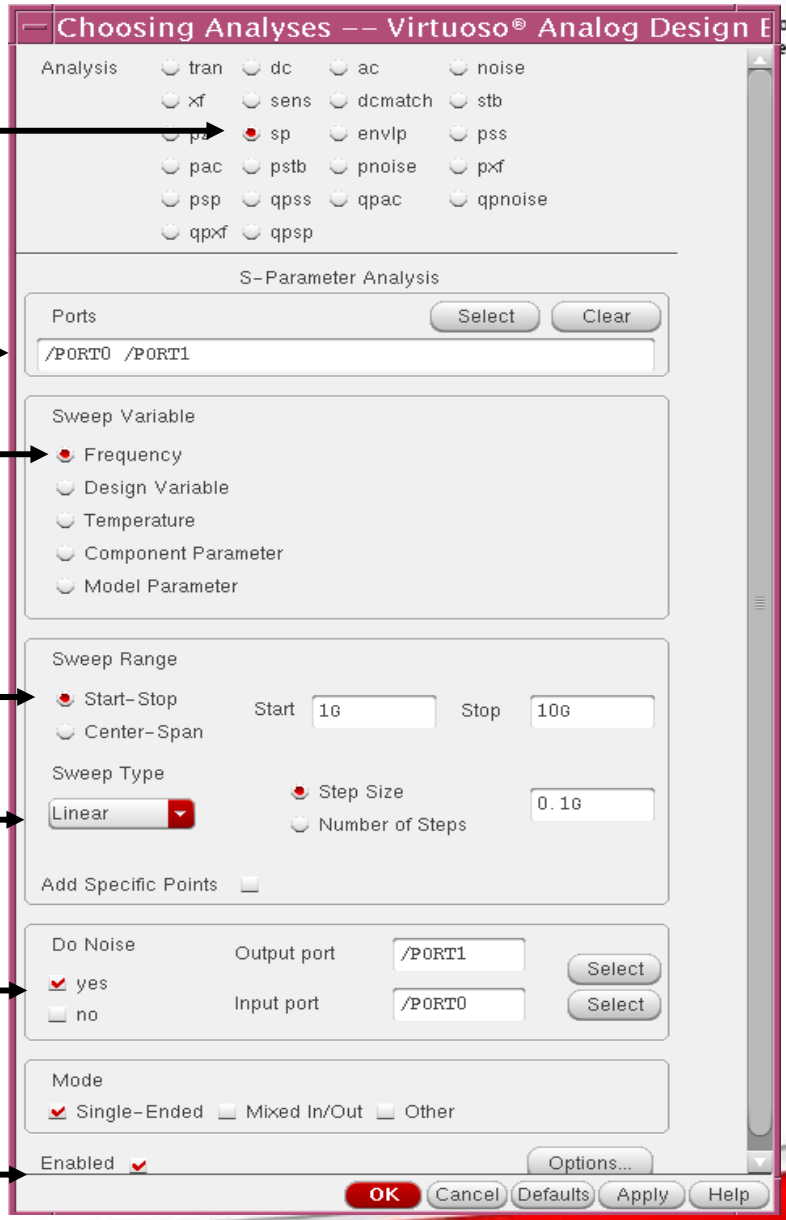
Set the Sweep Variable to **Frequency**

Set Start to 1G and stop to 10G

Set the sweep type to **Linear** with Step Size set to **0.1G**

Set **yes** to do noise simulation and select **PORT1** as output port and **PORT0** as input port in this case

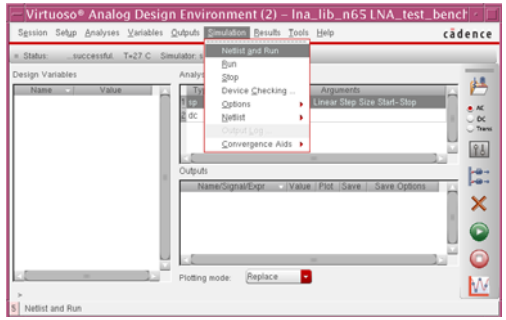
Turn on the **Enabled** field



Using Spectre Simulator (cont.)

7) Run simulation

To start the simulation, you can click “simulation-> netlist and run” from the Analog Design Environment (ADE) menu bar.



/home/lhtung/PDKQA/TsmcN65_CM_SP_012_k

File Help cadence

```
// Generated for: spectre
// Generated on: Apr 19 18:33:39 2007
// Design library name: lna_lib_n65
// Design cell name: LNA_test_bench
// Design view name: config
simulator lang=spectre
global 0 vdd!
include "/home/lhtung/PDKQA/TsmcN65_CM_SP_012_k1_v10a/0A/tsmcN65/../m
include "/home/lhtung/PDKQA/TsmcN65_CM_SP_012_k1_v10a/0A/tsmcN65/../m
include "/home/lhtung/PDKQA/TsmcN65_CM_SP_012_k1_v10a/0A/tsmcN65/../m

// Library name: lna_lib_n65
// Cell name: LNA_raw
// View name: schematic
// Inherited view list: spectre cmos_sch cmos.sch schematic veriloga
subckt LNA_raw in out
    Cbypass (vdd! 0 0) mimcap_um_sin_rf lt=80u wt=80u m=4 mimflag=2 \
        mismatchflag=0
    Cd (net1 out 0) mimcap_um_sin_rf lt=10u wt=10u m=1 mimflag=2 \
        mismatchflag=0
    M0 (net023 net012 net015 net015) nmos_rf lr=60n wr=2.5u nr=32 sig
        m=1 mismatchflag=0
    M1 (net1 vdd! net023 net023) nmos_rf lr=60n wr=2.5u nr=32 sigma=1
        mismatchflag=0
    Lg (in net012 0) spiral_std_Mu_z w=6u rad=58.0u nr=4.5 lay=6 gdis
        m=1
    Ls (net015 0 0) spiral_std_Mu_z w=6u rad=15.0u nr=1.5 lay=6 gdis=
        m=1
    Ld (vdd! net1 0) spiral_std_Mu_z w=6u rad=58.0u nr=3.75 lay=6 gdi
        m=1
ends LNA_raw
// End of subcircuit definition.

// Library name: lna_lib_n65
// Cell name: LNA_test_bench
// View name: schematic
// Inherited view list: spectre cmos_sch cmos.sch schematic veriloga
I1 (net9 net5) LNA_raw
V1 (net3 0) vsource dc=680.0m type=dc
V0 (vdd! 0) vsource dc=1.2 type=dc
PORT0 (net11 0) port r=50 type=sine
PORT1 (net5 0) port r=50 type=sine
L2 (net3 net9) inductor l=lm
```

6

/home/lhtung/PDKQA/TsmcN65_CM_SP_012_k

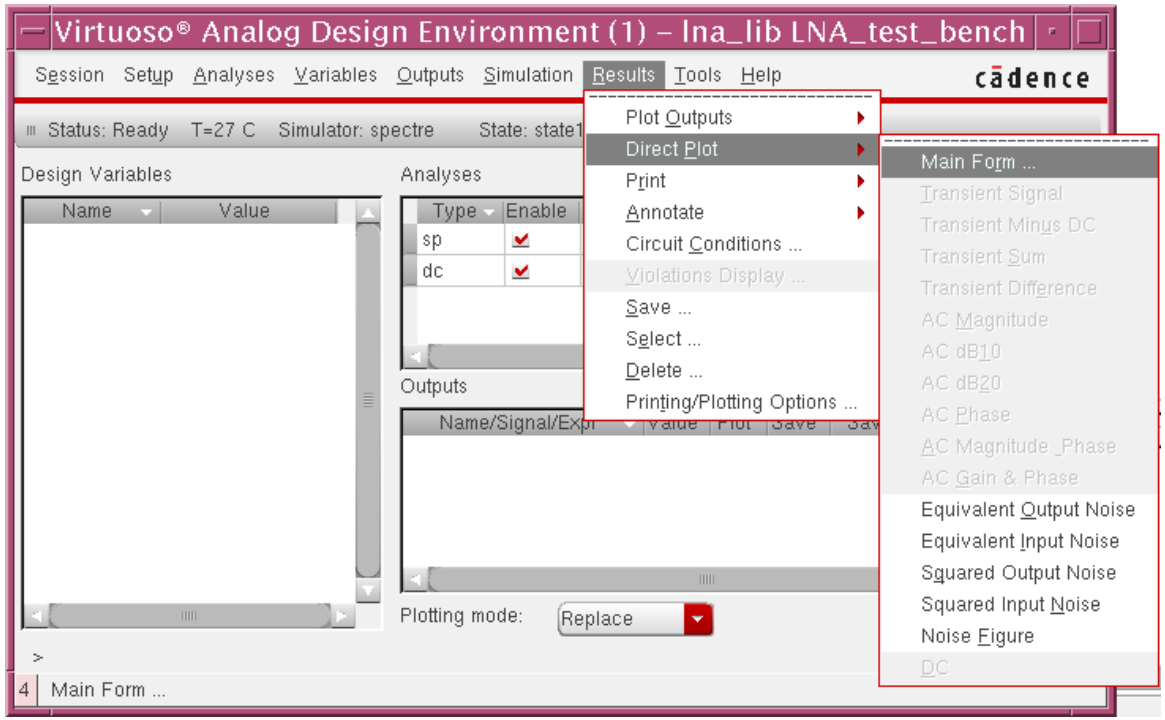
File Help cadence

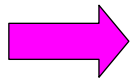
```
Convergence achieved in 5 iterations.
Accumulated dc load time = 0 s.
Accumulated dc factor time = 0 s.
Accumulated dc solve time = 0 s.
Accumulated dc output time = 0 s.
Total time required for dc analysis `dcOp' was 10 ms.

dcOpInfo: writing operating point information to rawfile.

*************************************************
S-Parameter Analysis `sp': freq = (1 GHz -> 10 GHz)
*************************************************
    sp: freq = 1.3 GHz      (3.33 %), step = 100 MHz      (1.11 %)
    sp: freq = 1.7 GHz      (7.78 %), step = 100 MHz      (1.11 %)
    sp: freq = 2.2 GHz      (13.3 %), step = 100 MHz      (1.11 %)
    sp: freq = 2.6 GHz      (17.8 %), step = 100 MHz      (1.11 %)
    sp: freq = 3.1 GHz      (23.3 %), step = 100 MHz      (1.11 %)
    sp: freq = 3.5 GHz      (27.8 %), step = 100 MHz      (1.11 %)
    sp: freq = 4 GHz        (33.3 %), step = 100 MHz      (1.11 %)
    sp: freq = 4.4 GHz      (37.8 %), step = 100 MHz      (1.11 %)
    sp: freq = 4.9 GHz      (43.3 %), step = 100 MHz      (1.11 %)
    sp: freq = 5.3 GHz      (47.8 %), step = 100 MHz      (1.11 %)
    sp: freq = 5.8 GHz      (53.3 %), step = 100 MHz      (1.11 %)
    sp: freq = 6.2 GHz      (57.8 %), step = 100 MHz      (1.11 %)
    sp: freq = 6.7 GHz      (63.3 %), step = 100 MHz      (1.11 %)
    sp: freq = 7.1 GHz      (67.8 %), step = 100 MHz      (1.11 %)
    sp: freq = 7.6 GHz      (73.3 %), step = 100 MHz      (1.11 %)
    sp: freq = 8 GHz        (77.8 %), step = 100 MHz      (1.11 %)
    sp: freq = 8.5 GHz      (83.3 %), step = 100 MHz      (1.11 %)
    sp: freq = 8.9 GHz      (87.8 %), step = 100 MHz      (1.11 %)
    sp: freq = 9.4 GHz      (93.3 %), step = 100 MHz      (1.11 %)
    sp: freq = 9.8 GHz      (97.8 %), step = 100 MHz      (1.11 %)
Accumulated DC solution time = 0 s.
Intrinsic sp analysis time = 40 ms.
Total time required for sp analysis `sp' was 40 ms.

modelParameter: writing model parameter values to rawfile.
element: writing instance parameter values to rawfile.
outputParameter: writing output parameter values to rawfile.
designParamVals: writing netlist parameters to rawfile.
primitives: writing primitives to rawfile.
subckts: writing subcircuits to rawfile.
```

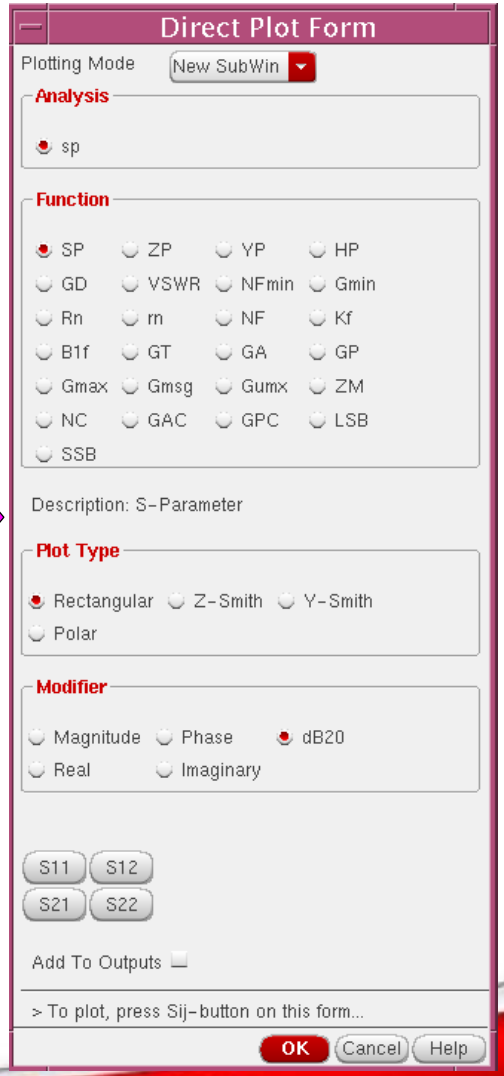
7

LNA Performance

After running simulation, we can see the result and performance of our design from Results->Direct Plot->Main Form…:

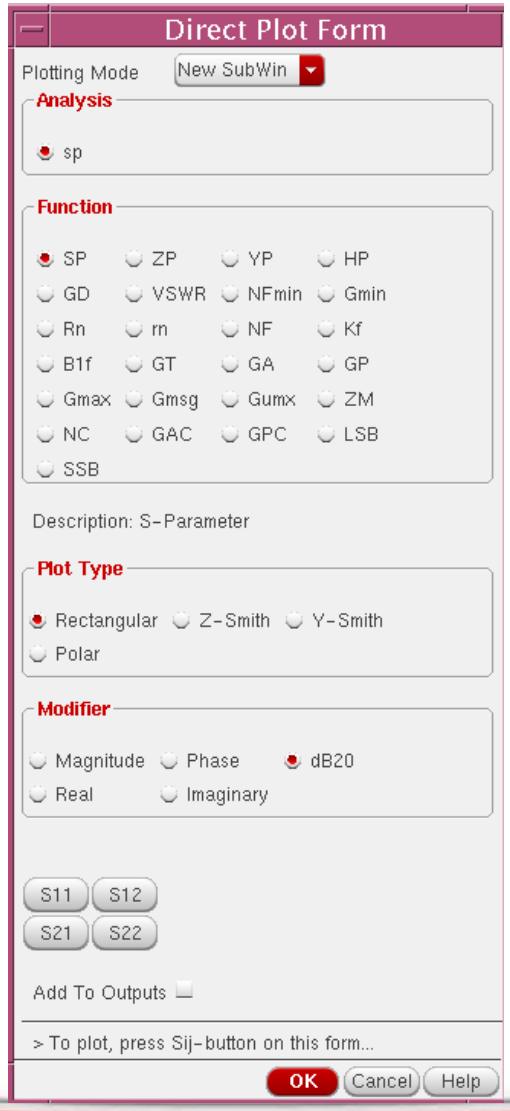


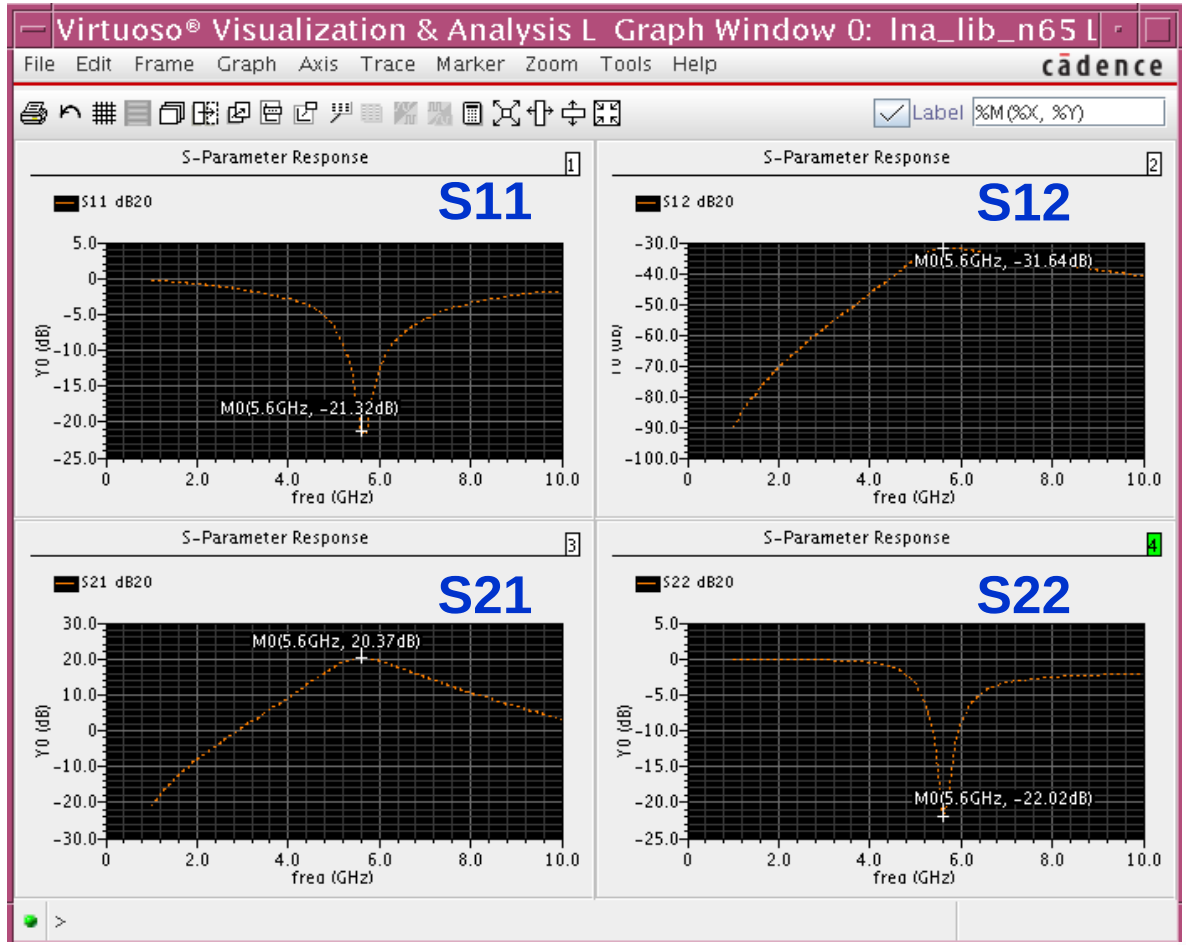




LNA Performance

By setting Direct Plot Form shown below, we can see the dB format S-parameter of our LNA design. Click on "S11/S12/S21/S22" to analyze S-parameter. And then, you can choose "trace->Delta cursor" to find out the 3dB Gain bandwidth. Finally, select "session->save state (state1) -> ok"

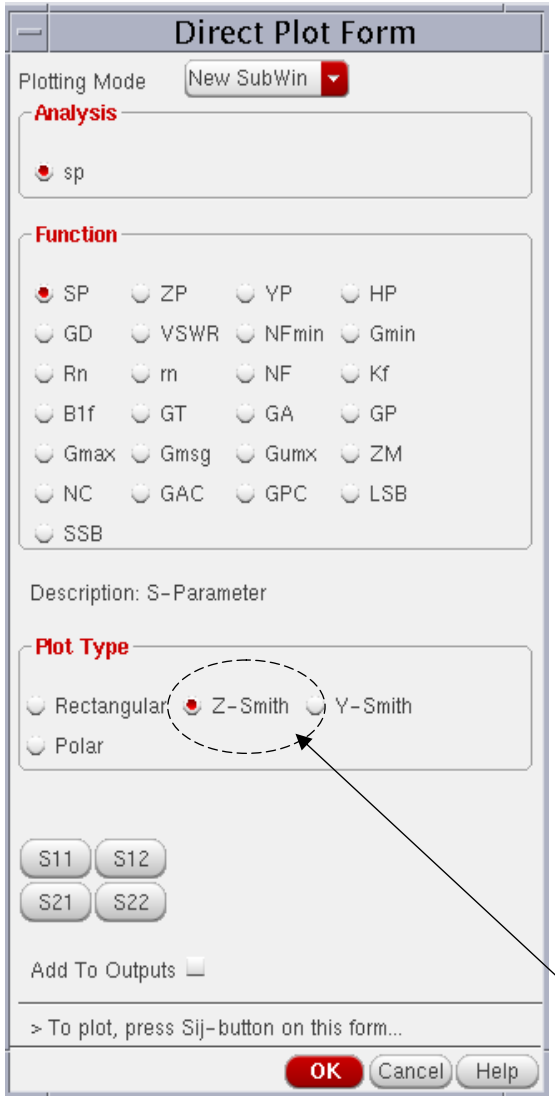


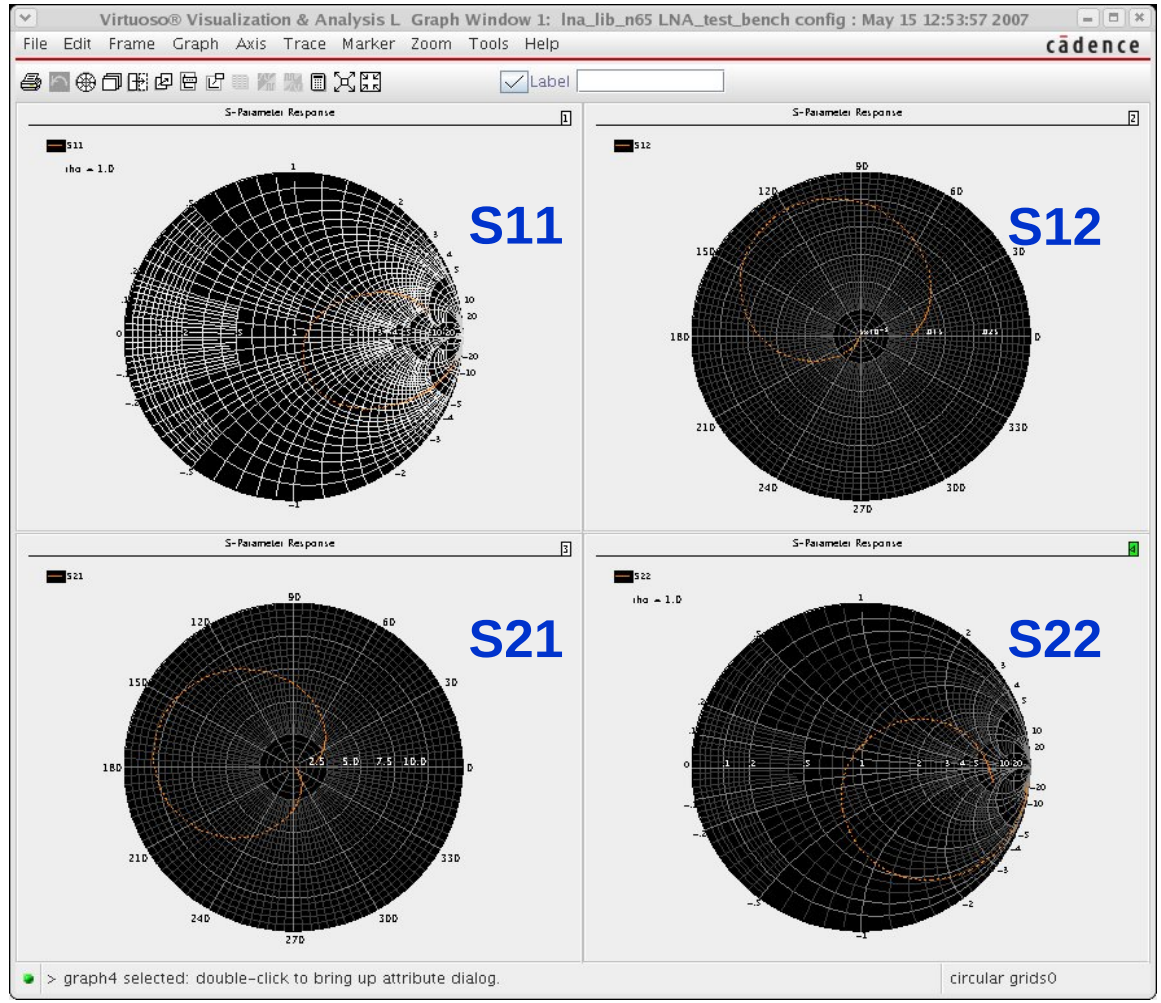


S11=-21.32 dB@5.6GHz; S12=-31.64 dB@5.6GHz
S21=20.37 dB@5.6GHz; S22=-22.02 dB@5.6GHz

LNA Performance

By setting Direct Plot Form shown below, we can see the S-parameter of our LNA design.

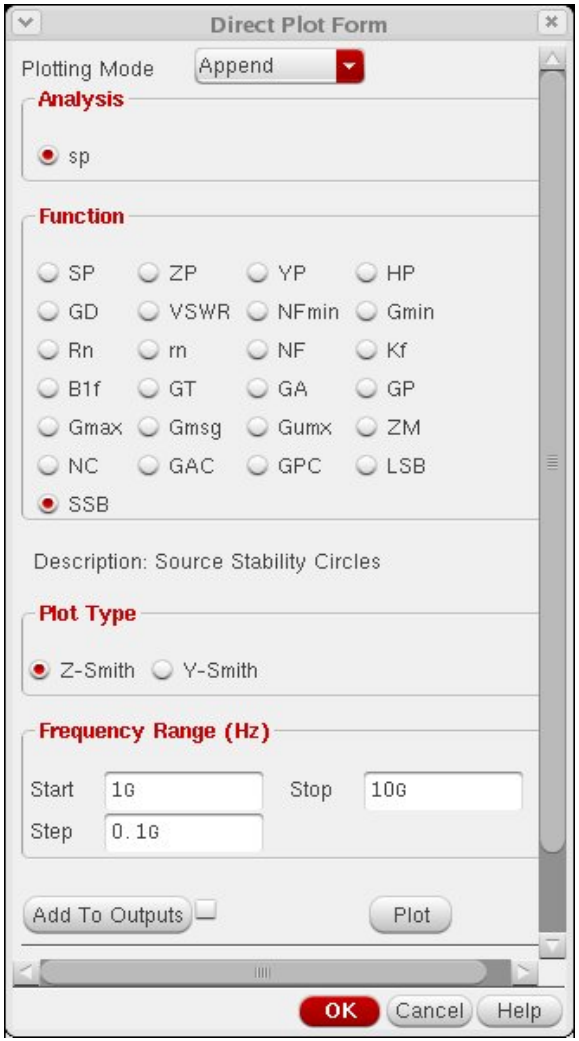


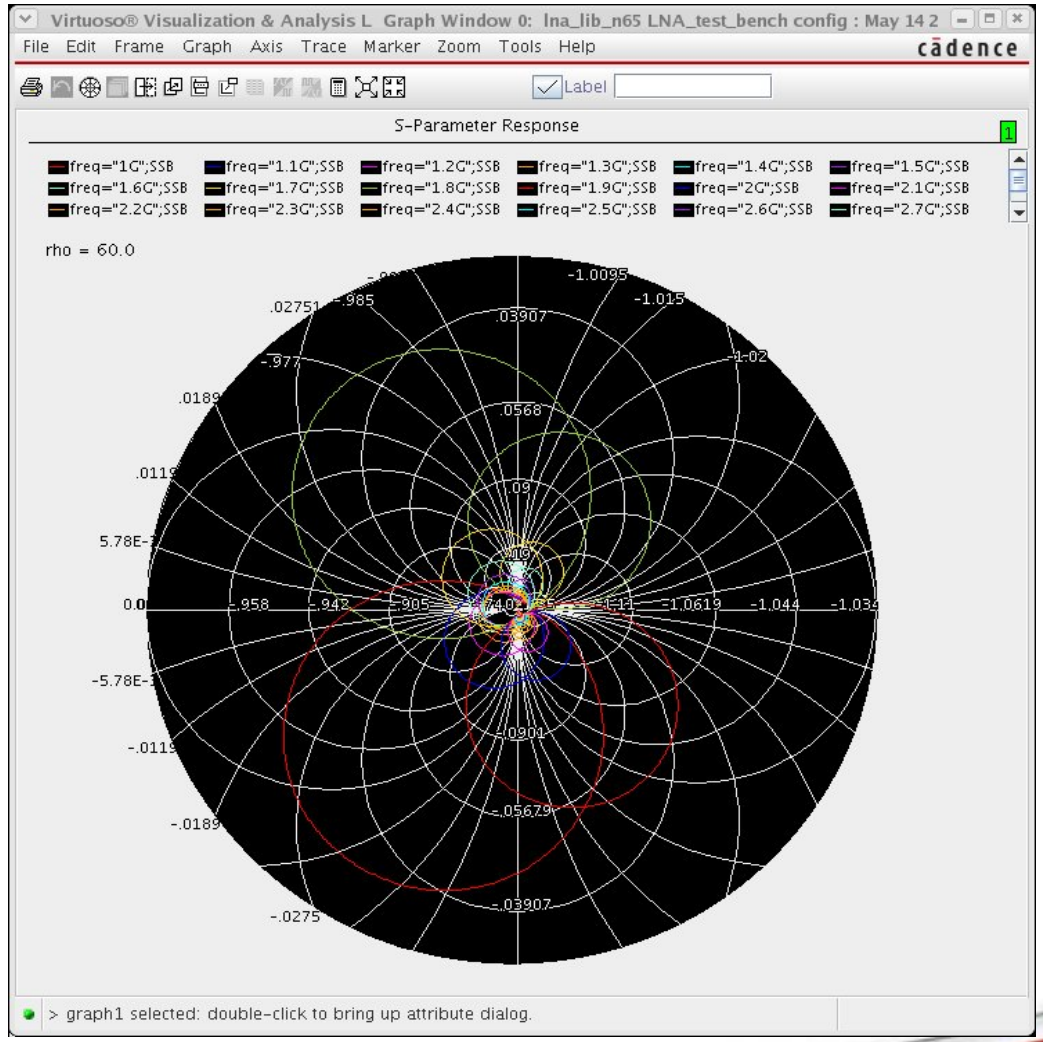


In this plot, “Plot type” is set to “Polar” on S21and S12

LNA Performance (cont.)

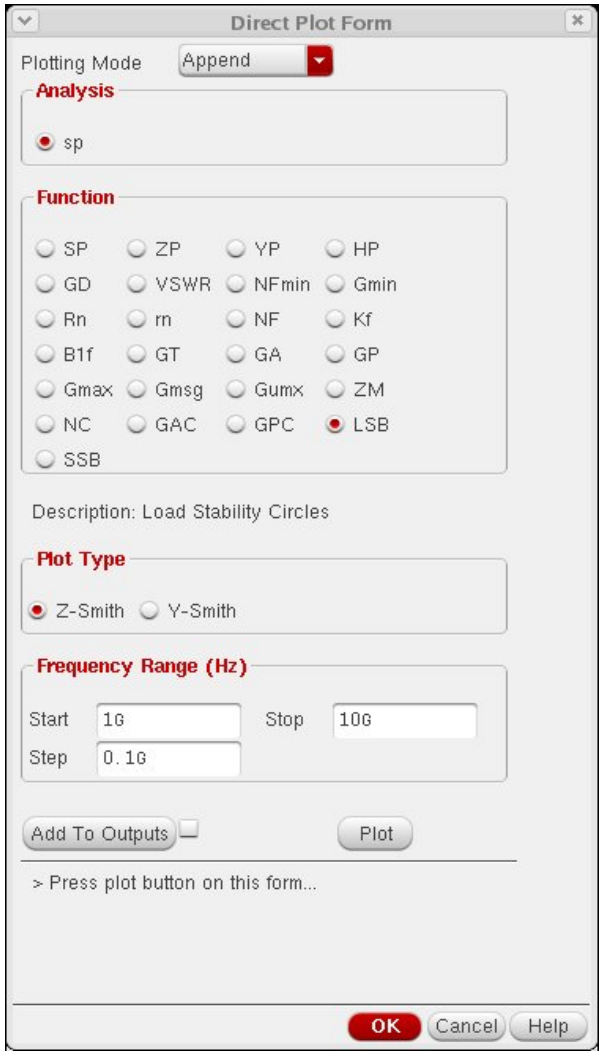
By setting Direct Plot Form shown below, we can see the source stability circles (SSB) of our LNA design.

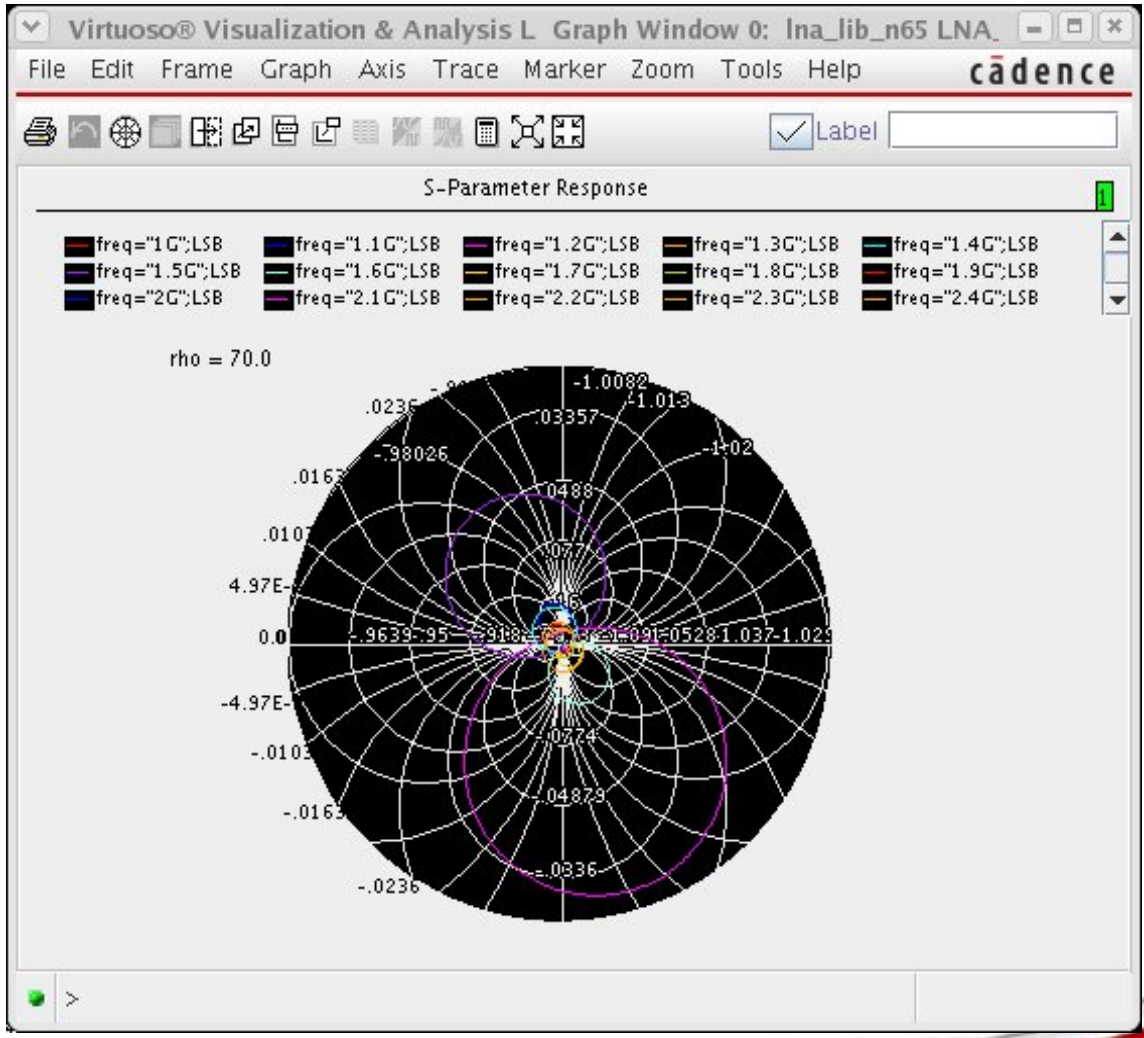




LNA Performance (cont.)

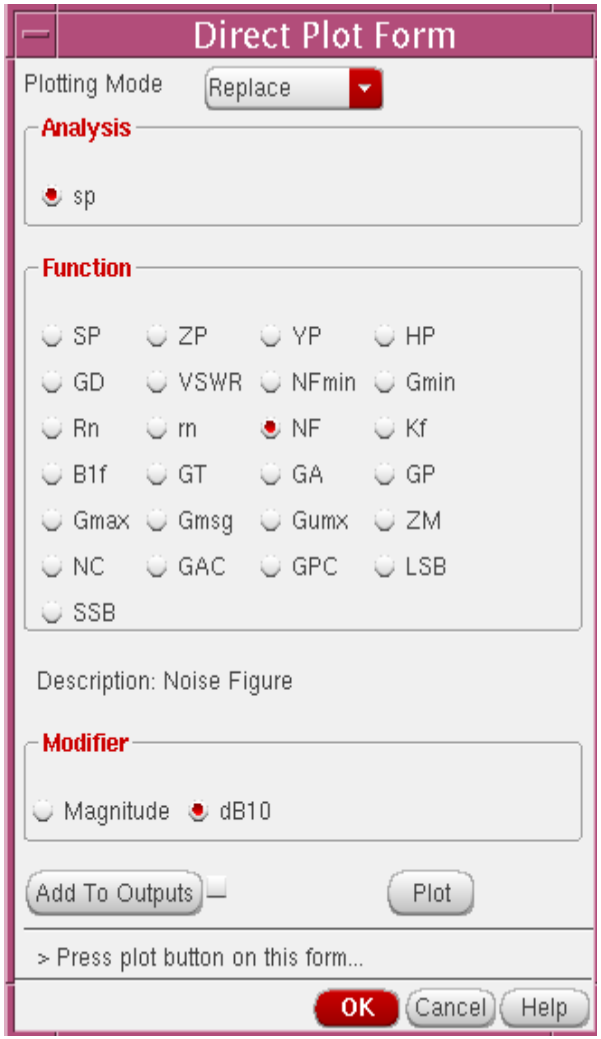
By setting Direct Plot Form shown below, we can see the load stability circles (LSB) of our LNA design.

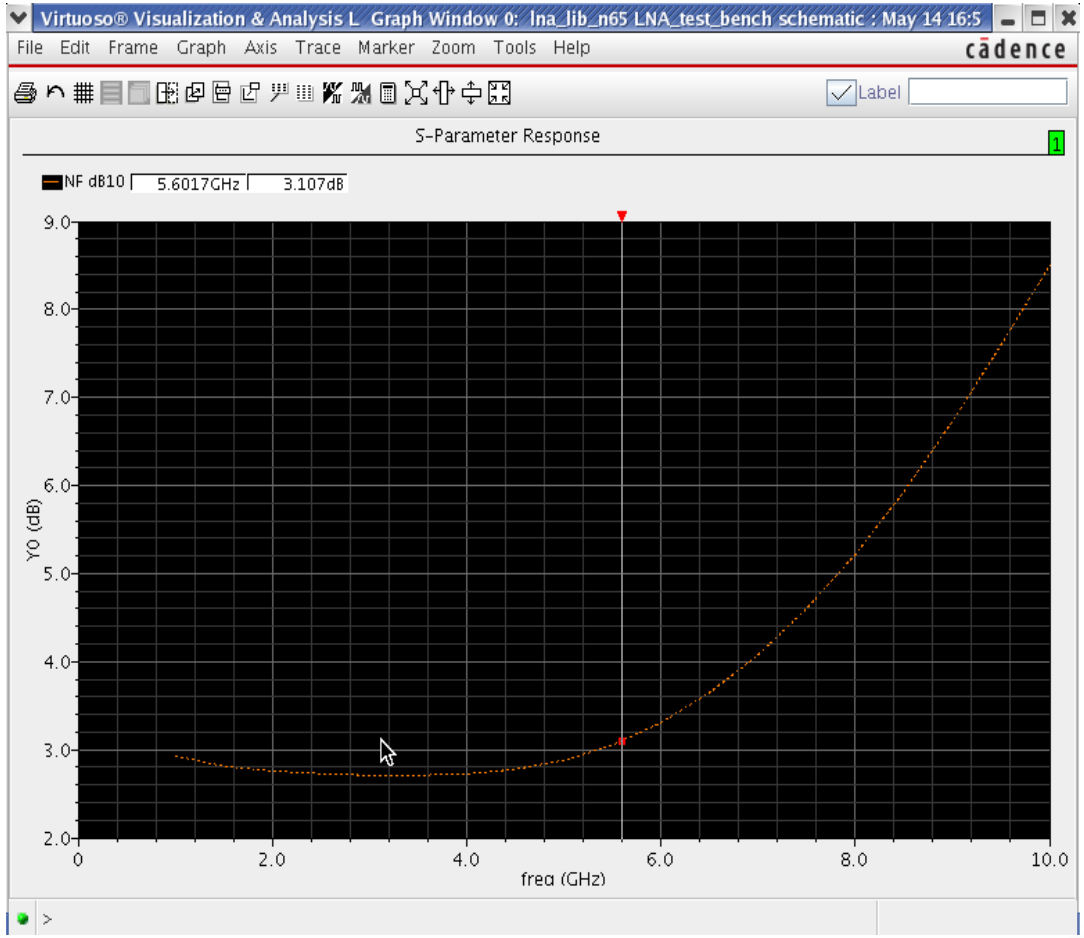




LNA Performance (cont.)

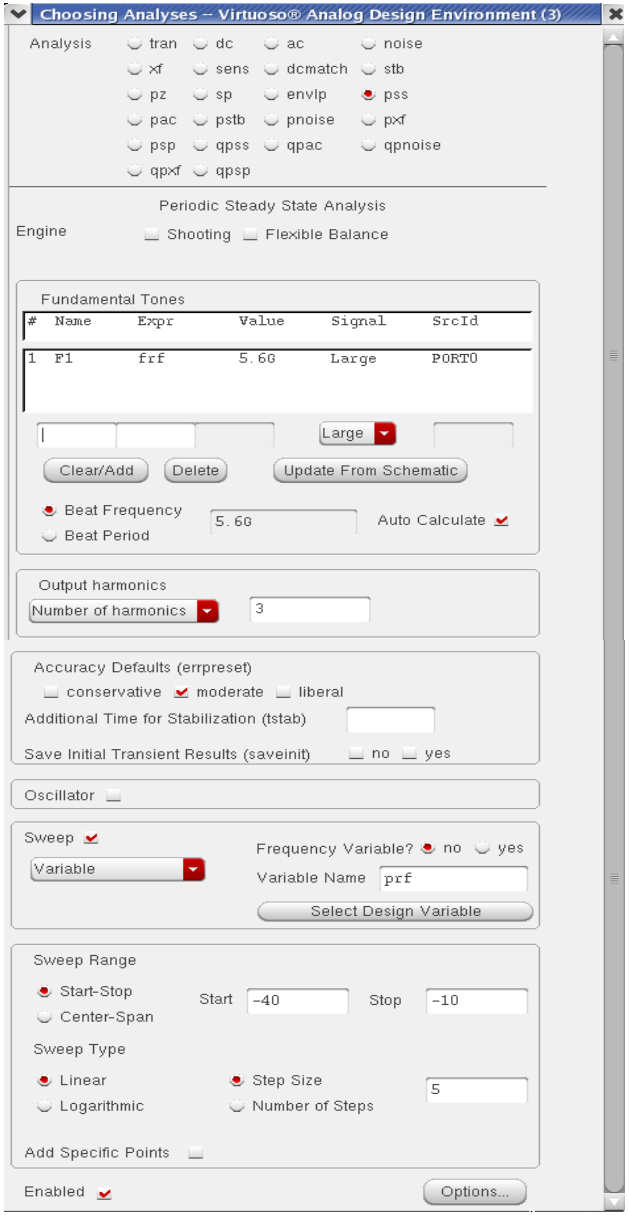
By setting Direct Plot Form shown below, we can see the Noise Figure (NF) of our LNA design. Finally, select "session->save state (state2) -> ok"





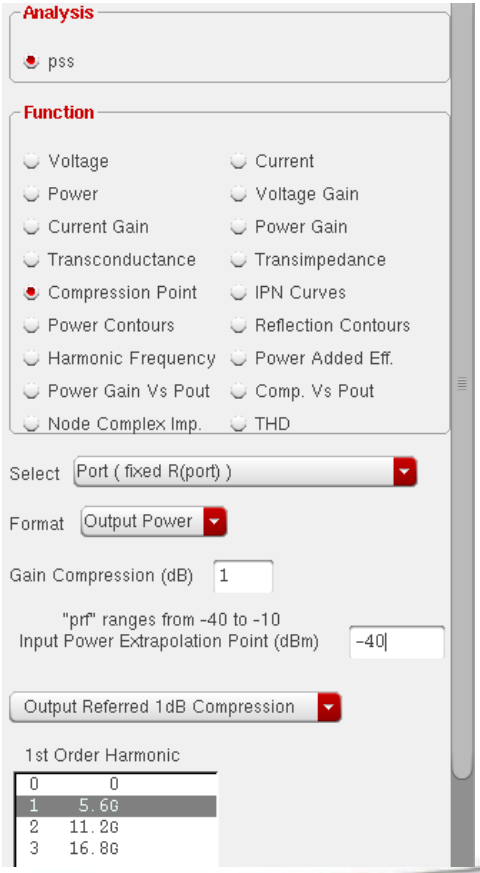
NF=3.107dB@5.6GHz

LNA Performance (3)

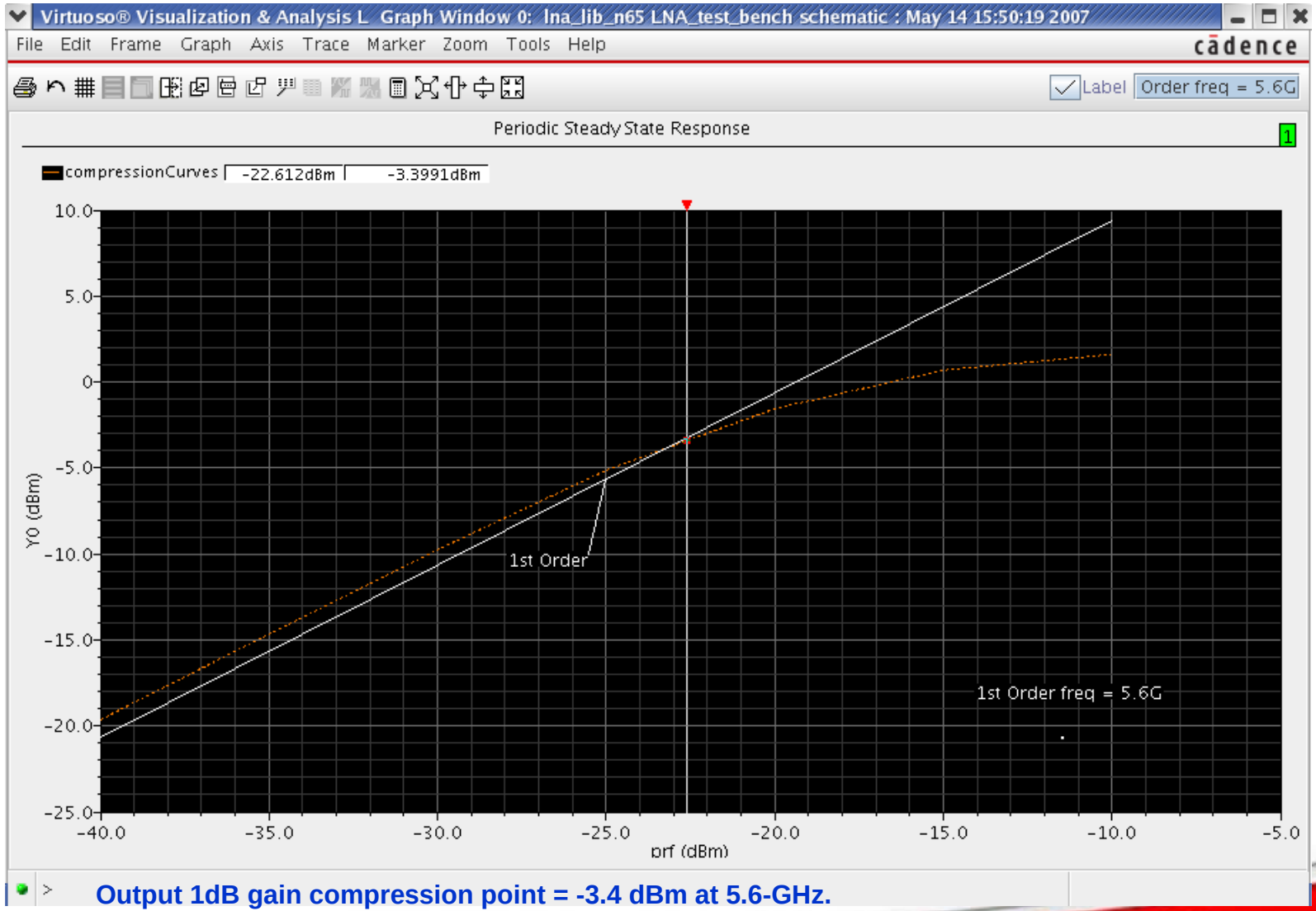


Run simulation, you can click "simulation-> netlist and run" from the Analog Design Environment (ADE) menu bar.

After finishing simulation, we can review the performance of LNA from Results->Direct Plot->Main Form…:



LNA Performance (3)

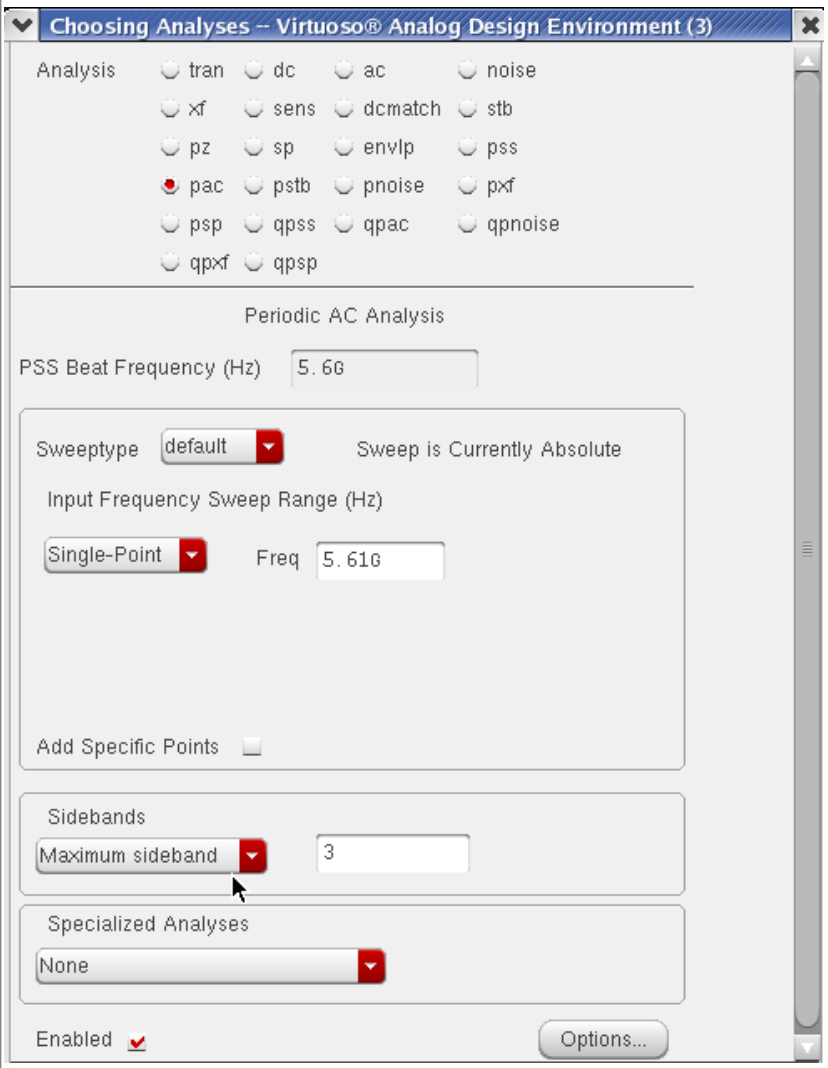


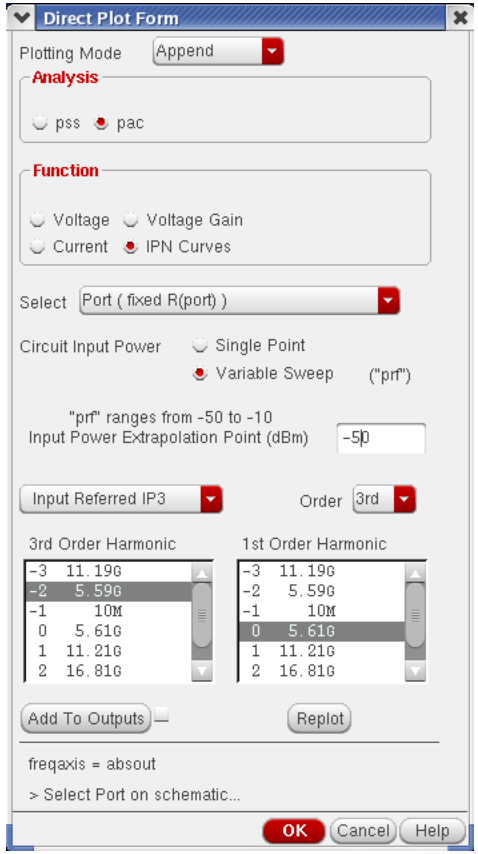
Output 1dB gain compression point = -3.4 dBm at 5.6-GHz.

LNA Performance (4)

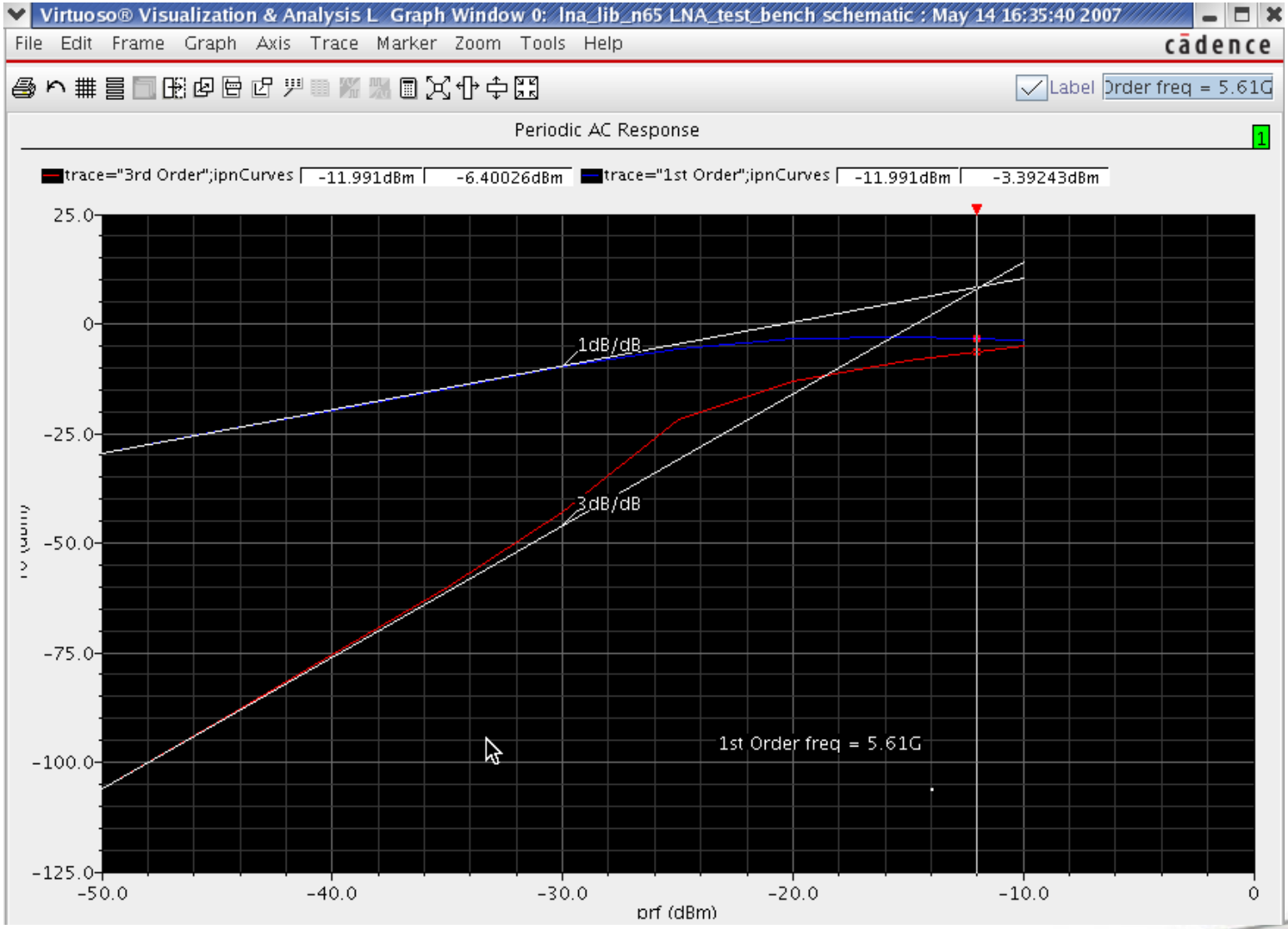
Run simulation, you can click **"simulation-> netlist and run"** from the Analog Design Environment (ADE) menu bar.

After finishing simulation, we can review the performance of LNA from **Results->Direct Plot->Main Form…:**





LNA Performance (4)



Input Referred IP3 = -12 dBm.

LNA Performance summary

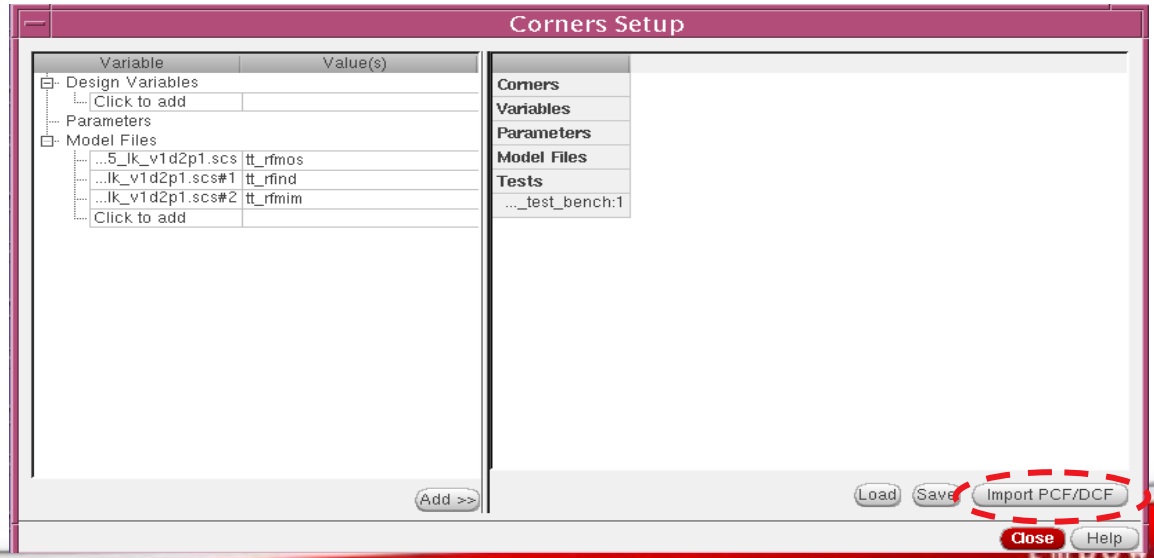
Specification	Simulation value	Unit
Center frequency	5.6	GHz
Gain	20.37	dB
Input return loss	-21.32	dB
Output return loss	-22.02	dB
Noise Figure	3.1	dB
3dB bandwidth	~1.5	GHz
Supply voltage	1.2	V
Supply current	7.05	mA
Output P1dB	-3.40	dBm
IP3	-12.0	dBm

Corner simulation

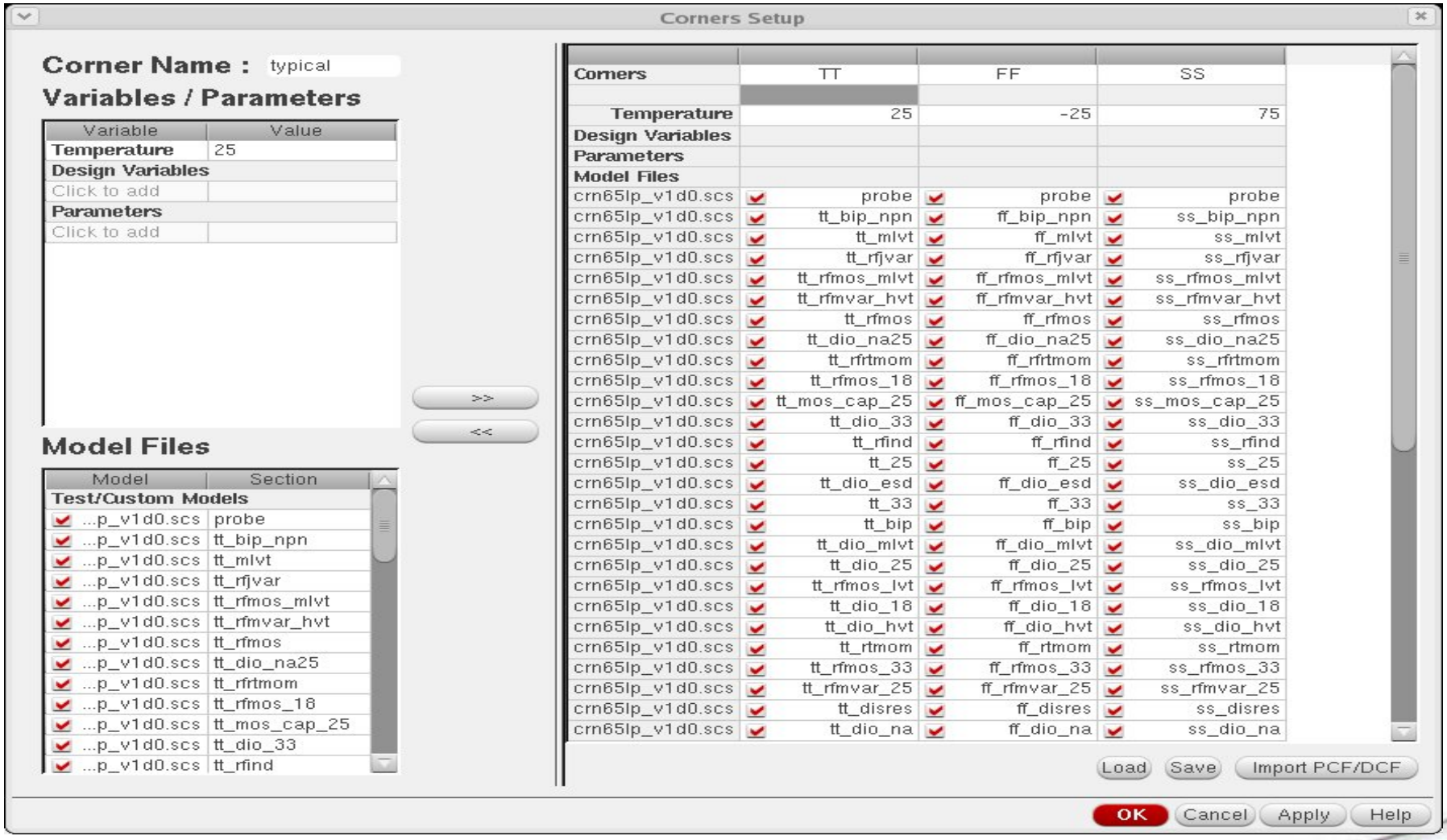
In addition to cover this typical case, we may sometimes want to simulate our design to cover the process variations in different corners. This can let us know whether the circuit performance specifications will still meet even when the process variation sift to different corner. Furthermore, this can also improve the product yield of our design. In our case, we simulate our design in three different corners: the typical case (tt_rfmos), the fast-best case (all devices in FF) and slow-worst (all devices in SS) case. By loading the well-defined PCF file released along with TSMC's PDK, you can find the corner analysis window shown below:

1) In schematic window, Launch->ADEXL
2) Right click on the "lna_lib:LNA_test_bench:1" in the "Tests and Analyses" assistant window and select "Load" to load "state1".
3) Click on "Corners" in the "Parameters, Sweeps, and Corner" assistant window.
4) Click on "import PCF/DCF" and select "./models/spectre/tsmcN65.PCF.sdb"
5) Turn off " Nominal corner" and click "Run" button in the "Run mode" assistant window.

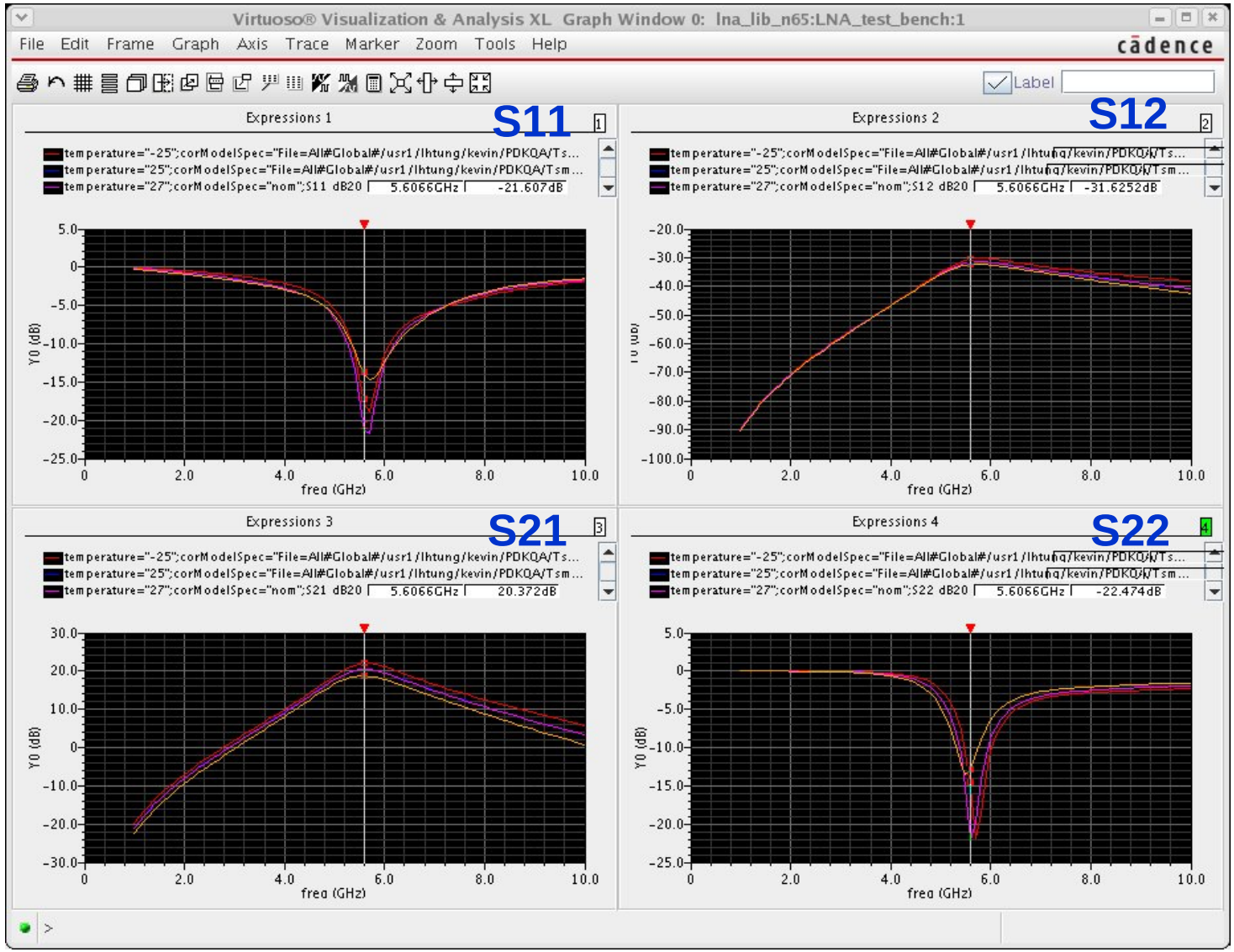
Import PCF file



Loading PCF file for corner analysis



Corner simulation result



Contents

Layout Creation

After completed the pre-layout simulation and make sure the functionality and the circuit performance are all correct and in the design specifications, we can now start to create the corresponding layout for our design. The layout creation procedures are partitioned into three parts: "Schematic-Driven-Layout", "Components Placement" and "Manual route".

- **Schematic-driven-layout**
- **Components placement**
- **Manual routing**

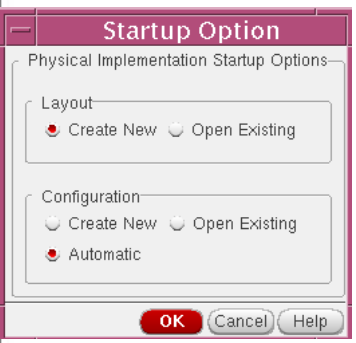
Schematic-driven-layout

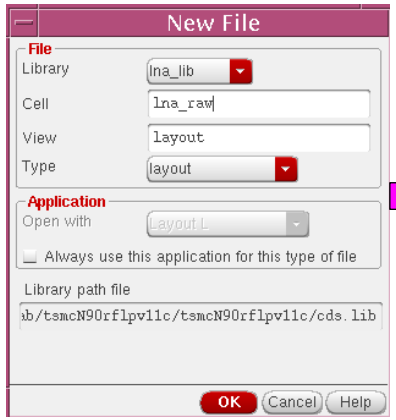
Steps for Schematic-Driven-Layout method:

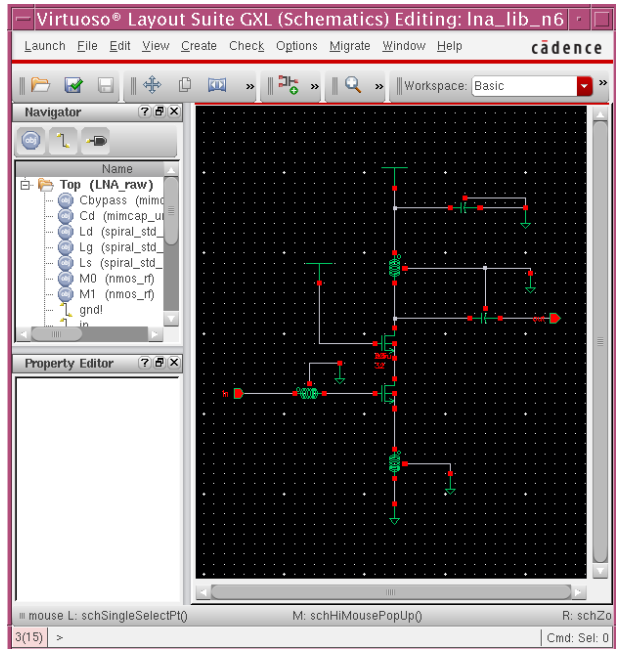
1. Open the schematic view of our design (lna_raw).

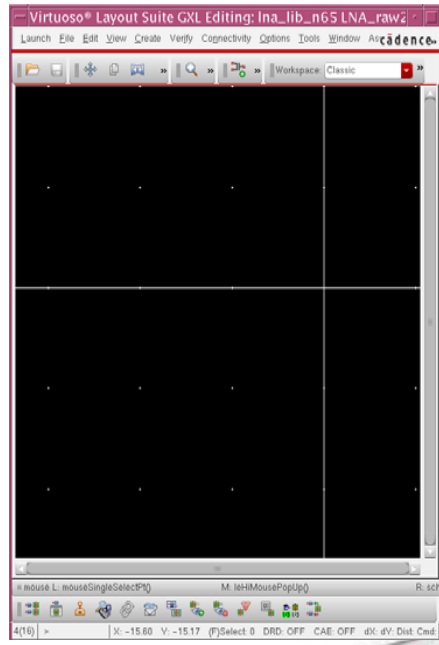
2. From the schematic menu select “Launch -> Layout XL (or Layout GXL)”.

3. After selecting this option, a small dialog box will first open to let users select the cell name and view name for the layout. Upon finished the selection of the cell name and view name, a Virtuoso XL layout window popup for layout generation.



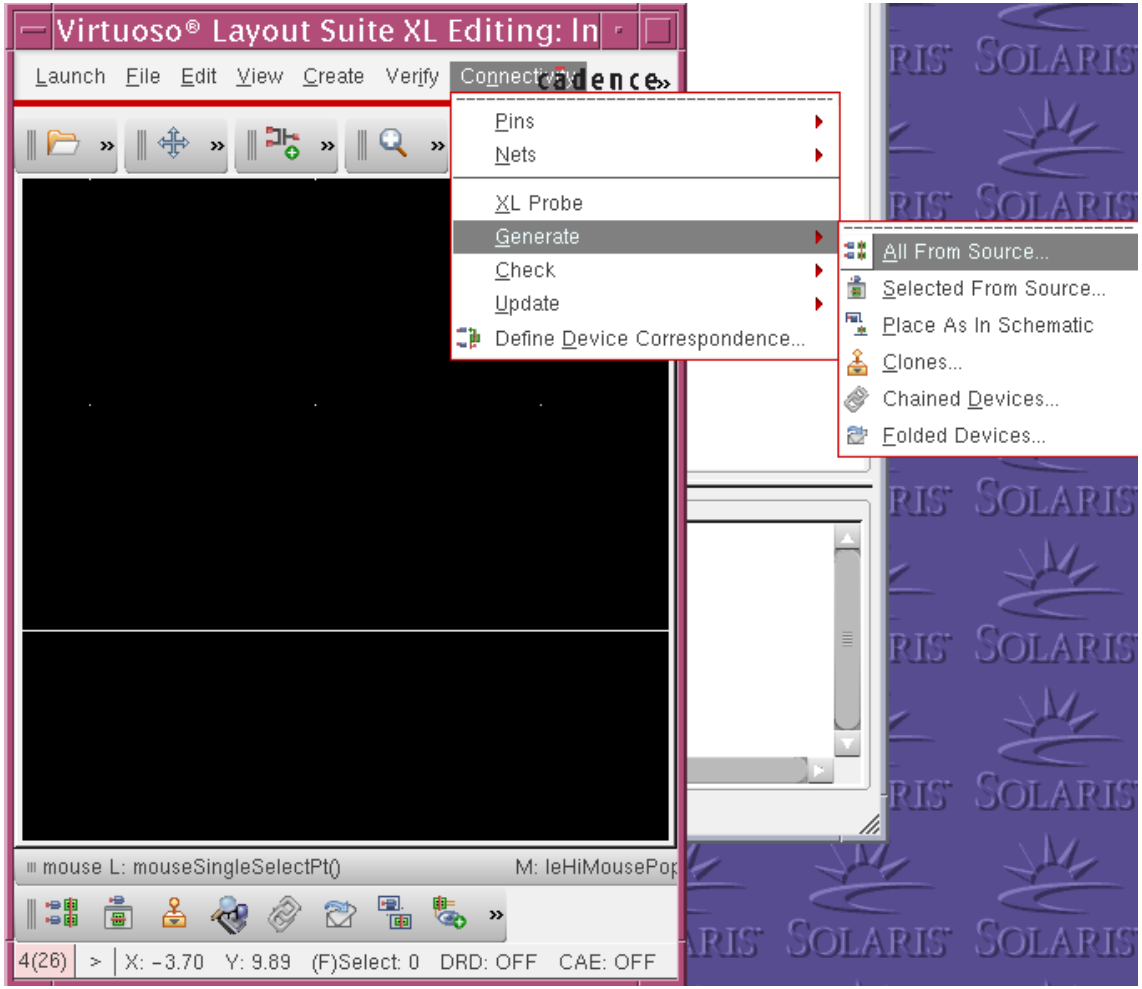






Schematic-driven-layout

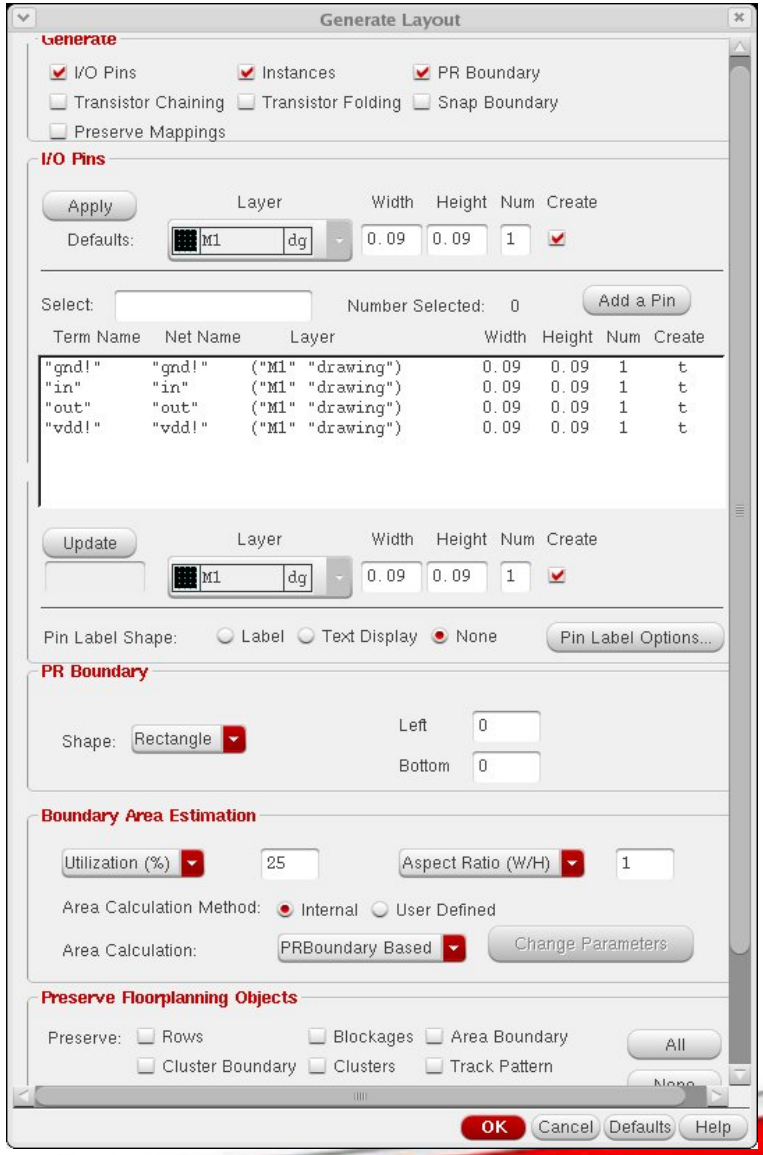
4. From the Virtuoso XL layout menu, select “Connectivity -> Generate -> All from source …”



Schematic-driven-layout

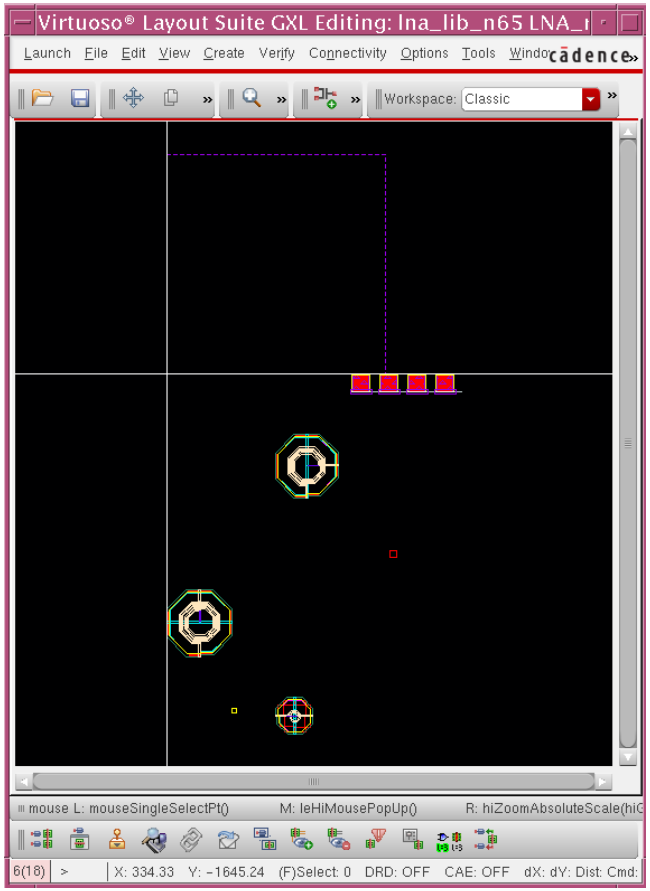
5. A layout generation options window appeared and prompts users to setup the pin layers, pin width, pin height, boundary layer …and so on for layout generation.

6. After finished the selection of above information, some rectangles that represent the components (transistor, inductor, capacitor and I/O pins will show up in the bottom of the layout window.

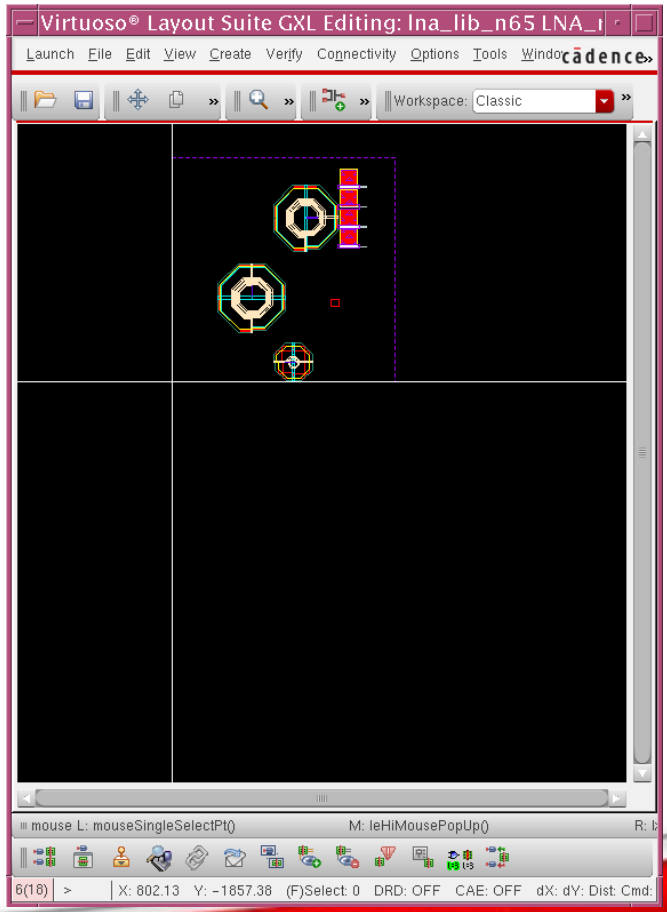


Components placement

The next step is to do the device placement. The only one thing that you need to do is to place all the components and I/O pins in the layout window into the design area (cell boundary). In IC61, use (**Connectivity -> Generate -> Place As In Schematic**) as your first placement reference. By selecting devices/IO pins and dragging them to proper locations inside the design area, we can complete the component placement. During the device movement and placement, the lines represent the connections of select object to other objects will show up. This can help you to decide where to properly locate the selected object.

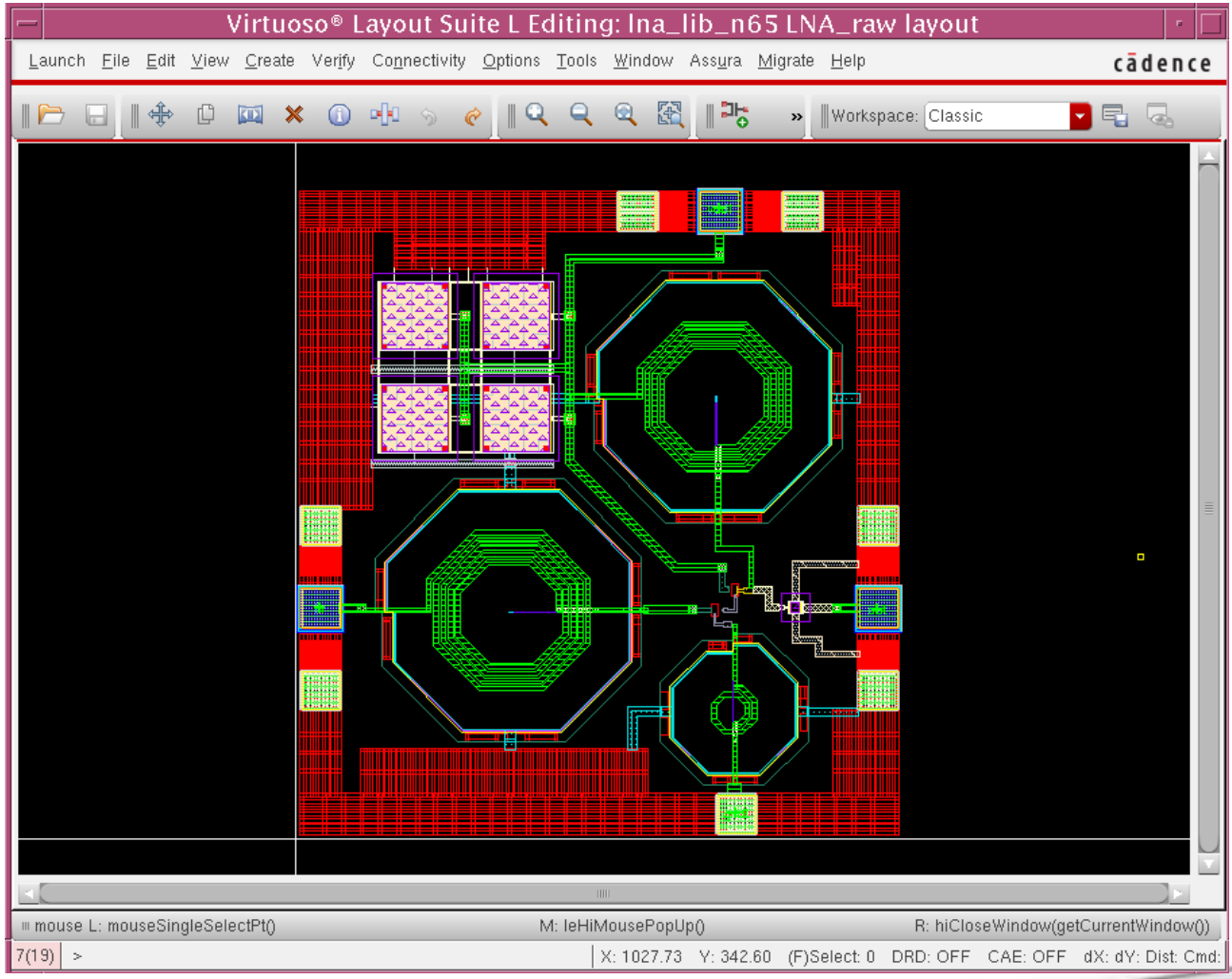


Place As In Schematic



Manual routing

For RFIC design, most designer prefer manual routing by themselves because the performance is layout-dependant. Different routing may cause different parasitic. Below is an layout example of the LNA_raw design.



Contents

Physical Verification

After the layout creation is completed, we have to start the physical verification to make sure this layout is DRC free and each device in the layout is completely match to its corresponding component in original schematic. After that, the parasitic extraction is necessary for post-layout simulation to make sure our design still work well after taking the parasitic R&C effects into account. Generally, the physical verification procedures can be divided into three parts: the design rule check (DRC), layout V.S. schematic check (LVS) and parasitic extraction (RCX).

- Assura DFII Flow
  - _DRC
  - _LVS
  - _RCX

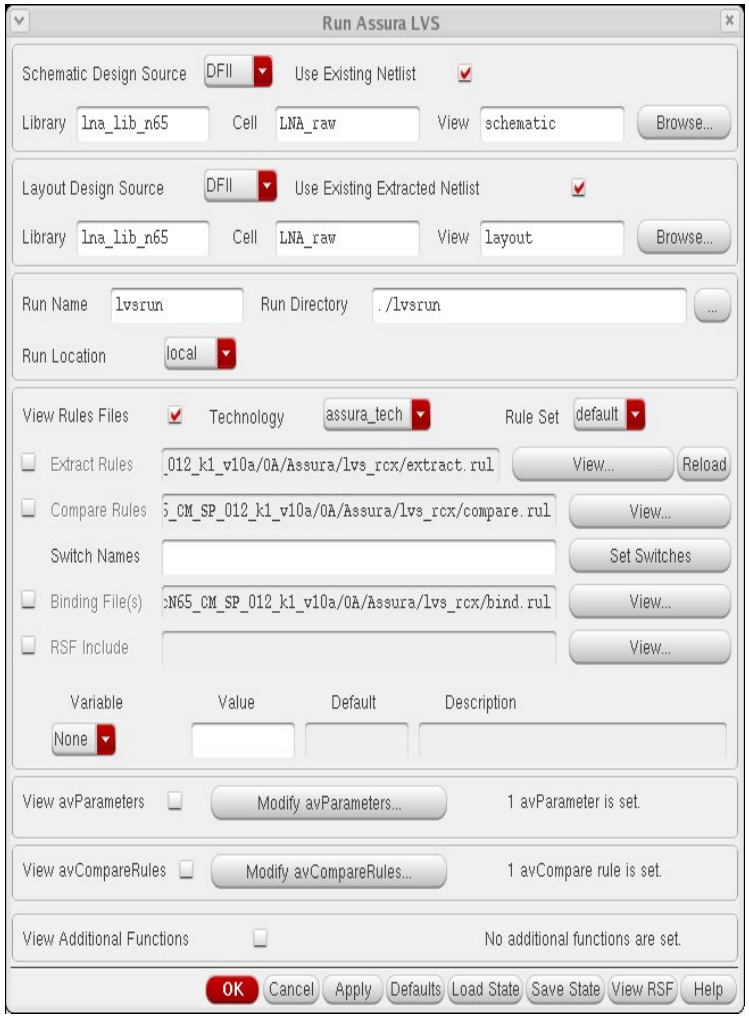
Assura DRC Flow

Currently TSMC has not supported Assura RF DRC deck yet.
You can refer the procedure of Calibre drc flow.

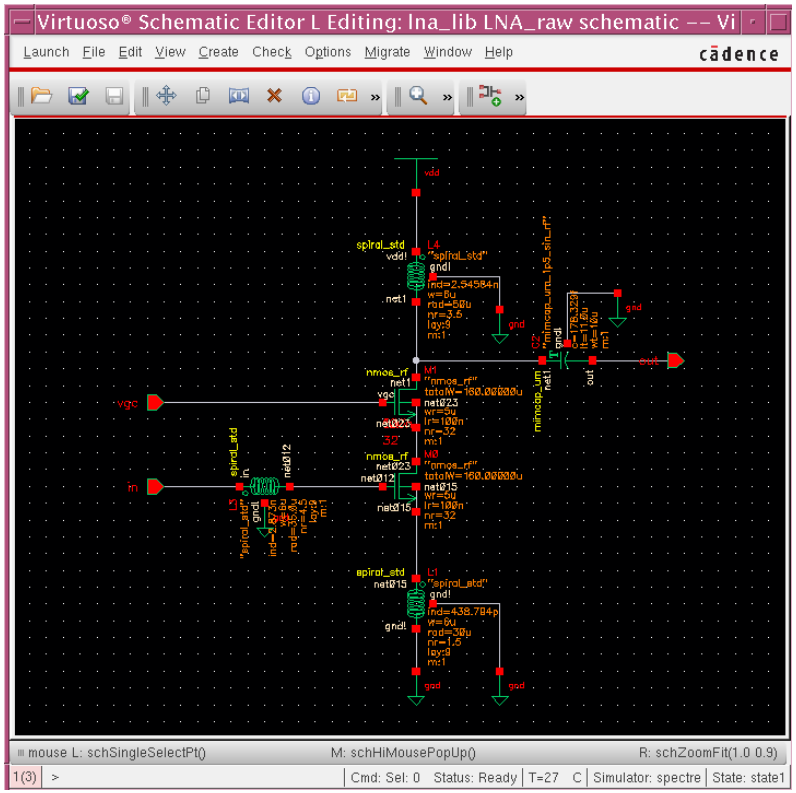
Assura LVS Flow

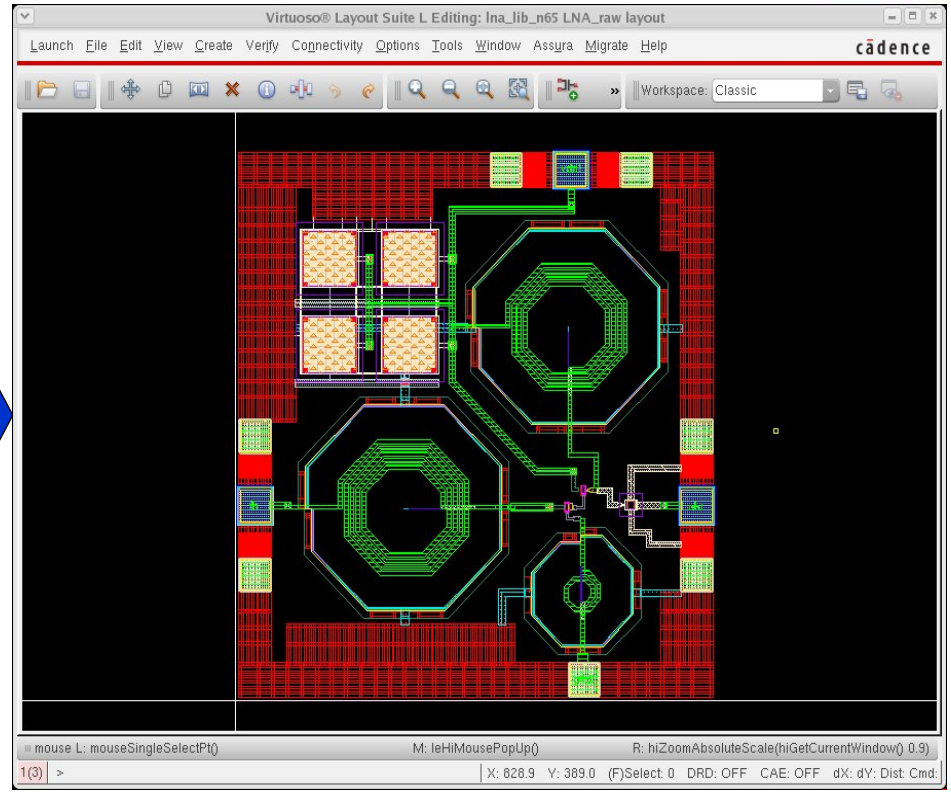
After the layout has no DRC violations (DRC free), the next step is to run the LVS check to make sure the layout is totally match to the schematic.

1. Click “Assura-> Technology” to load the file, “assura_tech.lib” in the PDK install directory.
2. Click “Assura-> Run LVS..” in layout window to invoke Assura LVS graphic user interface.
3. Fill in the “Assura run directory” and select the “Technology” field to “assura” in Assura LVS window.
4. Click “OK” to run the Assura LVS and see the result. If the layout isn’t matched to schematic, you have to manually re-edit the layout and re-run the LVS check to make the LVS result matched.

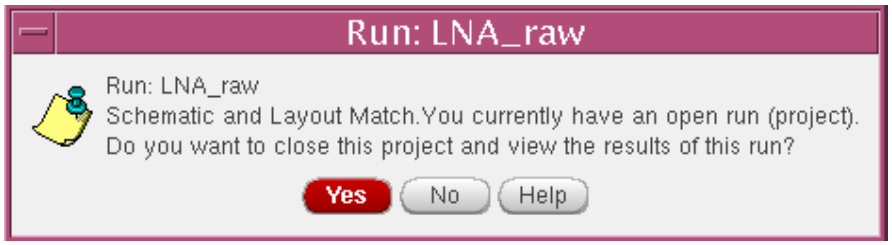


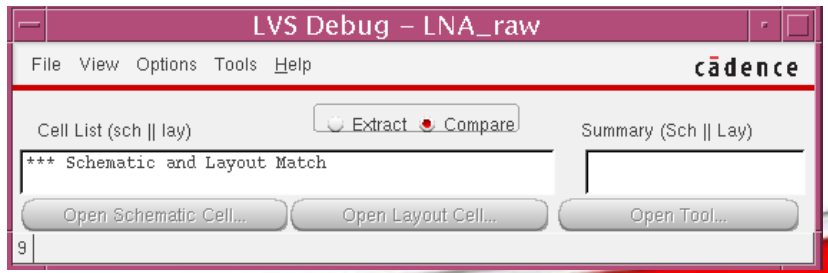
Assura LVS Flow (con't)





At this case, after running LVS, the result shows "Schematic and Layout Match"

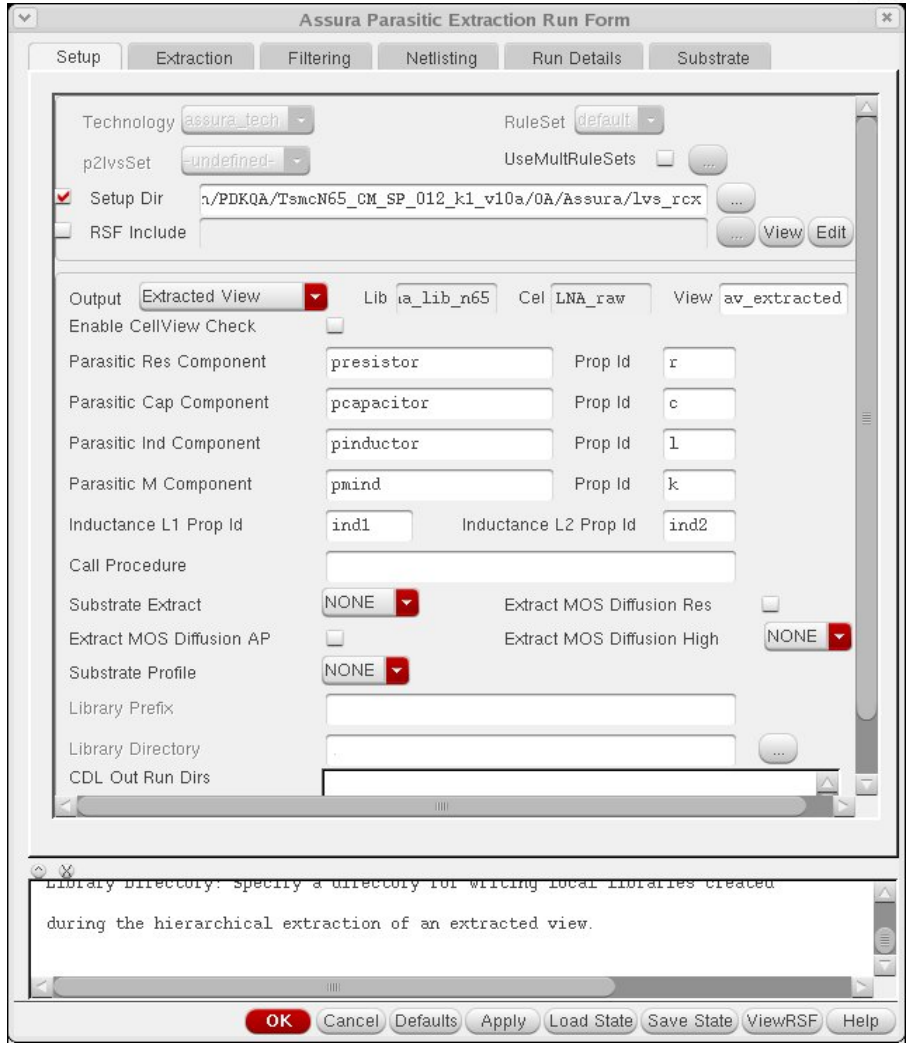




Assura RCX Flow

When the layout is DRC free and LVS clean, the next step is to perform the RC extraction. This step is to prepare the layout extracted netlist for post-layout simulation.

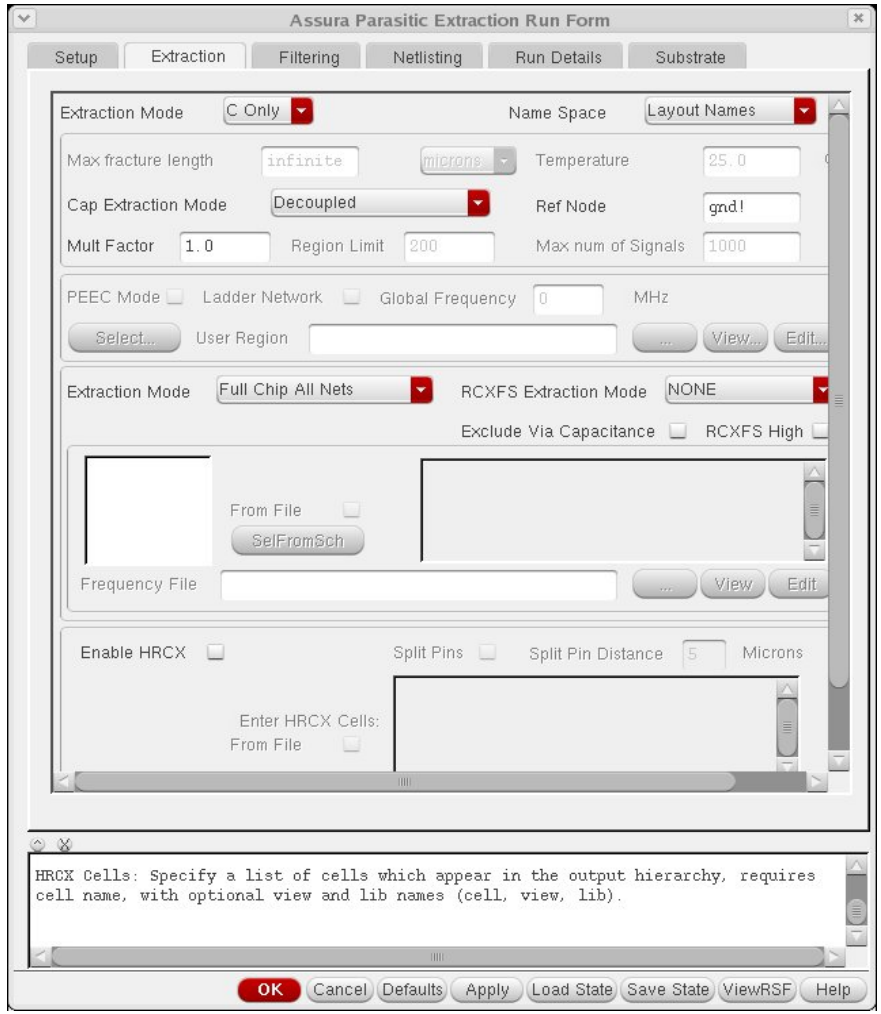
1. **Click “Assura-> Run RCX...” in layout window to invoke Assura RCX graphic user interface.**
2. **Select “Output” to “Extracted View” in “setup” folder of Assura RCX window to output extract result to “av_extracted” view.**



Assura RCX Flow (cont.)

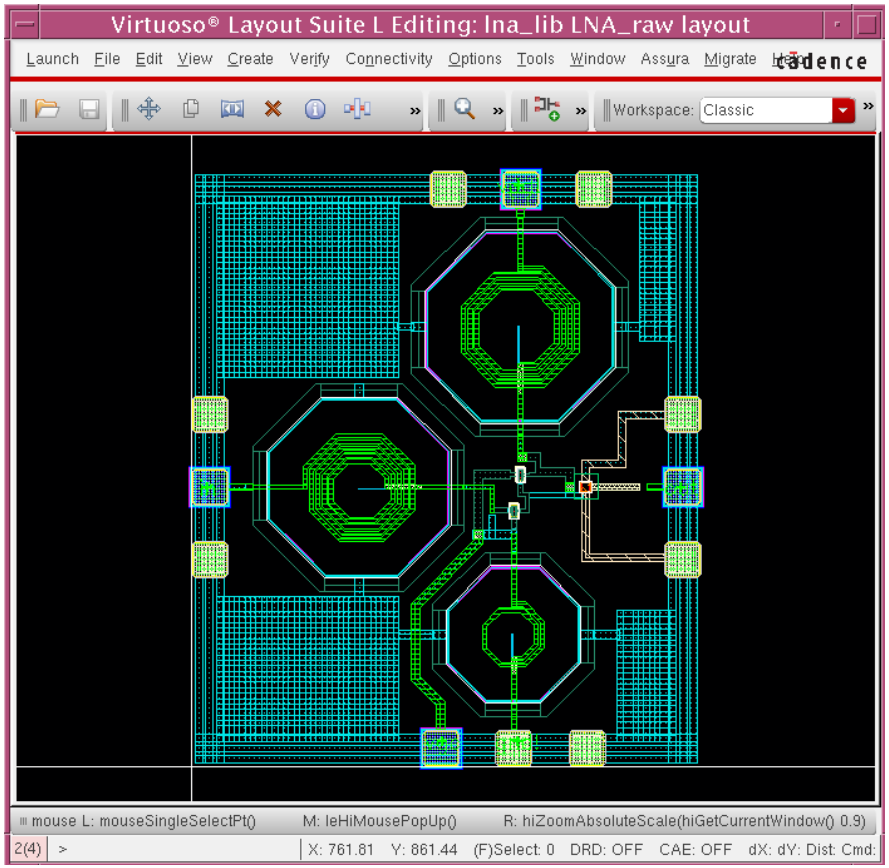
3. In the “Extraction” folder of Assura RCX window, select the “extraction mode” to “C only “ (if you want to extract only C), set the “Name space” to “Schematic Names” and fill in the “Ref Node” (here we use “gnd!”).

4. Click “OK” to start the Assura RC extraction. After the RC extraction is completed, a new view (“av_extracted” view) which contains not only the original components but also the parasitic devices will be generated and then can be used for post-layout simulation.

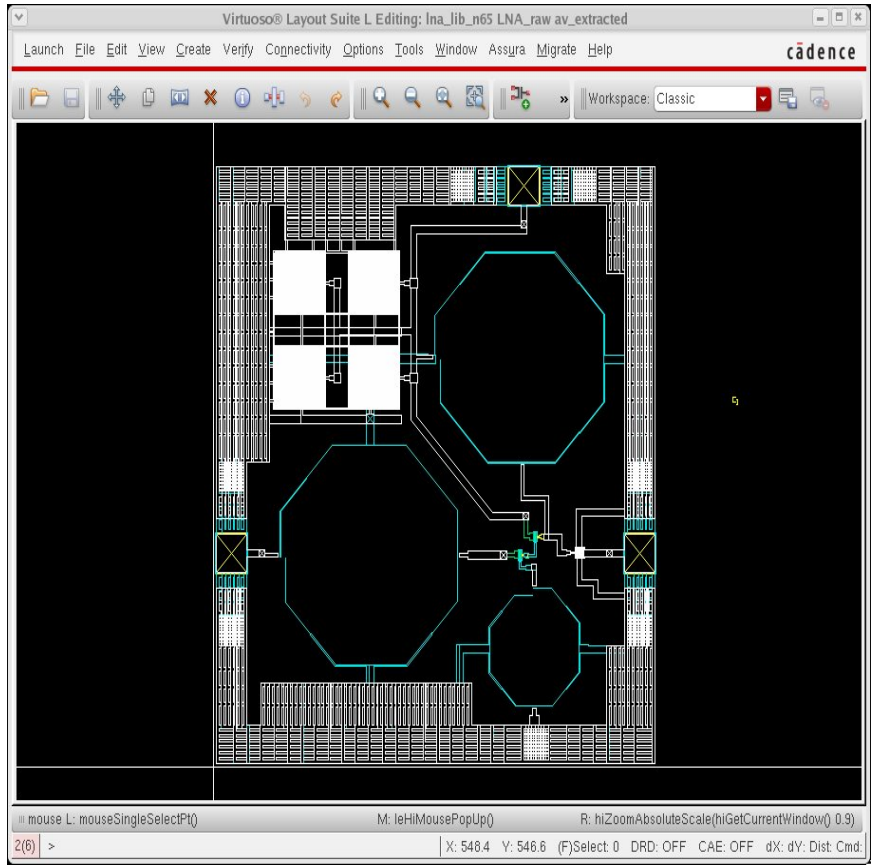


Assura RCX Flow (cont.)

The av-extracted view by C-only mode is showed below:



Original layout



Av-extracted view by C-mode only

Contents

Post-layout Simulation

When you accomplished the physical verification, the last step to tape-out is to perform the post-layout simulation on the extracted netlist/view. During the post-layout simulation, not only the original components but also the parasitic R&C (depends on what you have extracted in RCX stage) of the interconnections are taken into consideration.Therefore, we can say that the post-layout simulation result is much closer to the real silicon measurement data than the original pre-layout simulation result.

- **Assura RCX extracted view**
  - **_C-only mode**

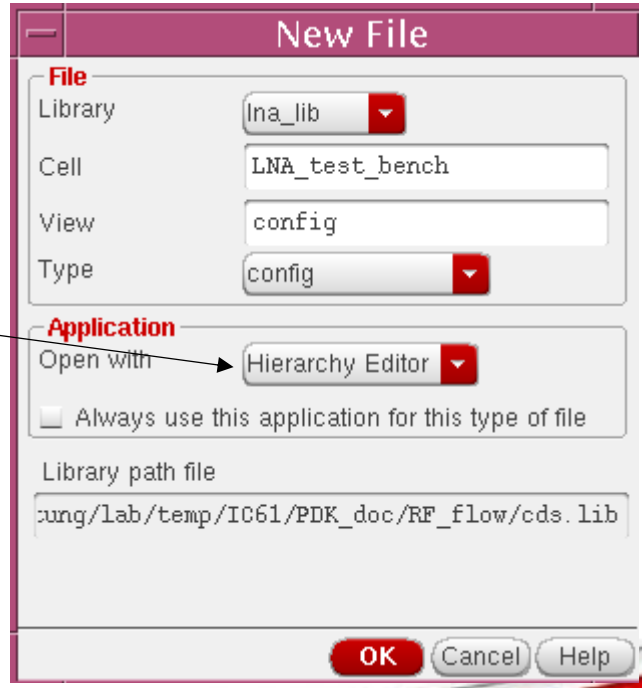
Post-layout Simulation

•Creating a Configuration file for Post-layout simulation

1. In the CIW or Library Manager, select File-> New->Cellview
2. Set up the Create New File form as follows:

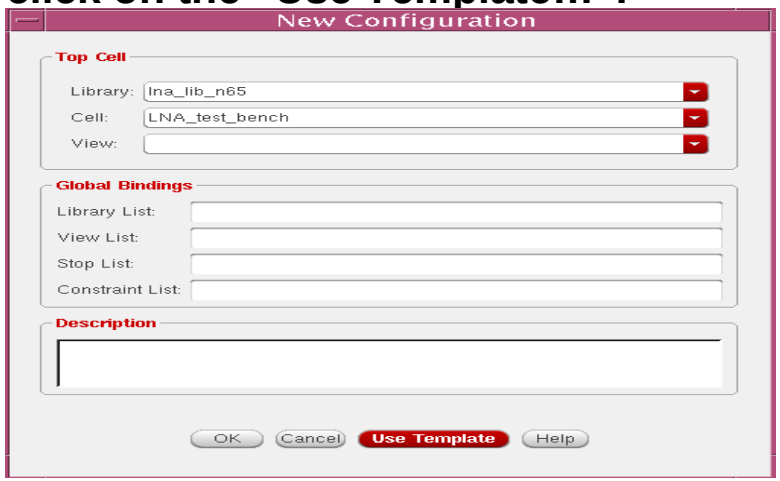
Enter the new view name into the Name field

Select “Hierarchy-Editor”

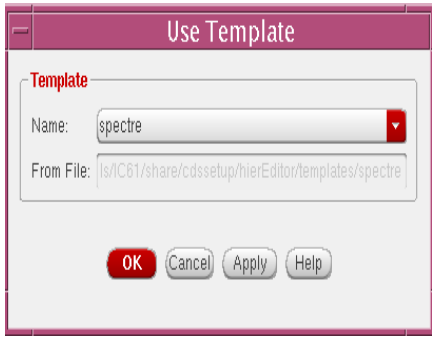


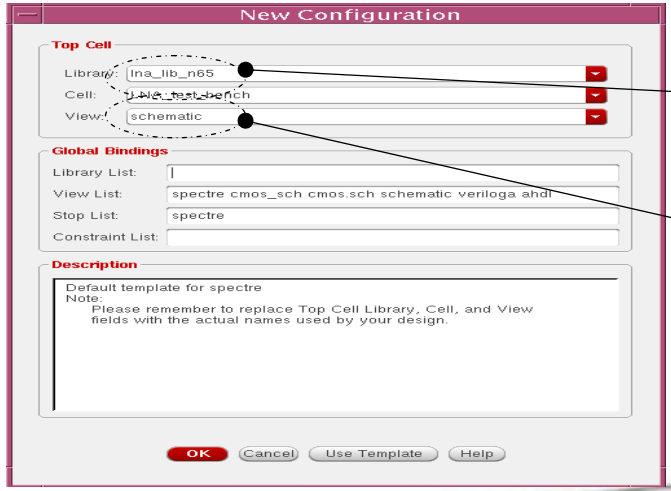
Post-layout Simulation

3. At the top the form enter the view to “schematic”, and at the bottom, click on the “Use Template…”.



4. The Use Template form opens; cycle the Name to spectre and click OK then the “New Configuration” form is like below:

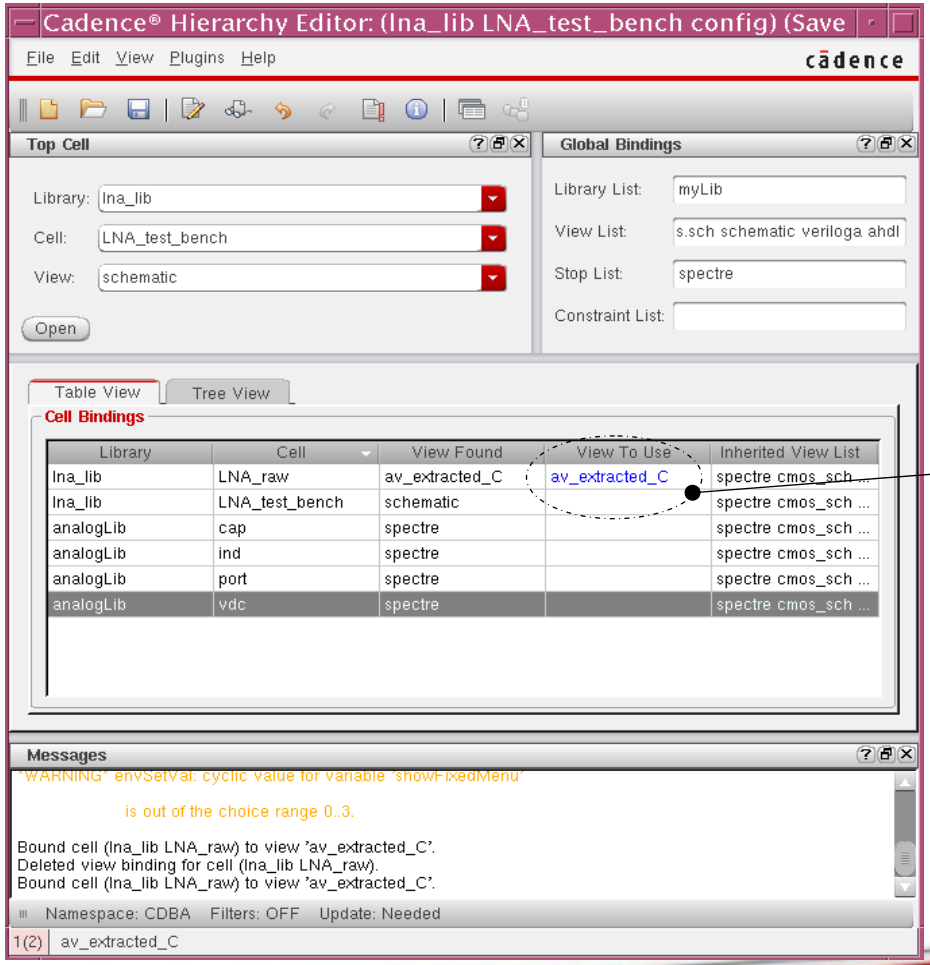




Post-layout Simulation (cont.)

5. **Edit the hierarchy for the design:**

Change the “View to Use” to which you want at “LNA_raw “ cell and save the file.

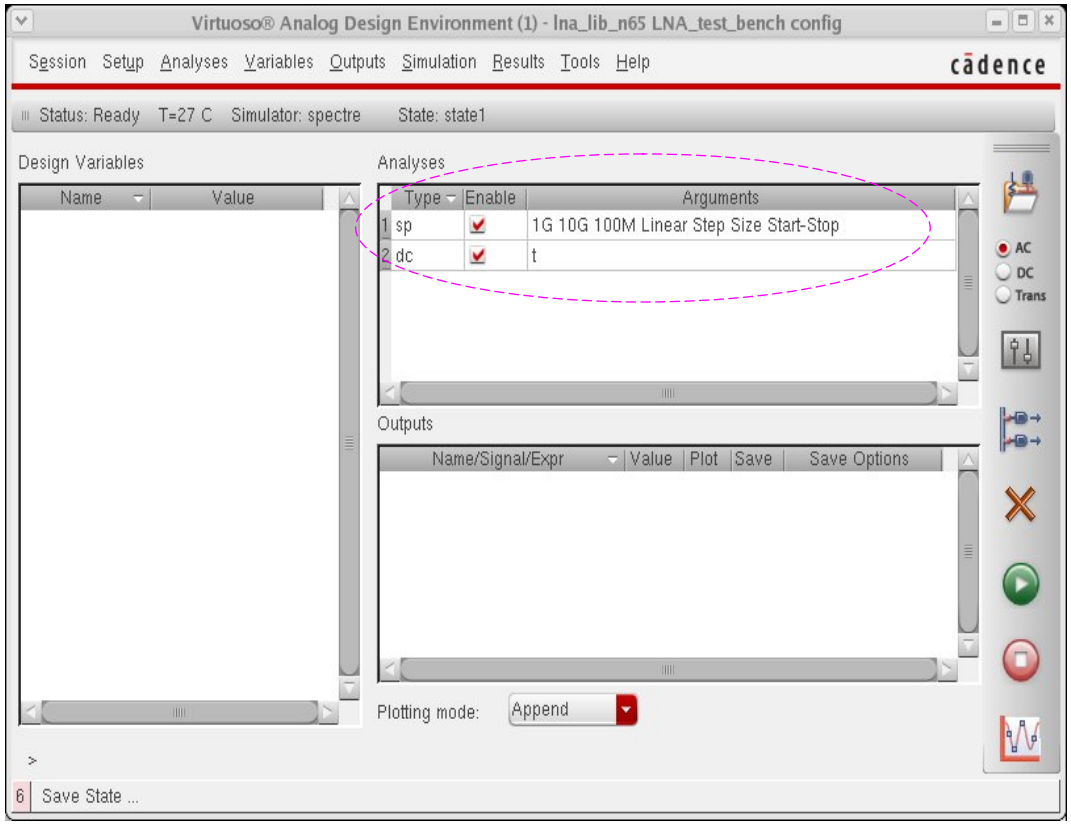


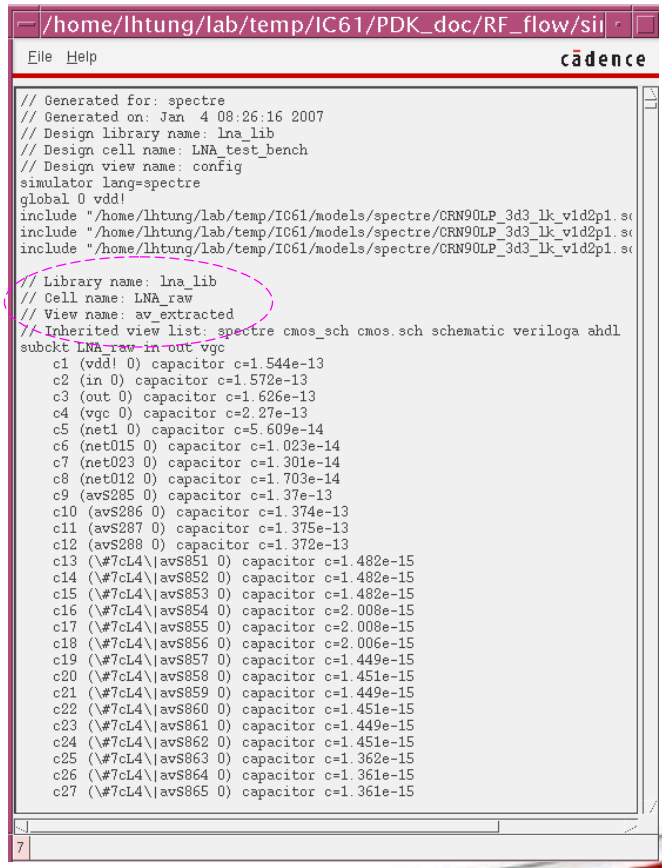
You can change the view that you have created by extraction tool.

Post-layout Simulation (cont.)

•Run post-layout simulation with extracted view

Now, you can run post-layout simulation by changing design setup to configuration created previously. You can output the netlist to make sure a correct view is used for post-layout simulation.





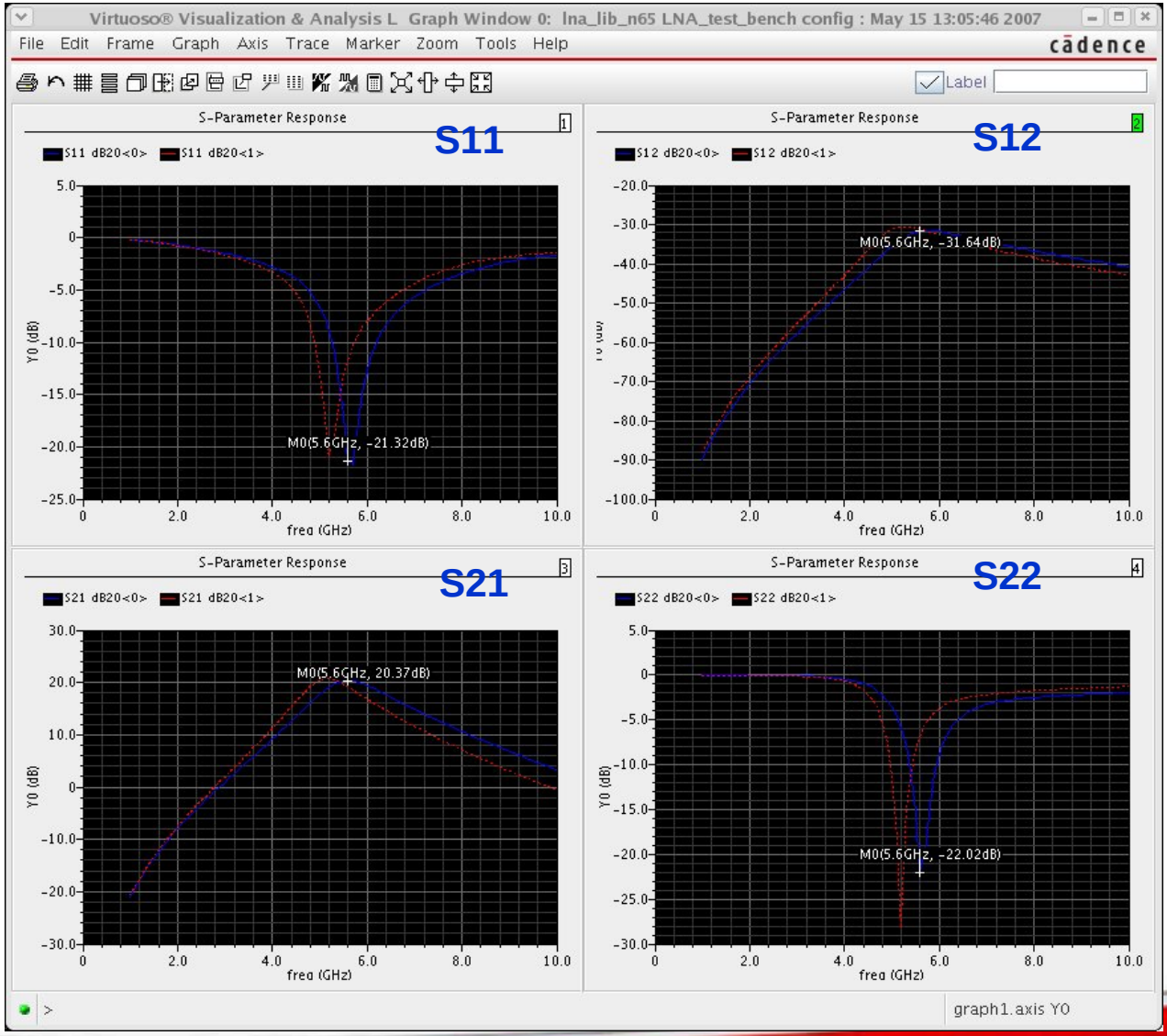
Post-layout Simulation Result

After completing the post-layout simulation. We will check the result between different RC extraction methods.

- **Assura RCX extracted view**
  - **Assura C-only mode v.s. Pre-layout simulation**

Assura C-only mode v.s. Pre-layout simulation

S-parameter (20dB)

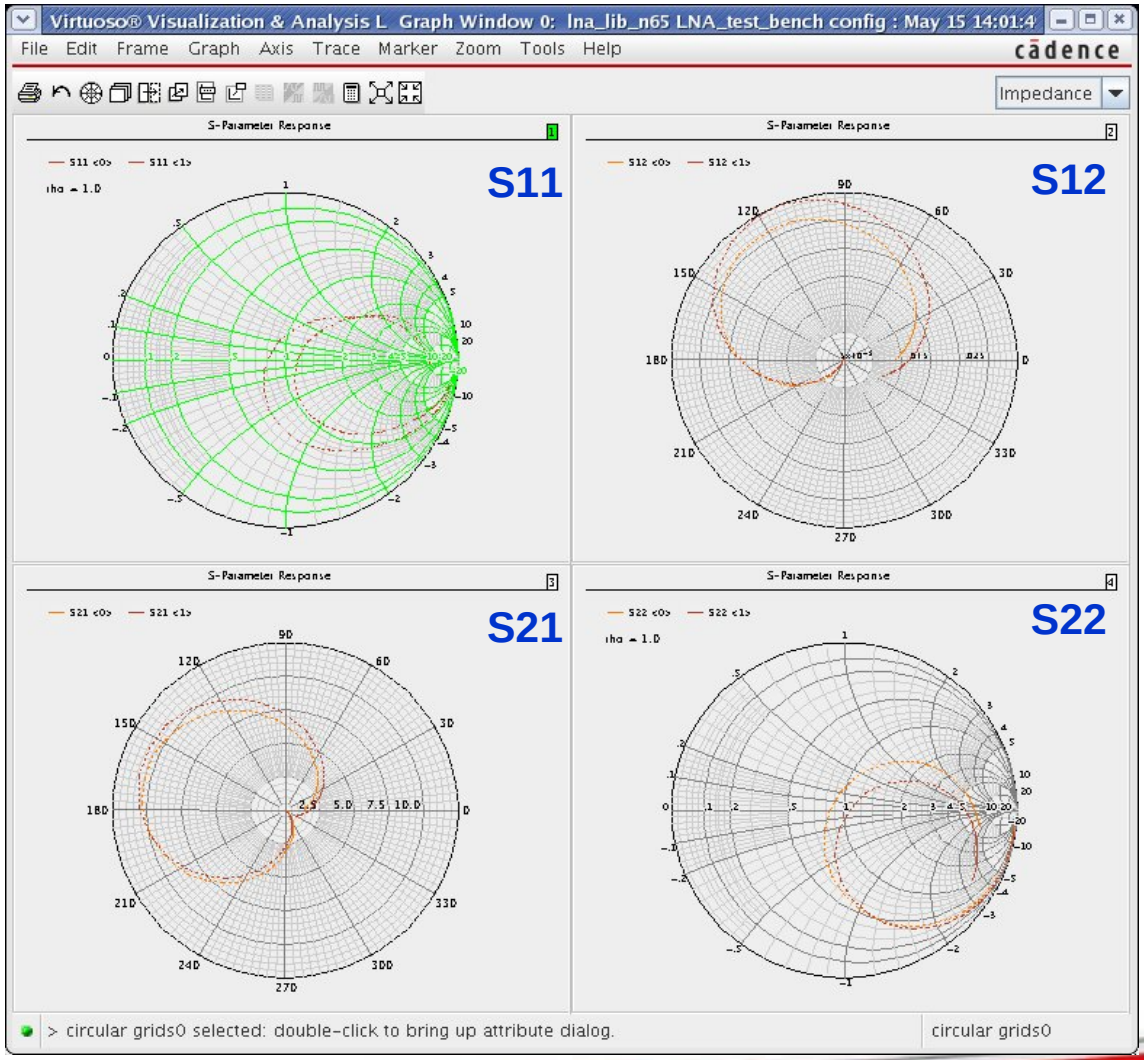


Solid line: pre-layout

Dash line: C-only mode

Assura C-only mode v.s. Pre-layout simulation

S-parameter

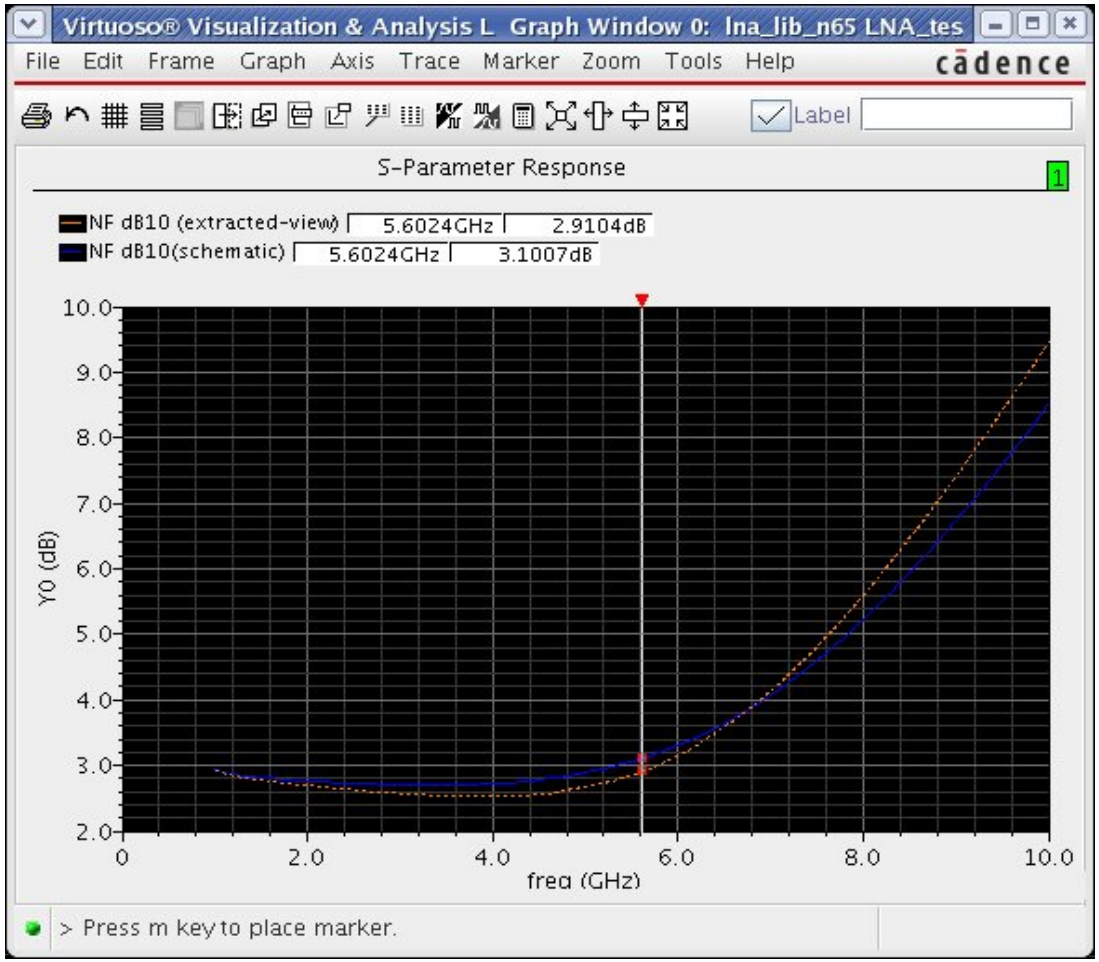


Red line: pre-layout

Brown line: C-only mode

Assura RC mode v.s. Pre-layout simulation

Noise Figure



Solid line: pre-layout

Dash line: RC mode

This slide is intentionally left blank