

# TSMC PDK RF Flow Guide (IC61): A Low-Noise Amplifier (LNA) Design Flow Example of TSMC CRN65LP Process Design Kits (PDK)

Copyright 2007, Taiwan Semiconductor Manufacturing Company, Ltd. All Rights Reserved. No part of this publication may be reproduced in whole or in part by any means without prior written consent.

## 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

- ☐ Chapter 1 Introduction
- ☐ Chapter 2 Schematic Capture
- ☐ Chapter 3 Pre-layout Simulation
- ☐ Chapter 4 Layout Creation
- ☐ Chapter 5 Physical Verification
- ☐ Chapter 6 Post-layout Simulation

# Contents

- ☒ **Chapter 1 Introduction**
- ☐ Chapter 2 Schematic Capture
- ☐ Chapter 3 Pre-layout Simulation
- ☐ Chapter 4 Layout Creation
- ☐ Chapter 5 Physical Verification
- ☐ Chapter 6 Post-layout Simulation

# Introduction

The major purpose of this user guide is to introduce the basic usage of a TSMC's PDK for those 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

- ❑ Chapter 1 Introduction
- ❑ **Chapter 2 Schematic Capture**
- ❑ Chapter 3 Pre-layout Simulation
- ❑ Chapter 4 Layout Creation
- ❑ Chapter 5 Physical Verification
- ❑ Chapter 6 Post-layout Simulation

# 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 achieved 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.

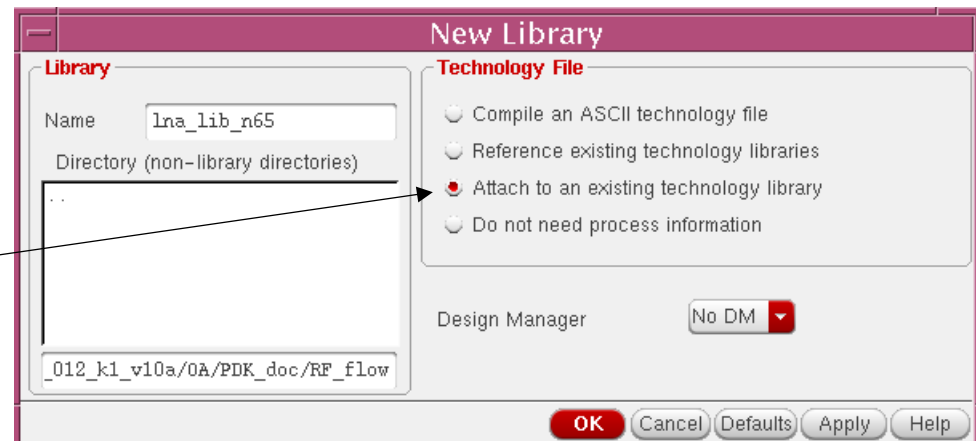
1) In the CIW, select “File->New->Library

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

Enter the new library name into the Name field.



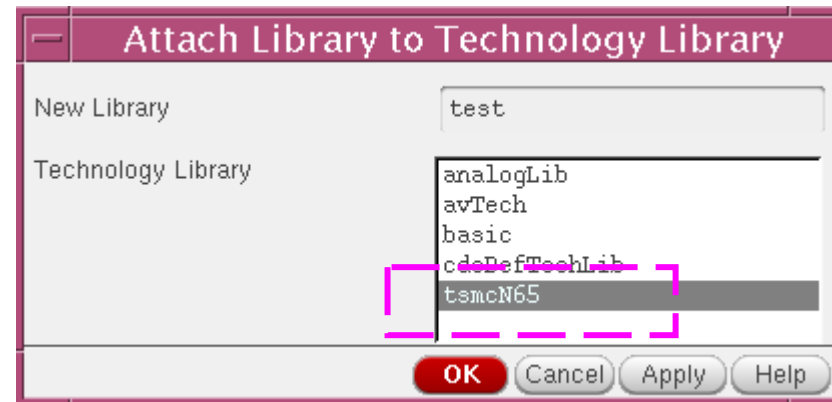
Select “Attach to an existing techfile”

The "New Library" dialog box has two main sections: "Library" and "Technology File".

- Library Section:**
  - Name:** A text field containing "lna\_lib\_n65".
  - Directory (non-library directories):** A list box showing a file path: "\_012\_k1\_v10a/0A/PDK\_doc/RF\_flow".
- Technology File Section:**
  - Four radio buttons:
    - Compile an ASCII technology file
    - Reference existing technology libraries
    - Attach to an existing technology library** (selected)
    - Do not need process information
  - Design Manager:** A dropdown menu currently set to "No DM".

Buttons at the bottom: OK, Cancel, Defaults, Apply, Help.



The "Attach Library to Technology Library" dialog box has two main sections: "New Library" and "Technology Library".

- New Library:** A text field containing "test".
- Technology Library:** A list box showing several technology libraries:
  - analogLib
  - avTech
  - basic
  - cdeDefTechLib
  - tsmcN65** (highlighted with a pink dashed box)

Buttons at the bottom: OK, Cancel, Apply, Help.

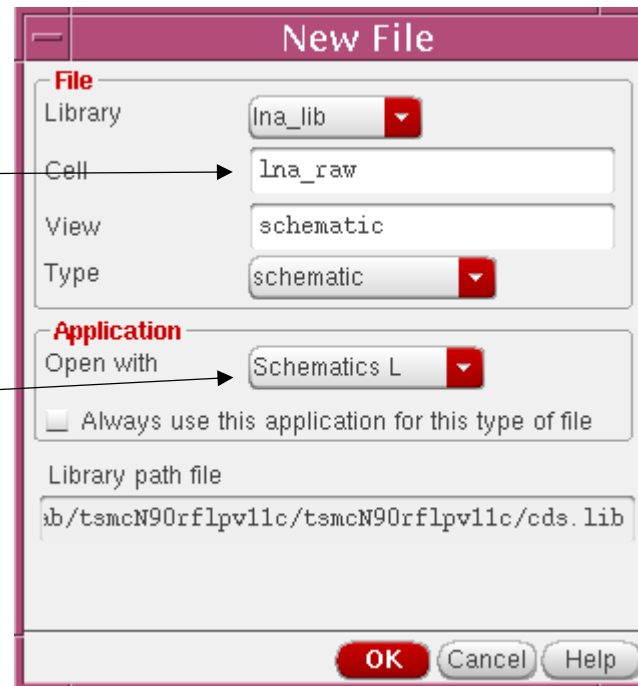
# Creating a design

## 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”



**New File**

**File**

Library: lna\_lib

Cell: lna\_raw

View: schematic

Type: schematic

**Application**

Open with: Schematics L

☐ Always use this application for this type of file

Library path file: ab/tsmcN90rflpv11c/tsmcN90rflpv11c/cds.lib

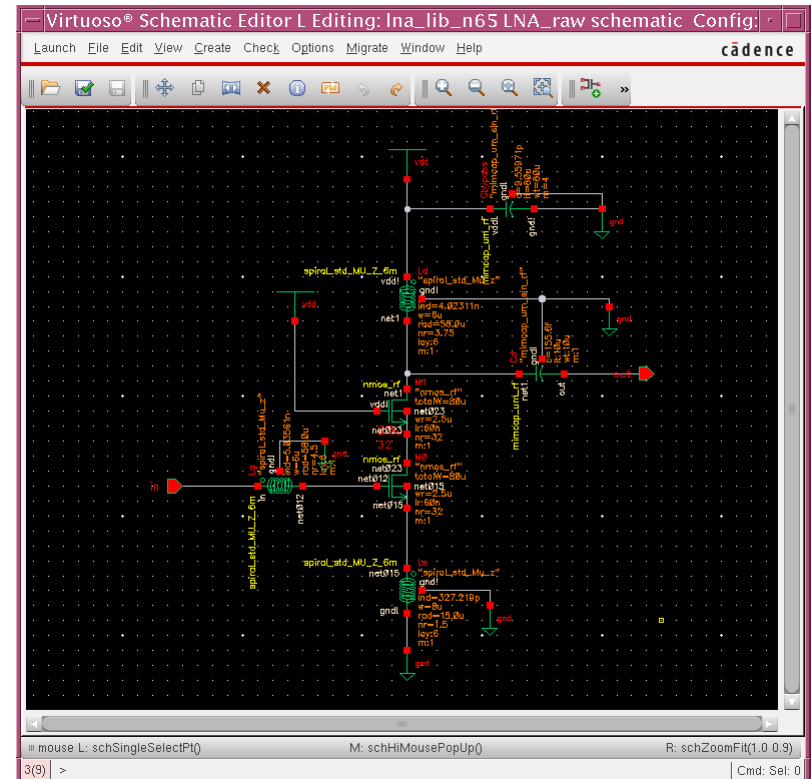
OK Cancel Help

# Creating a design

## Adding Components to a Schematic

Build the **Ina\_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.



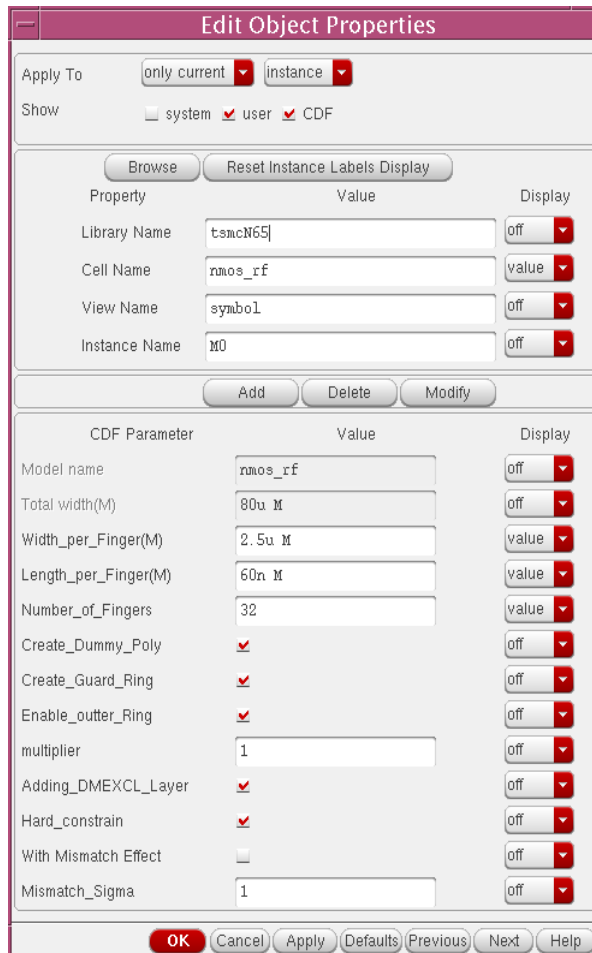
Another way to fill in the Add Instance form is to click on the Browse button. This button opens up a Library Browser from which you can select components to place your left mouse button.

# Creating a design

Component	Library Name	Cell Name	Model Name	Properties
MO	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:



**Edit Object Properties**

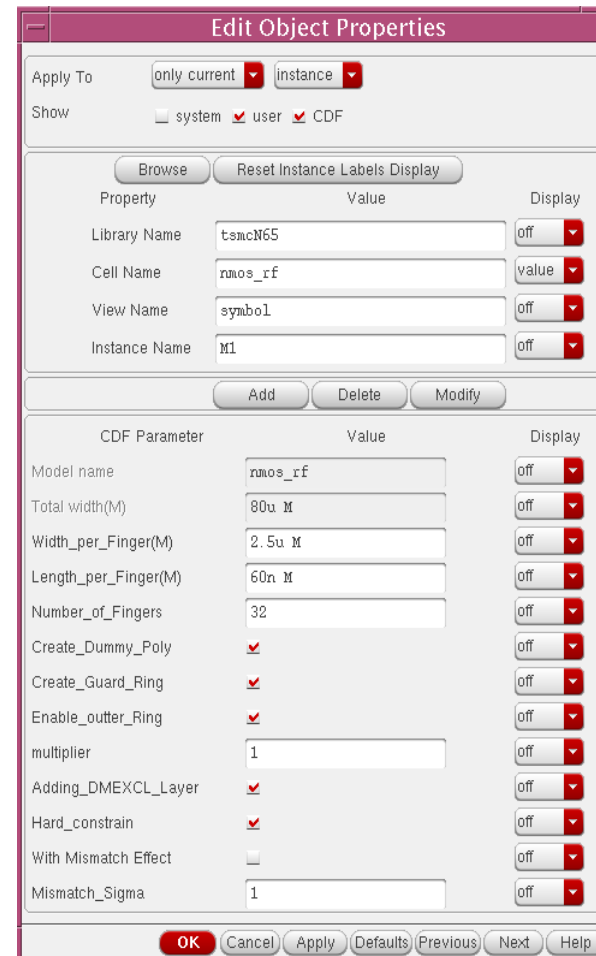
Apply To:

Show: ☐ system ☒ user ☒ CDF

Property	Value	Display
Library Name	tsmcN65	off
Cell Name	rmos_rf	value
View Name	symbol	off
Instance Name	M0	off

CDF Parameter	Value	Display
Model name	rmos_rf	off
Total width(M)	80u M	off
Width_per_Finger(M)	2.5u M	value
Length_per_Finger(M)	60n M	value
Number_of_Fingers	32	value
Create_Dummy_Poly	<input checked="" type="checkbox"/>	off
Create_Guard_Ring	<input checked="" type="checkbox"/>	off
Enable_outter_Ring	<input checked="" type="checkbox"/>	off
multiplier	1	off
Adding_DMEXCL_Layer	<input checked="" type="checkbox"/>	off
Hard_constrain	<input checked="" type="checkbox"/>	off
With Mismatch Effect	<input type="checkbox"/>	off
Mismatch_Sigma	1	off

M0



**Edit Object Properties**

Apply To:

Show: ☐ system ☒ user ☒ CDF

Property	Value	Display
Library Name	tsmcN65	off
Cell Name	rmos_rf	value
View Name	symbol	off
Instance Name	M1	off

CDF Parameter	Value	Display
Model name	rmos_rf	off
Total width(M)	80u M	off
Width_per_Finger(M)	2.5u M	off
Length_per_Finger(M)	60n M	off
Number_of_Fingers	32	off
Create_Dummy_Poly	<input checked="" type="checkbox"/>	off
Create_Guard_Ring	<input checked="" type="checkbox"/>	off
Enable_outter_Ring	<input checked="" type="checkbox"/>	off
multiplier	1	off
Adding_DMEXCL_Layer	<input checked="" type="checkbox"/>	off
Hard_constrain	<input checked="" type="checkbox"/>	off
With Mismatch Effect	<input type="checkbox"/>	off
Mismatch_Sigma	1	off

M1

### Edit Object Properties

Apply To: only current instance

Show: ☐ system ☒ user ☒ CDF

Browse Reset Instance Labels Display

Property	Value	Display
Library Name	tsmcN65	off
Cell Name	mimcap_um_rf	value
View Name	symbol	off
Instance Name	Cd	off

Add Delete Modify

CDF Parameter	Value	Display
Model name	mimcap_um_sin_rf	off
select CAP	MIM_1.5fF	off
Entry_mode	I_&_w	off
Approx. capacitance(F)	155.6f F	off
Length(M)	10u M	off
Width(M)	10u M	off
multiplier	1	off
Hard_constrain	<input checked="" type="checkbox"/>	off
With Mismatch Effect	<input type="checkbox"/>	off

OK Cancel Apply Defaults Previous Next Help

Cd

### Edit Object Properties

Apply To: only current instance

Show: ☐ system ☒ user ☒ CDF

Browse Reset Instance Labels Display

Property	Value	Display
Library Name	tsmcN65	off
Cell Name	mimcap_um_rf	value
View Name	symbol	off
Instance Name	Cbypass	off

Add Delete Modify

CDF Parameter	Value	Display
Model name	mimcap_um_sin_rf	off
select CAP	MIM_1.5fF	off
Entry_mode	I_&_w	off
Approx. capacitance(F)	9.55971p F	off
Length(M)	80u M	off
Width(M)	80u M	off
multiplier	4	off
Hard_constrain	<input checked="" type="checkbox"/>	off
With Mismatch Effect	<input type="checkbox"/>	off

OK Cancel Apply Defaults Previous Next Help

Cbypass

### Edit Object Properties

Apply To:

Show: ☐ system ☒ user ☒ CDF

Property	Value	Display
Library Name	tsmcN65	off
Cell Name	spiral_std_MU_Z	value
View Name	symbol	off
Instance Name	Ls	off

CDF Parameter	Value	Display
Model name	spiral_std_Mu_z	off
Inductor_Width_(M)	6u	off
Inner_Radius(M)	15.0u M	off
Number_Of_Turns	1.5	off
Guard_Ring_Distances_(M)	50u M	off
temp(C)	27 c	off
freq(Hz)	2.4G Hz	off

Plot_start_freq(Hz)	1G Hz	off
Plot_stop_freq(Hz)	10G Hz	off
Plot_step_freq(Hz)	1G Hz	off

Approx. inductance(H)	327.219p H	off
Q_factor	10.0464	off
Inductor_area_width(M)	184.000000u M	off
Inductor_area_length(M)	179.5u M	off
TopMetal	9	off
multiplier	1	off

Ls

### Edit Object Properties

Apply To:

Show: ☐ system ☒ user ☒ CDF

Property	Value	Display
Library Name	tsmcN65	off
Cell Name	spiral_std_MU_Z	value
View Name	symbol	off
Instance Name	Lg	off

CDF Parameter	Value	Display
Model name	spiral_std_Mu_z	off
Inductor_Width_(M)	6u	off
Inner_Radius(M)	58.0u M	off
Number_Of_Turns	4.5	off
Guard_Ring_Distances_(M)	50u M	off
temp(C)	27 c	off
freq(Hz)	2.4G Hz	off

Plot_start_freq(Hz)	1G Hz	off
Plot_stop_freq(Hz)	10G Hz	off
Plot_step_freq(Hz)	1G Hz	off

Approx. inductance(H)	5.03561n H	off
Q_factor	11.4194	off
Inductor_area_width(M)	324.000000u M	off
Inductor_area_length(M)	319.5u M	off
TopMetal	9	off
multiplier	1	off

Lg

### Edit Object Properties

Apply To:

Show: ☐ system ☒ user ☒ CDF

Property	Value	Display
Library Name	tsmcN65	off
Cell Name	spiral_std_MU_Z	value
View Name	symbol	off
Instance Name	Ld	off

CDF Parameter	Value	Display
Model name	spiral_std_Mu_z	off
Inductor_Width_(M)	6u	off
Inner_Radius(M)	58.0u M	off
Number_Of_Turns	3.75	off
Guard_Ring_Distances_(M)	50u M	off
temp(C)	27 c	off
freq(Hz)	5.6G Hz	off

Plot_start_freq(Hz)	1G Hz	off
Plot_stop_freq(Hz)	10G Hz	off
Plot_step_freq(Hz)	1G Hz	off

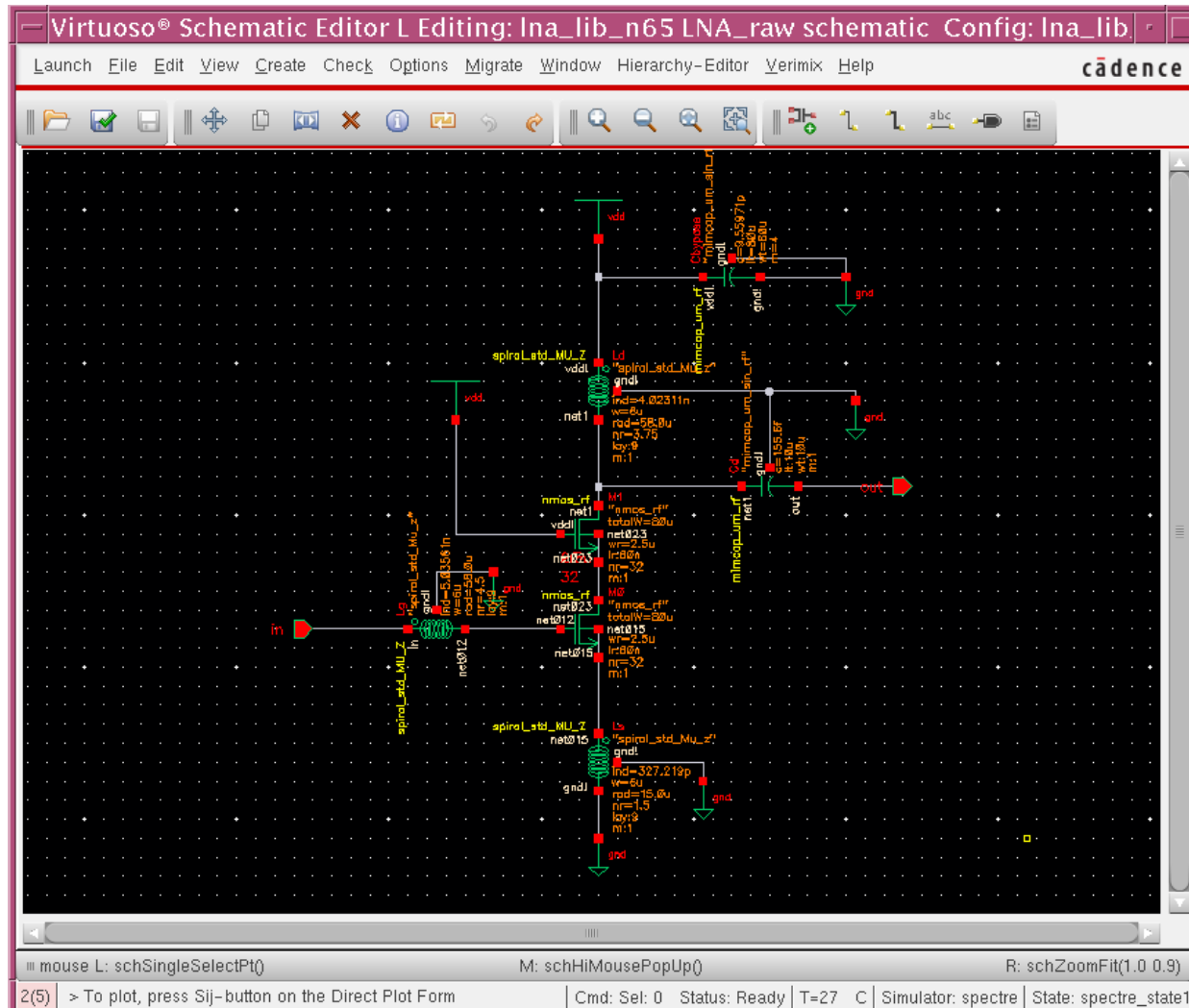
Approx. inductance(H)	4.02311n H	off
Q_factor	8.44657	off
Inductor_area_width(M)	306.000000u M	off
Inductor_area_length(M)	310.5u M	off
TopMetal	9	off
multiplier	1	off

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.**

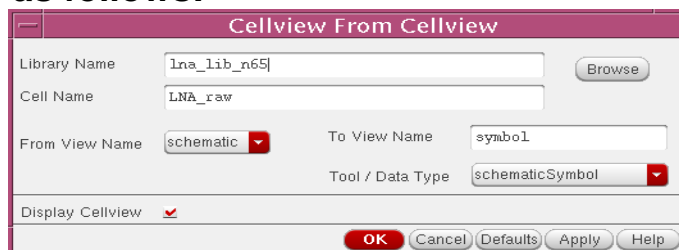




# Creating a symbol

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:



Cellview From Cellview

Library Name: lna\_lib\_n65 [Browse]

Cell Name: LNA\_raw

From View Name: schematic [v] To View Name: symbol

Tool / Data Type: schematicSymbol [v]

Display Cellview: ☒

[OK] [Cancel] [Defaults] [Apply] [Help]

3. Edit the Pins according to the following figure.



Symbol Generation Options

Library Name: lna\_lib\_n65 Cell Name: LNA\_raw View Name: symbol

Pin Specifications:

Pin	Specification	Attributes
Left Pins	in	[List]
Right Pins	out	[List]
Top Pins		[List]
Bottom Pins		[List]

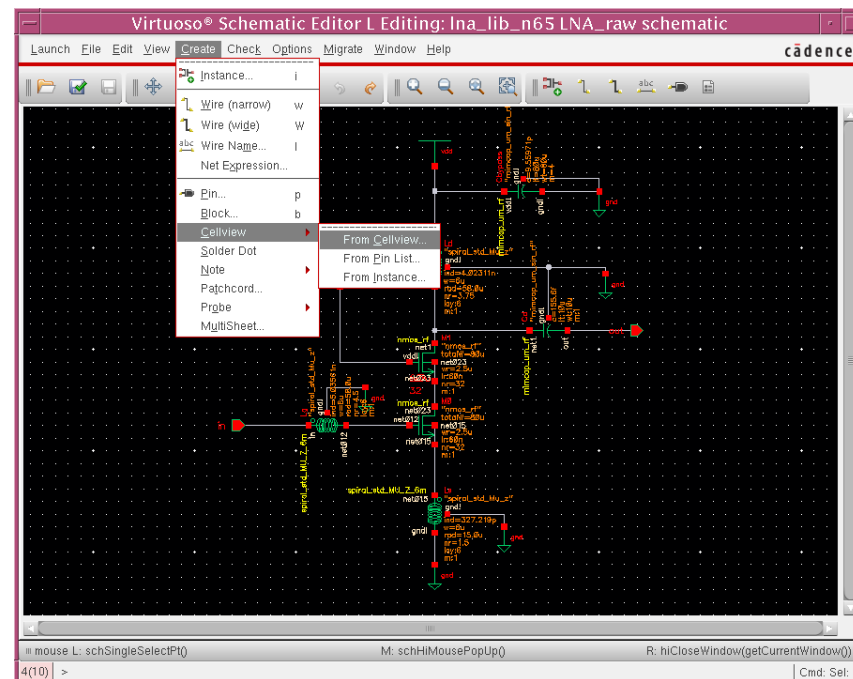
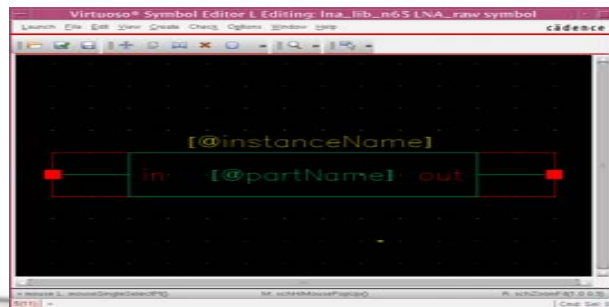
Exclude Inherited Connection Pins:

☒ None ☐ All ☐ Only these: [ ]

Load/Save [ ] Edit Attributes [ ] Edit Labels [ ] Edit Properties [ ]

[OK] [Cancel] [Apply] [Help]

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

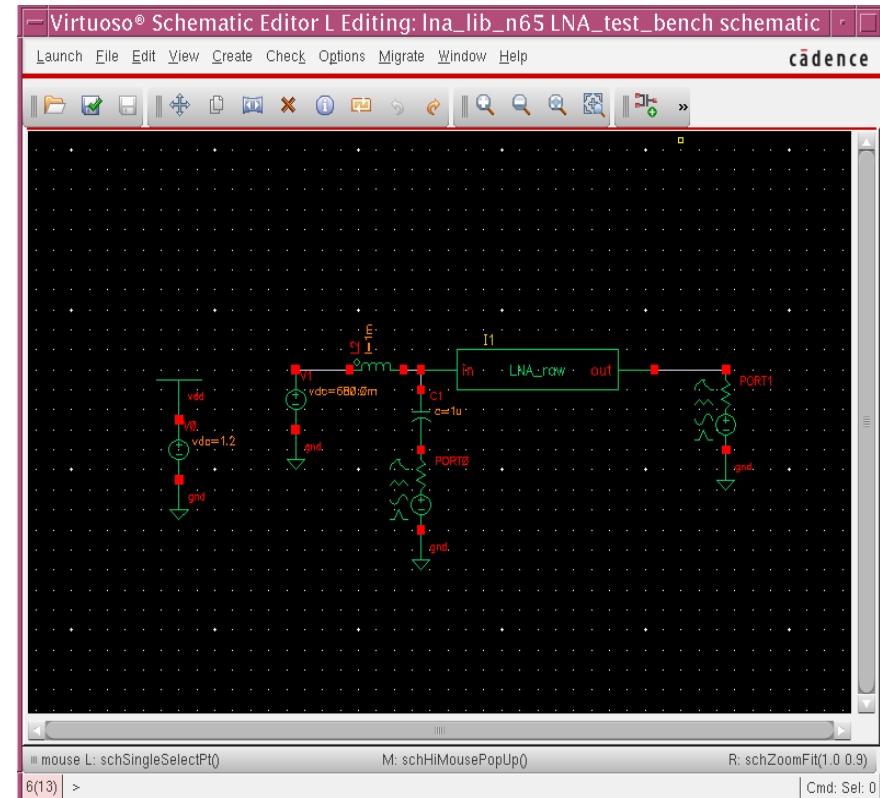


# Creating a test fixture

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
Ina_lib_n65	LNA_raw	Core design

Component table



LNA Test bench

# Contents

- ❑ Chapter 1 Introduction
- ❑ Chapter 2 Schematic Capture
- ❑ **Chapter 3 Pre-layout Simulation**
- ❑ Chapter 4 Layout Creation
- ❑ Chapter 5 Physical Verification
- ❑ Chapter 6 Post-layout Simulation

# Pre-layout Simulation

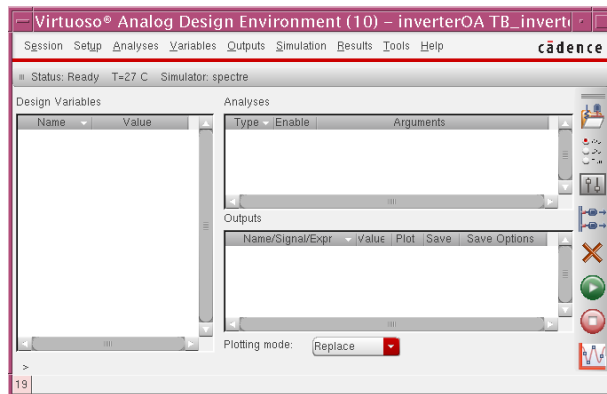
After completing the creation of test fixture schematic-capture, we need to run simulation to check its function and performance. In this chapter, we will use Spectre as the simulator.

- Using Spectre simulator
- LNA Performance
- Corner simulation

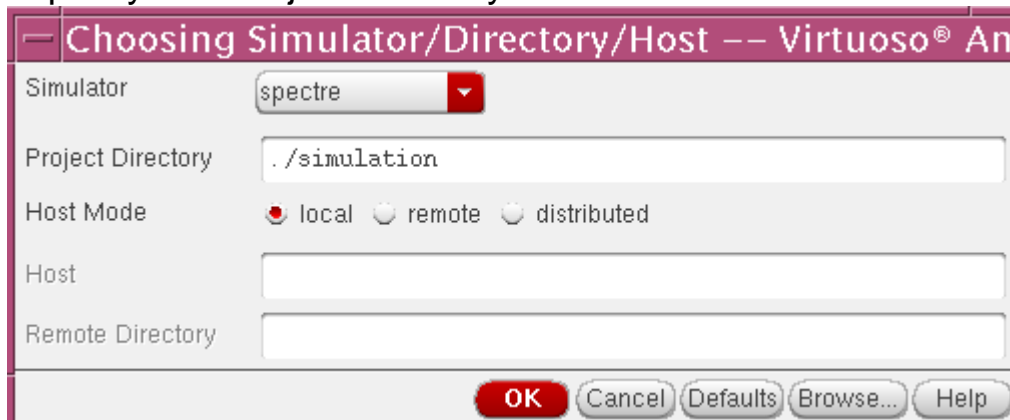
# Using Spectre Simulator

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.



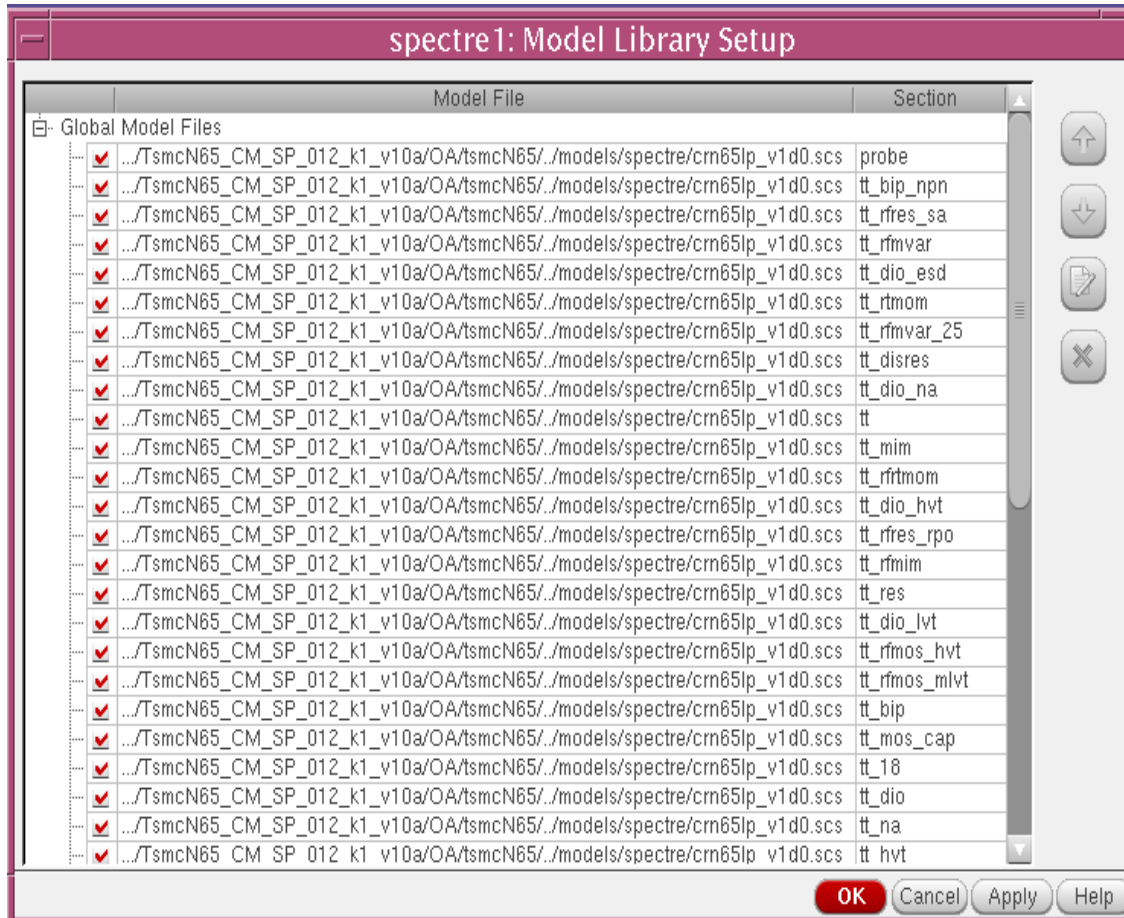
- 2) Select simulator  
Setup->simulator/Director/Host and set simulator to “spectre” and specify the “Project Directory” to “./simulation”.



# 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**

Choosing Analyses -- Virtuoso® Analog Design Environment

Analysis ☐ tran ☒ dc ☐ ac ☐ noise  
☐ xf ☐ sens ☐ dcmatch ☐ stb  
☐ pz ☐ sp ☐ envlp ☐ pss  
☐ pac ☐ pstb ☐ pnoise ☐ pxf  
☐ psp ☐ qpss ☐ qpac ☐ qpnoise  
☐ qpxf ☐ qpsp

DC Analysis

Save DC Operating Point ☒

Sweep Variable

☐ Temperature  
☐ Design Variable  
☐ Component Parameter  
☐ Model Parameter

Enabled ☒

Options...

OK Cancel Defaults Apply Help

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.

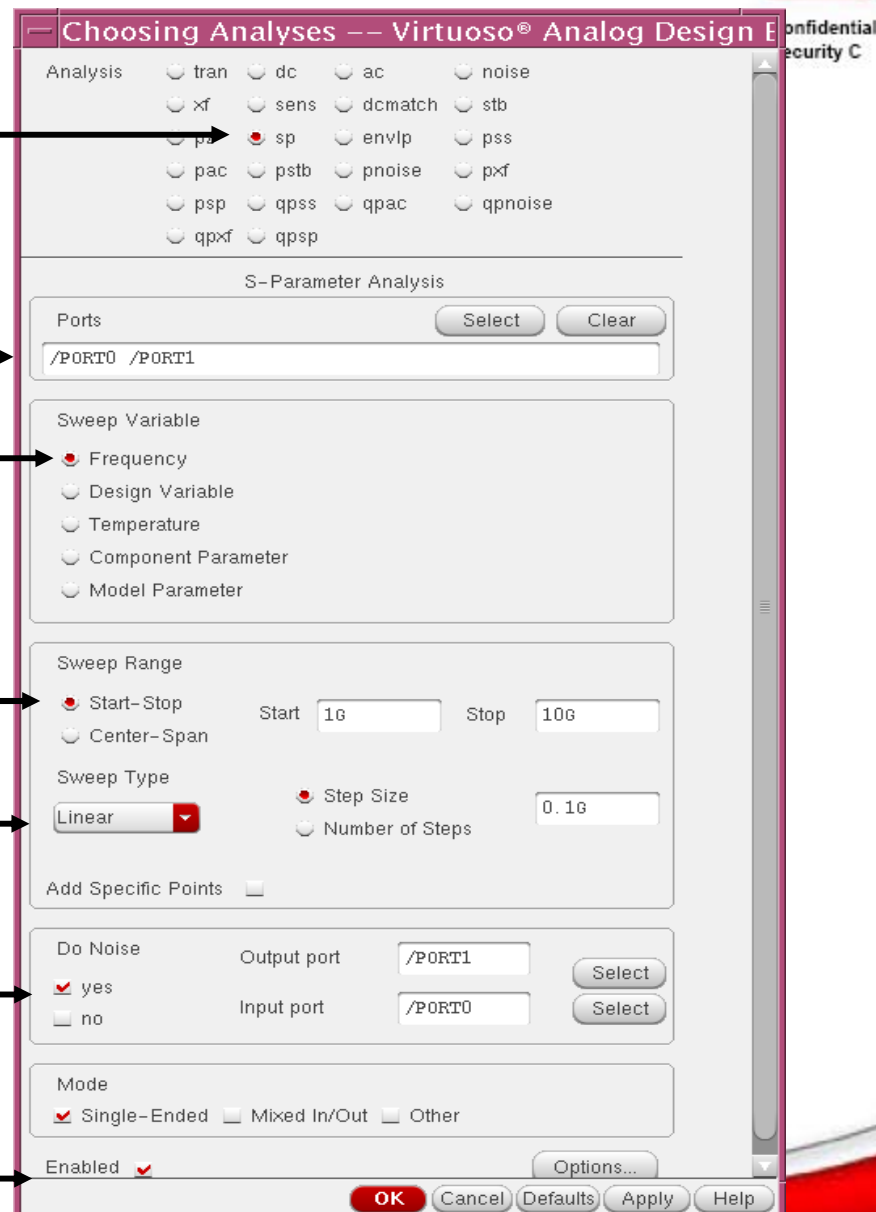
Set the Sweep Variable to **Frequency**

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

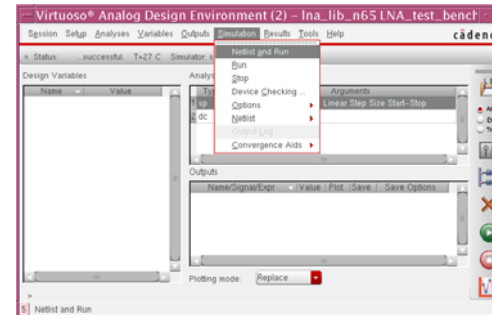




# Using Spectre Simulator (cont.)

## 7) Run simulation

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



```
// 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 verilog a
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) rmos_rf lr=60n wr=2.5u nr=32 sigr
    m=1 mismatchflag=0
  M1 (net1 vdd! net023 net023) rmos_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 gdis=
    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 verilog a
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=1m
```

```
// /home/lhtung/PDKQA/TsmcN65_CM_SP_012_k1_v10a/0A/tsmcN65/./m
cādence

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.

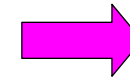
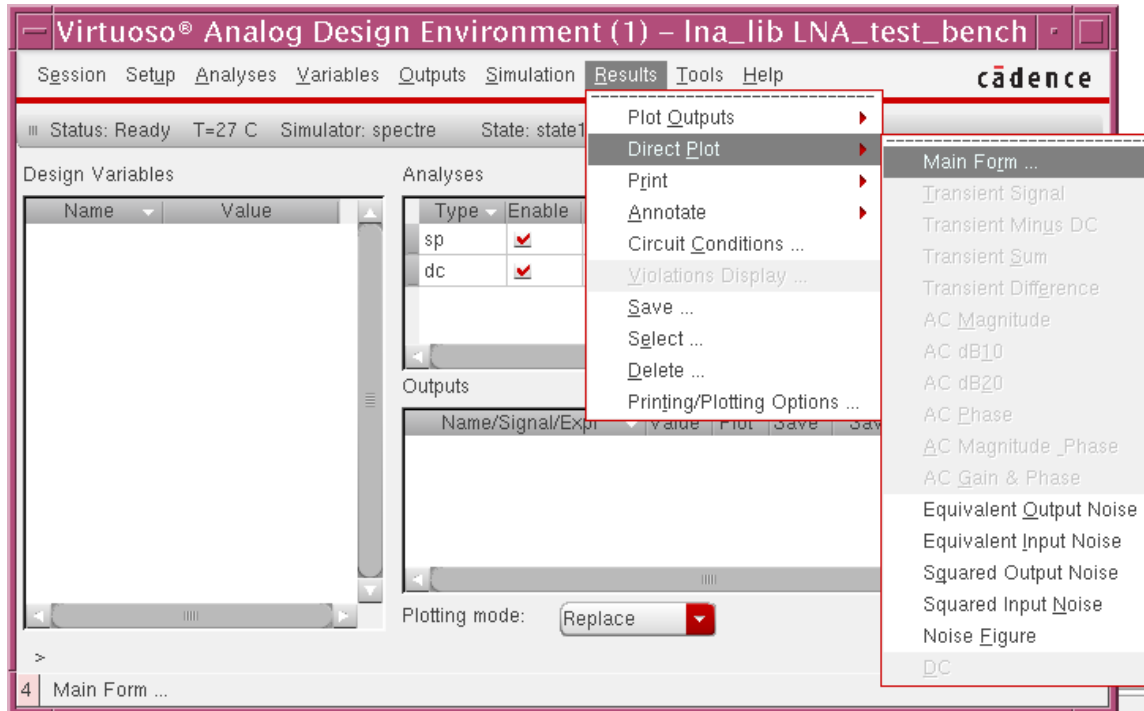
*****
S-Parameter Analysis 'sp': freq = (1 GHz -> 10 GHz)
*****
sp: freq = 1.3 GHz (3.33 %), step = 100 MHz (1.11 %)
sp: freq = 1.7 GHz (7.78 %), step = 100 MHz (1.11 %)
sp: freq = 2.2 GHz (13.3 %), step = 100 MHz (1.11 %)
sp: freq = 2.6 GHz (17.8 %), step = 100 MHz (1.11 %)
sp: freq = 3.1 GHz (23.3 %), 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.
```

# LNA Performance

After running simulation, we can see the result and performance of our design from

Results->Direct Plot->Main Form....:



**Direct Plot Form**

Plotting Mode: New SubWin

**Analysis**

☒ sp

**Function**

☒ SP   ☐ ZP   ☐ YP   ☐ HP  
☐ GD   ☐ VSWR   ☐ NFmin   ☐ Gmin  
☐ Rn   ☐ m   ☐ NF   ☐ Kf  
☐ B1f   ☐ GT   ☐ GA   ☐ GP  
☐ Gmax   ☐ Gmsg   ☐ Gmxx   ☐ ZM  
☐ NC   ☐ GAC   ☐ GPC   ☐ LSB  
☐ SSB

Description: S-Parameter

**Plot Type**

☒ Rectangular   ☐ Z-Smith   ☐ Y-Smith  
☐ Polar

**Modifier**

☐ Magnitude   ☐ Phase   ☒ dB20  
☐ Real   ☐ Imaginary

Add To Outputs ☐

> To plot, press Sij-button on this 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”

**Direct Plot Form**

Plotting Mode

**Analysis**

☒ sp

**Function**

☒ SP ☐ ZP ☐ YP ☐ HP  
☐ GD ☐ VSWR ☐ NFmin ☐ Gmin  
☐ Rn ☐ rn ☐ NF ☐ Kf  
☐ B1f ☐ GT ☐ GA ☐ GP  
☐ Gmax ☐ Gmsg ☐ Gumx ☐ ZM  
☐ NC ☐ GAC ☐ GPC ☐ LSB  
☐ SSB

Description: S-Parameter

**Plot Type**

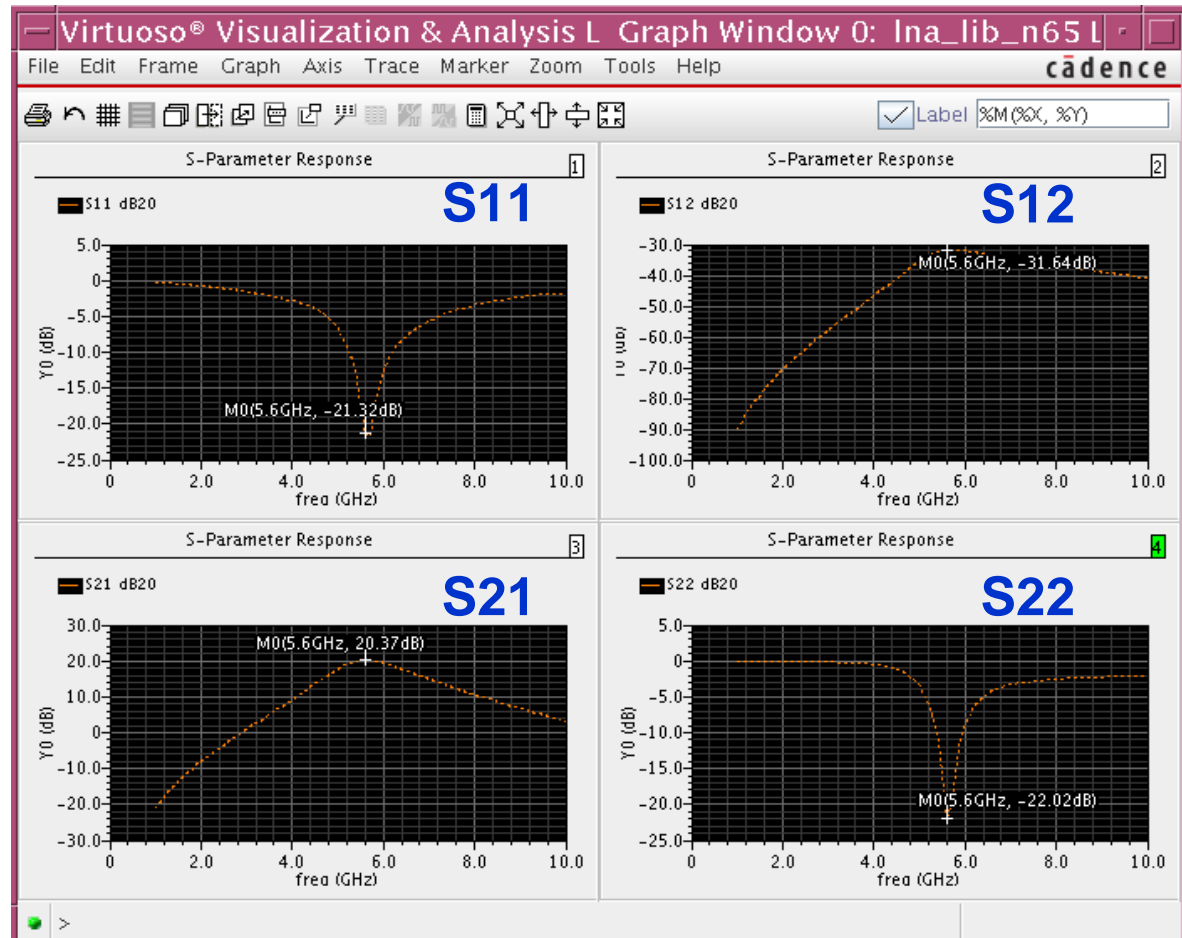
☒ Rectangular ☐ Z-Smith ☐ Y-Smith  
☐ Polar

**Modifier**

☐ Magnitude ☐ Phase ☒ dB20  
☐ Real ☐ Imaginary

Add To Outputs ☐

> To plot, press Sij-button on this form...



**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.

**Direct Plot Form**

Plotting Mode: New SubWin

**Analysis**

☒ sp

**Function**

☒ SP   ☐ ZP   ☐ YP   ☐ HP  
☐ GD   ☐ VSWR   ☐ NFmin   ☐ Gmin  
☐ Rn   ☐ rn   ☐ NF   ☐ Kf  
☐ B1f   ☐ GT   ☐ GA   ☐ GP  
☐ Gmax   ☐ Gmsg   ☐ Gumx   ☐ ZM  
☐ NC   ☐ GAC   ☐ GPC   ☐ LSB  
☐ SSB

Description: S-Parameter

**Plot Type**

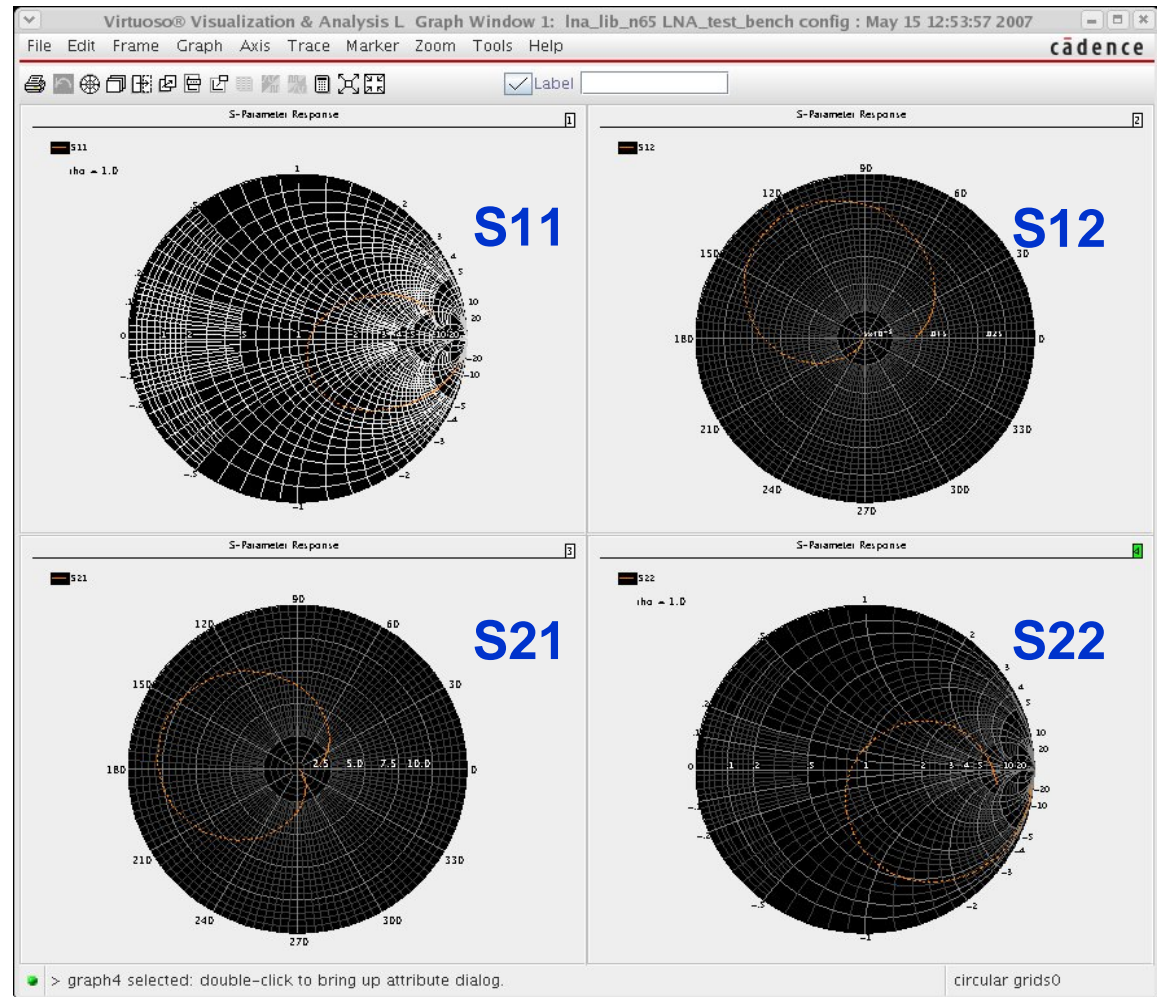
☐ Rectangular   ☒ Z-Smith   ☐ Y-Smith  
☐ Polar

S11   S12  
S21   S22

Add To Outputs ☐

> To plot, press Sij-button on this form...

OK Cancel Help



**In this plot, “Plot type” is set to “Polar” on S21 and S12**



# LNA Performance (cont.)

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

Direct Plot Form

Plotting Mode

Append

Analysis

☒ sp

Function

☐ SP ☐ ZP ☐ YP ☐ HP  
☐ GD ☐ VSWR ☐ NFmin ☐ Gmin  
☐ Rn ☐ rn ☐ NF ☐ Kf  
☐ B1f ☐ GT ☐ GA ☐ GP  
☐ Gmax ☐ Gmsg ☐ Gumx ☐ ZM  
☐ NC ☐ GAC ☐ GPC ☐ LSB  
☒ SSB

Description: Source Stability Circles

Plot Type

☒ Z-Smith ☐ Y-Smith

Frequency Range (Hz)

Start

10

Stop

100

Step

0.10

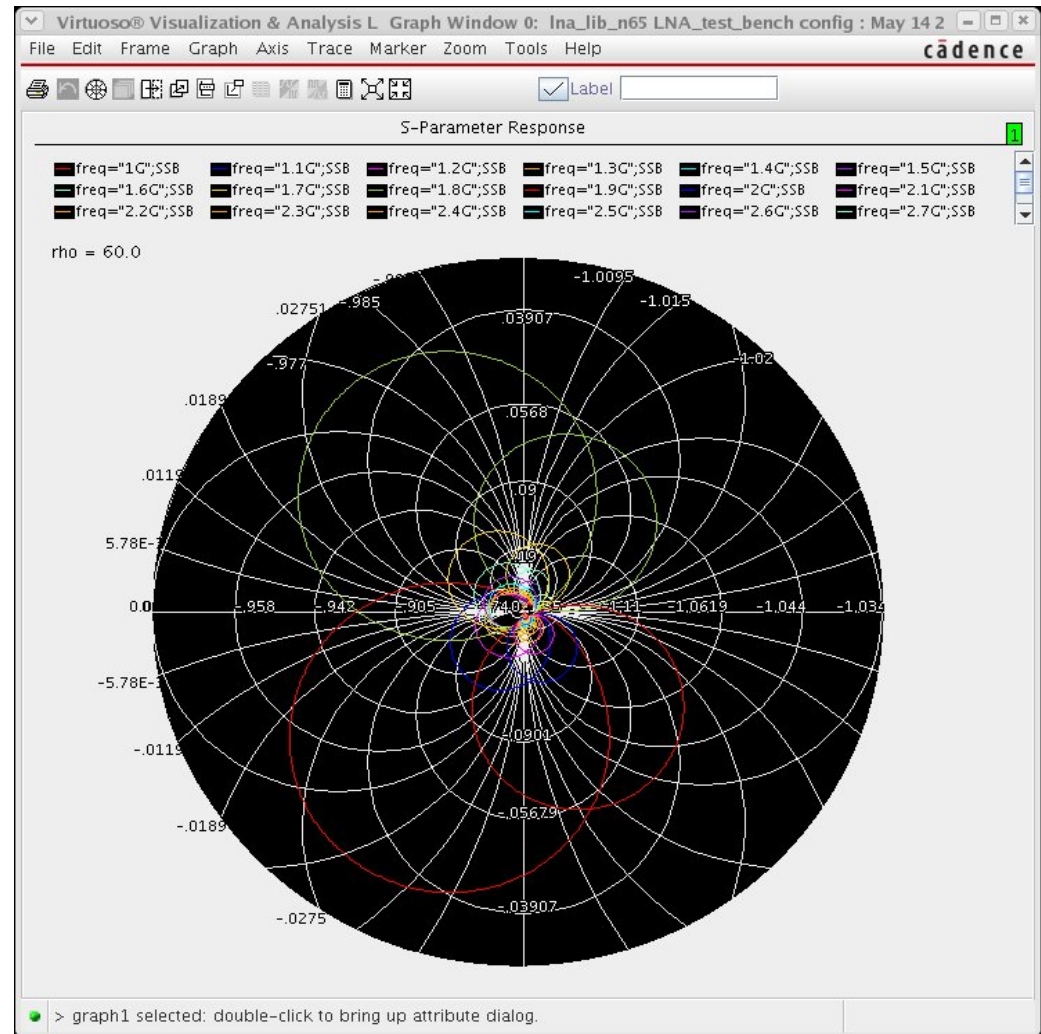
Add To Outputs

Plot

OK

Cancel

Help



# LNA Performance (cont.)

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

Direct Plot Form

Plotting Mode

Append

Analysis

☒ sp

Function

☐ SP ☐ ZP ☐ YP ☐ HP  
☐ GD ☐ VSWR ☐ NFmin ☐ Gmin  
☐ Rn ☐ m ☐ NF ☐ Kf  
☐ B1f ☐ GT ☐ GA ☐ GP  
☐ Gmax ☐ Gmsg ☐ Gumx ☐ ZM  
☐ NC ☐ GAC ☐ GPC ☒ LSB  
☐ SSB

Description: Load Stability Circles

Plot Type

☒ Z-Smith ☐ Y-Smith

Frequency Range (Hz)

Start

10

Stop

100

Step

0.10

Add To Outputs

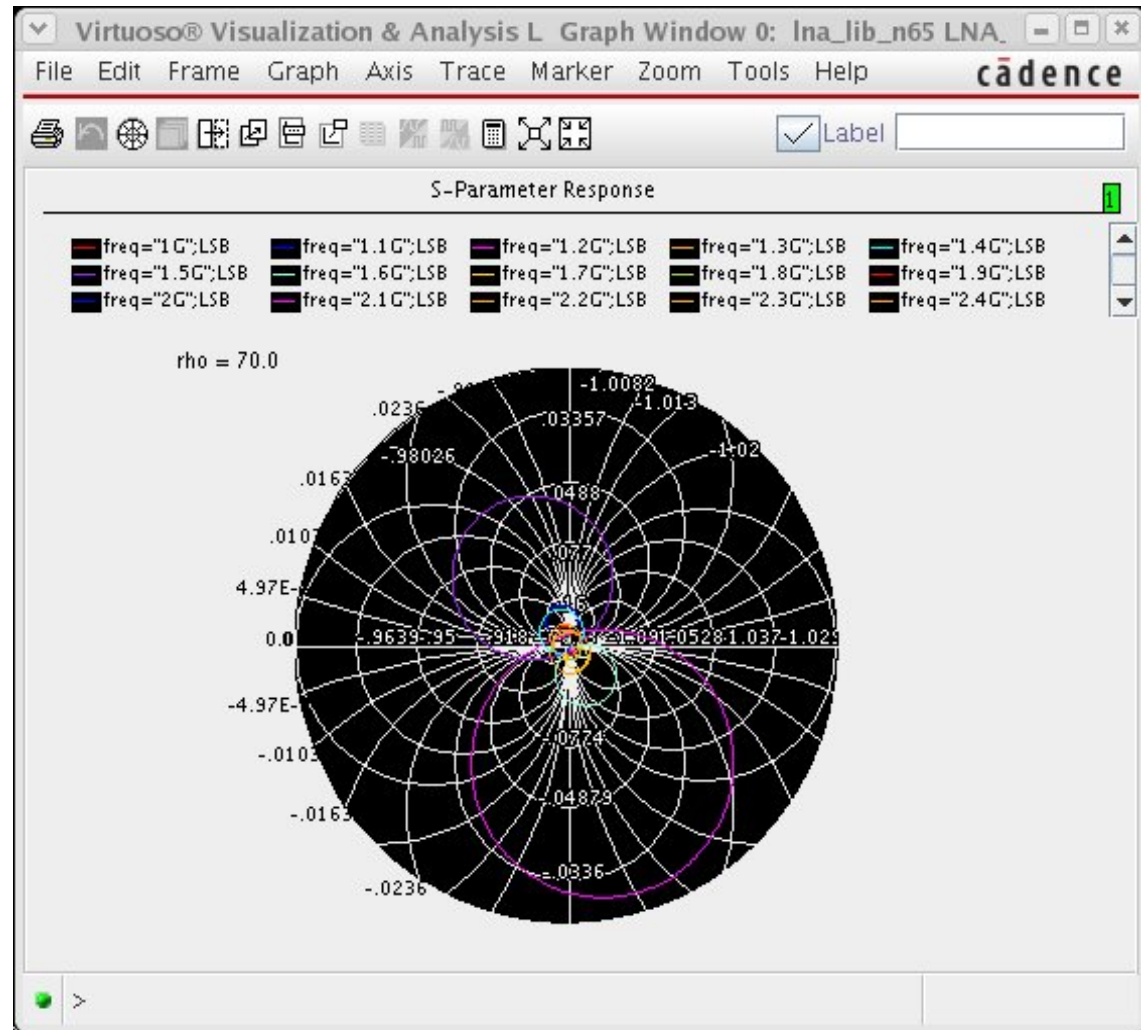
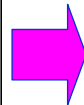
Plot

> Press plot button on this form...

OK

Cancel

Help



# 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”

**Direct Plot Form**

Plotting Mode: Replace

**Analysis**

sp

**Function**

☐ SP ☐ ZP ☐ YP ☐ HP  
☐ GD ☐ VSWR ☐ NFmin ☐ Gmin  
☐ Rn ☐ rn ☒ NF ☐ Kf  
☐ B1f ☐ GT ☐ GA ☐ GP  
☐ Gmax ☐ Gmsg ☐ Gumx ☐ ZM  
☐ NC ☐ GAC ☐ GPC ☐ LSB  
☐ SSB

Description: Noise Figure

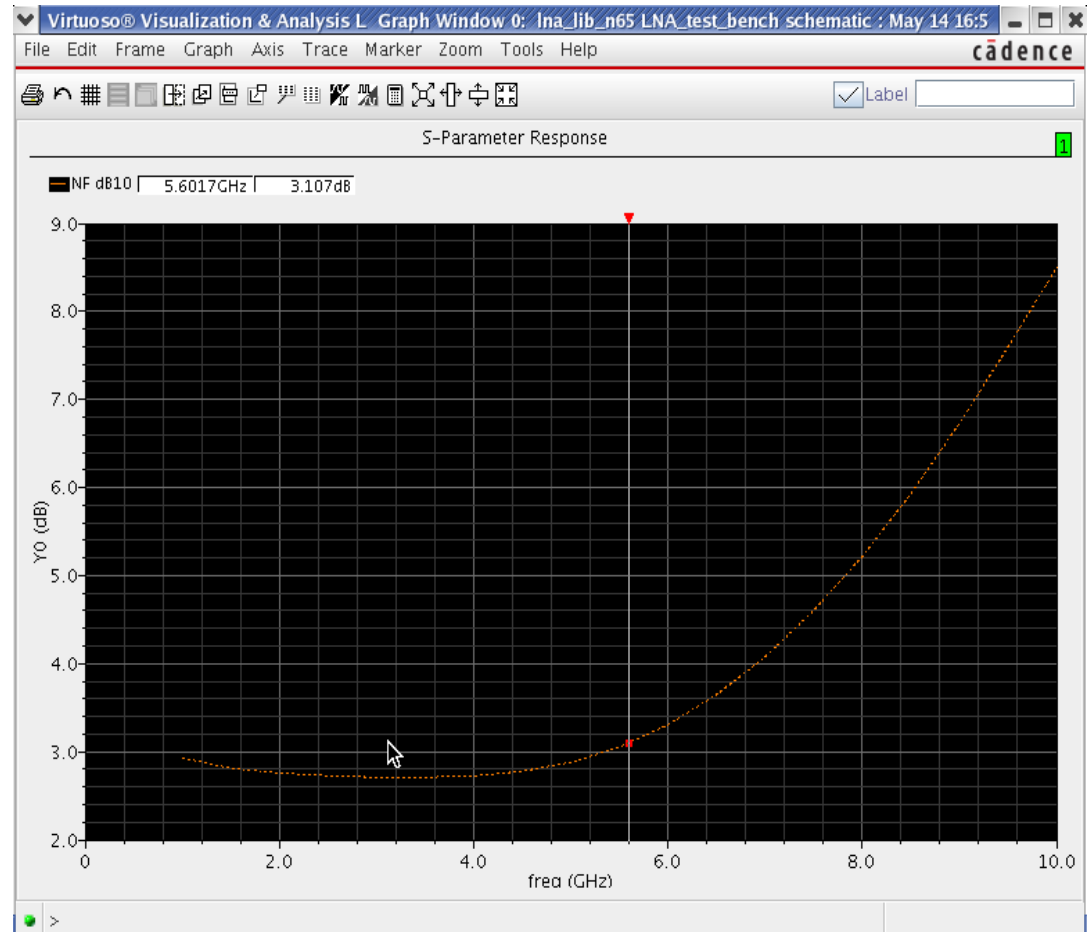
**Modifier**

☐ Magnitude ☒ dB10

Add To Outputs Plot

> Press plot button on this form...

OK Cancel Help



**NF=3.107dB @5.6GHz**

# LNA Performance (3)

**Choosing Analyses – Virtuoso® Analog Design Environment (3)**

Analysis: ☐ tran ☐ dc ☐ ac ☐ noise  
☐ xf ☐ sens ☐ dcmatch ☐ stb  
☐ pz ☐ sp ☐ envlp ☒ pss  
☐ pac ☐ pstb ☐ pnoise ☐ pxf  
☐ psp ☐ qpss ☐ qpac ☐ qpnoise  
☐ qpxf ☐ qpsp

Periodic Steady State Analysis  
 Engine: ☐ Shooting ☐ Flexible Balance

Fundamental Tones

#	Name	Expr	Value	Signal	SrcId
1	F1	frf	5.60	Large	PORT0

Clear/Add Delete Update From Schematic

Beat Frequency ☒ Beat Period ☐ 5.60 Auto Calculate ☒

Output harmonics  
 Number of harmonics 3

Accuracy Defaults (errpreset)  
☐ conservative ☒ moderate ☐ liberal  
 Additional Time for Stabilization (tstab)   
 Save Initial Transient Results (saveinit) ☐ no ☐ yes

Oscillator ☐

Sweep ☒ Frequency Variable? ☒ no ☐ yes  
 Variable  Variable Name prf Select Design Variable

Sweep Range  
☒ Start-Stop Start -40 Stop -10  
☐ Center-Span

Sweep Type  
☒ Linear ☐ Logarithmic  
☒ Step Size 5  
☐ Number of Steps

Add Specific Points ☐

Enabled ☒ Options...

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...:

**Analysis**

☒ pss

**Function**

☐ Voltage ☐ Current  
☐ Power ☐ Voltage Gain  
☐ Current Gain ☐ Power Gain  
☐ Transconductance ☐ Transimpedance  
☒ Compression Point ☐ IPN Curves  
☐ Power Contours ☐ Reflection Contours  
☐ Harmonic Frequency ☐ Power Added Eff.  
☐ Power Gain Vs Pout ☐ Comp. Vs Pout  
☐ Node Complex Imp. ☐ THD

Select Port ( fixed R(port) )

Format Output Power

Gain Compression (dB) 1

"prf" ranges from -40 to -10  
 Input Power Extrapolation Point (dBm) -40

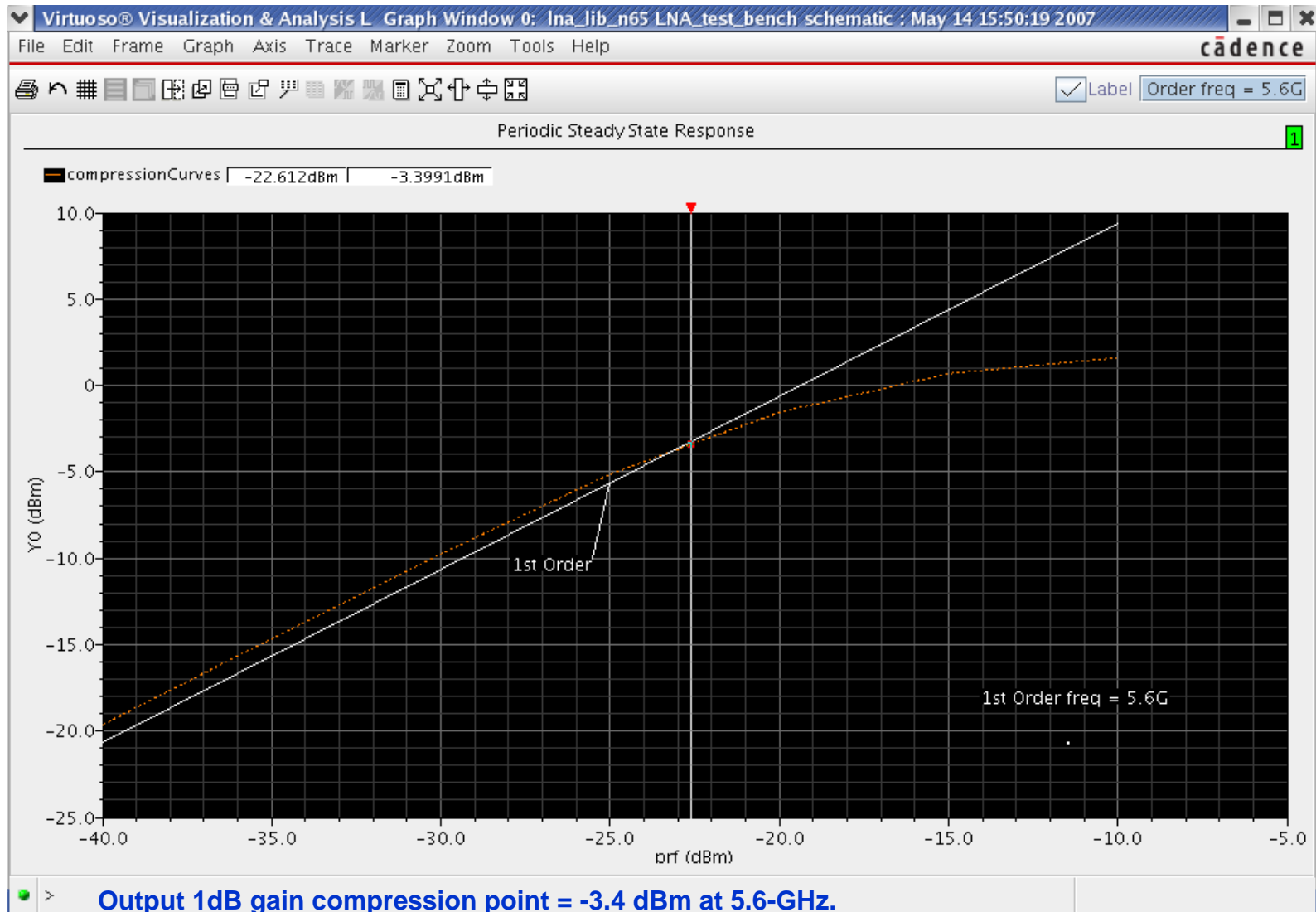
Output Referred 1dB Compression

1st Order Harmonic

0	0
1	5.60
2	11.20
3	16.80



# LNA Performance (3)



# 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...:

Choosing Analyses – Virtuoso® Analog Design Environment (3)

Analysis

☐ tran   ☐ dc   ☐ ac   ☐ noise  
☐ xf   ☐ sens   ☐ dcmatch   ☐ stb  
☐ pz   ☐ sp   ☐ envlp   ☐ pss  
☒ pac   ☐ pstb   ☐ pnoise   ☐ pxf  
☐ psp   ☐ qpss   ☐ qpac   ☐ qpnoise  
☐ qpzf   ☐ qpzp

Periodic AC Analysis

PSS Beat Frequency (Hz) 5.6G

Sweeptype default Sweep is Currently Absolute

Input Frequency Sweep Range (Hz)

Single-Point Freq 5.61G

Add Specific Points

Sidebands

Maximum sideband 3

Specialized Analyses

None

Enabled Options...

Direct Plot Form

Plotting Mode Append

Analysis

☐ pss   ☒ pac

Function

☐ Voltage   ☐ Voltage Gain  
☐ Current   ☒ IPN Curves

Select Port ( fixed R(port) )

Circuit Input Power

☐ Single Point  
☒ Variable Sweep ("prf")

"prf" ranges from -50 to -10

Input Power Extrapolation Point (dBm) -50

Input Referred IP3 Order 3rd

3rd Order Harmonic		1st Order Harmonic	
-3	11.19G	-3	11.19G
-2	5.59G	-2	5.59G
-1	10M	-1	10M
0	5.61G	0	5.61G
1	11.21G	1	11.21G
2	16.81G	2	16.81G

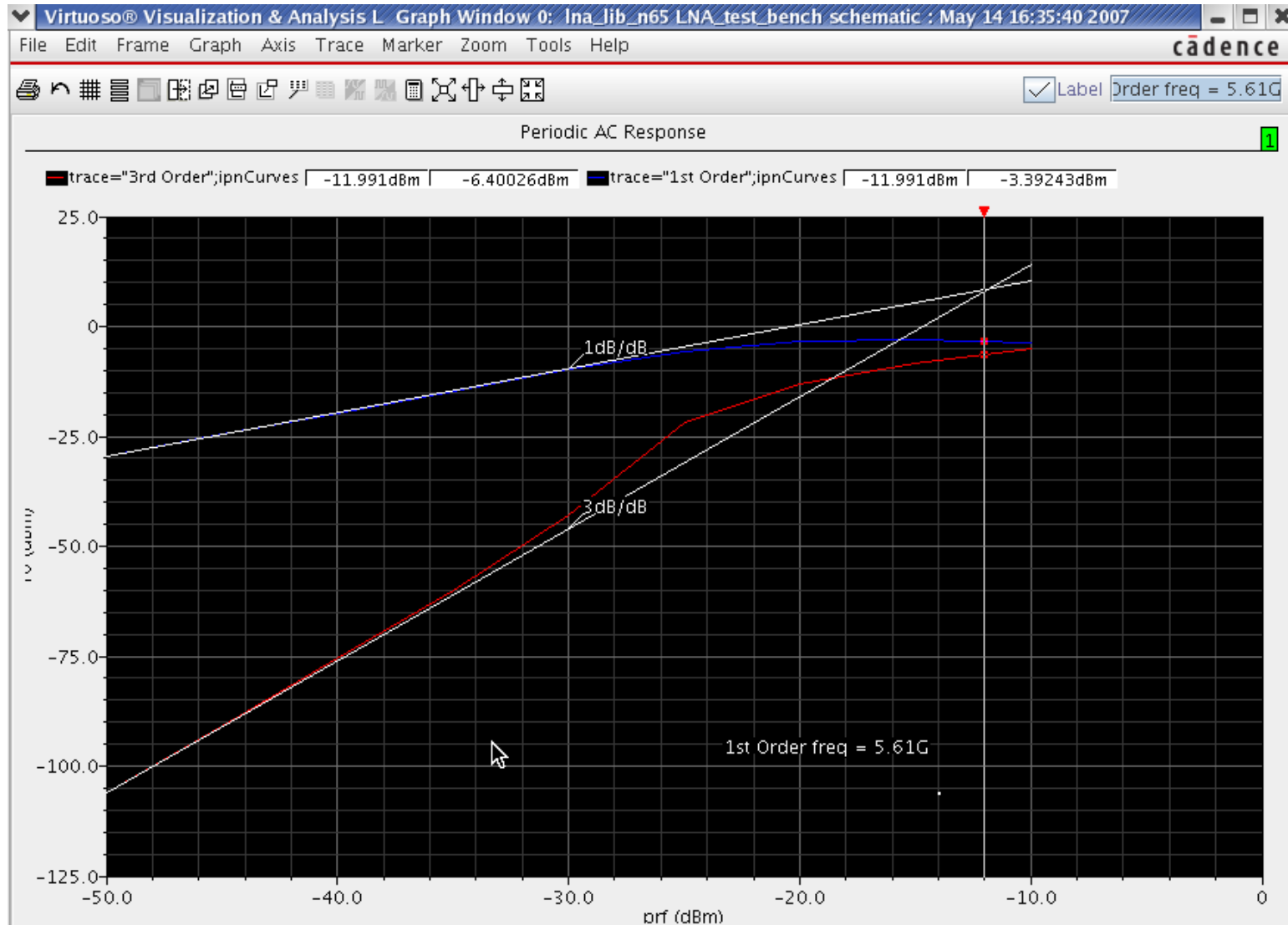
Add To Outputs Replot

freqaxis = absout

> Select Port on schematic...

OK Cancel Help

# 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

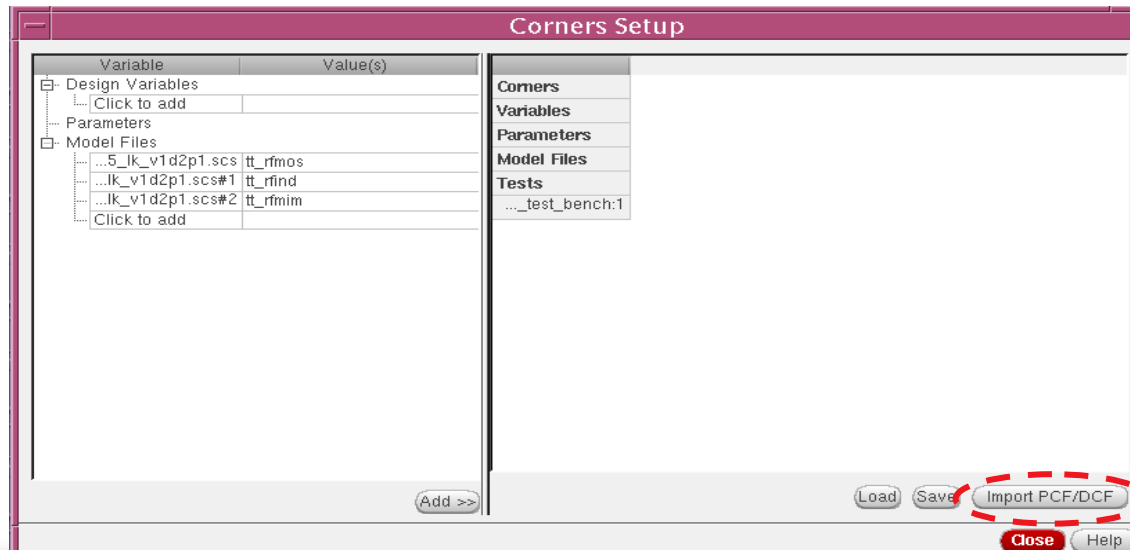
# Corner simulation

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 shift 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\_rfmoss), 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 Name : typical

Variables / Parameters

Variable	Value
Temperature	25
<b>Design Variables</b>	
Click to add	
<b>Parameters</b>	
Click to add	

>>

<<

Model Files

Model	Section
<b>Test/Custom Models</b>	
<input checked="" type="checkbox"/> ...p_v1d0.scs	probe
<input checked="" type="checkbox"/> ...p_v1d0.scs	tt_bip_npn
<input checked="" type="checkbox"/> ...p_v1d0.scs	tt_mlv
<input checked="" type="checkbox"/> ...p_v1d0.scs	tt_rfjvar
<input checked="" type="checkbox"/> ...p_v1d0.scs	tt_rfmos_mlv
<input checked="" type="checkbox"/> ...p_v1d0.scs	tt_rfmvar_hvt
<input checked="" type="checkbox"/> ...p_v1d0.scs	tt_rfmos
<input checked="" type="checkbox"/> ...p_v1d0.scs	tt_dio_na25
<input checked="" type="checkbox"/> ...p_v1d0.scs	tt_rfrtmom
<input checked="" type="checkbox"/> ...p_v1d0.scs	tt_rfmos_18
<input checked="" type="checkbox"/> ...p_v1d0.scs	tt_mos_cap_25
<input checked="" type="checkbox"/> ...p_v1d0.scs	tt_dio_33
<input checked="" type="checkbox"/> ...p_v1d0.scs	tt_rfind

Corners Setup

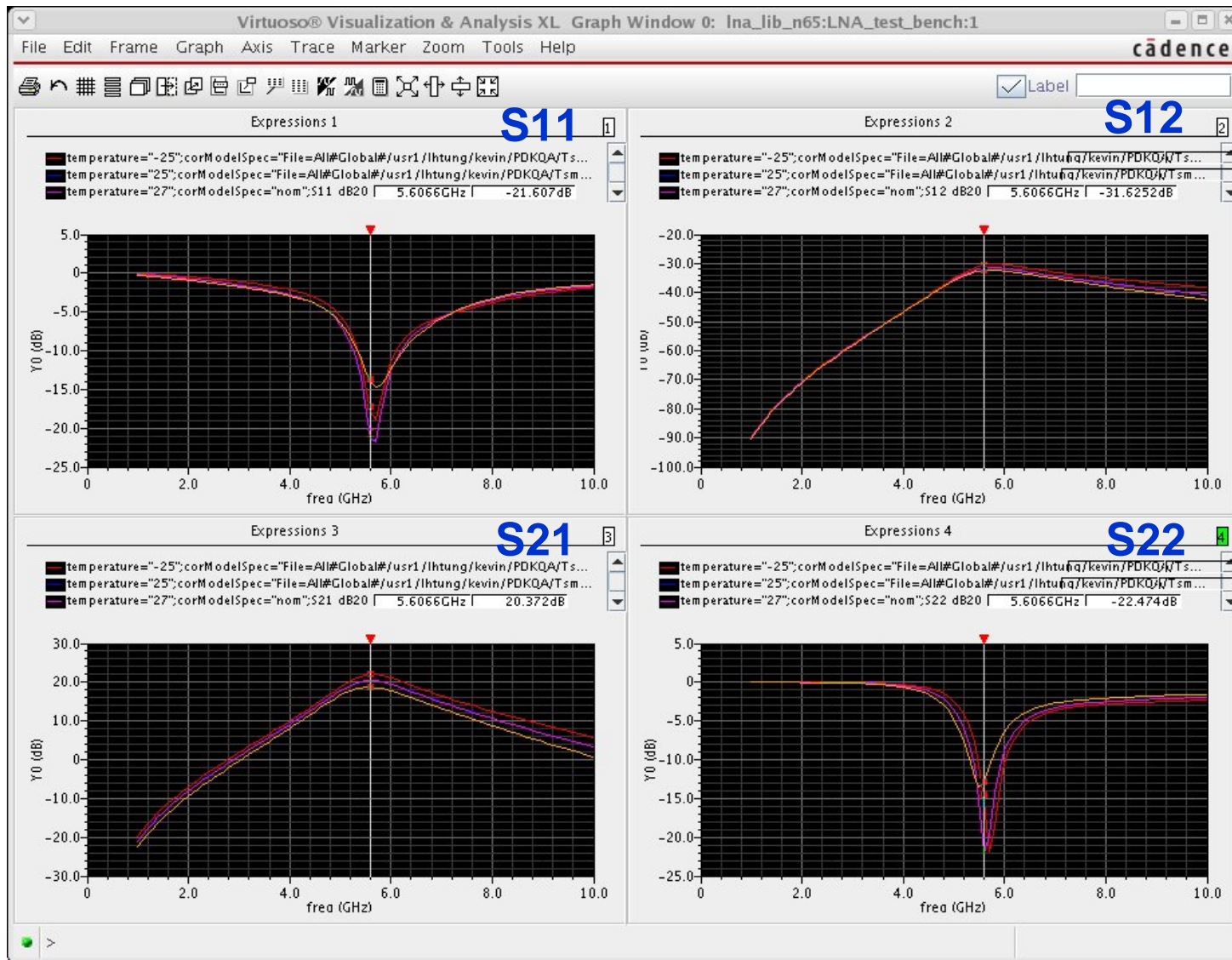
Corners	TT	FF	SS
<b>Temperature</b>	25	-25	75
<b>Design Variables</b>			
<b>Model Files</b>			
crn65lp_v1d0.scs	<input checked="" type="checkbox"/> probe	<input checked="" type="checkbox"/> probe	<input checked="" type="checkbox"/> probe
crn65lp_v1d0.scs	<input checked="" type="checkbox"/> tt_bip_npn	<input checked="" type="checkbox"/> ff_bip_npn	<input checked="" type="checkbox"/> ss_bip_npn
crn65lp_v1d0.scs	<input checked="" type="checkbox"/> tt_mlv	<input checked="" type="checkbox"/> ff_mlv	<input checked="" type="checkbox"/> ss_mlv
crn65lp_v1d0.scs	<input checked="" type="checkbox"/> tt_rfjvar	<input checked="" type="checkbox"/> ff_rfjvar	<input checked="" type="checkbox"/> ss_rfjvar
crn65lp_v1d0.scs	<input checked="" type="checkbox"/> tt_rfmos_mlv	<input checked="" type="checkbox"/> ff_rfmos_mlv	<input checked="" type="checkbox"/> ss_rfmos_mlv
crn65lp_v1d0.scs	<input checked="" type="checkbox"/> tt_rfmvar_hvt	<input checked="" type="checkbox"/> ff_rfmvar_hvt	<input checked="" type="checkbox"/> ss_rfmvar_hvt
crn65lp_v1d0.scs	<input checked="" type="checkbox"/> tt_rfmos	<input checked="" type="checkbox"/> ff_rfmos	<input checked="" type="checkbox"/> ss_rfmos
crn65lp_v1d0.scs	<input checked="" type="checkbox"/> tt_dio_na25	<input checked="" type="checkbox"/> ff_dio_na25	<input checked="" type="checkbox"/> ss_dio_na25
crn65lp_v1d0.scs	<input checked="" type="checkbox"/> tt_rfrtmom	<input checked="" type="checkbox"/> ff_rfrtmom	<input checked="" type="checkbox"/> ss_rfrtmom
crn65lp_v1d0.scs	<input checked="" type="checkbox"/> tt_rfmos_18	<input checked="" type="checkbox"/> ff_rfmos_18	<input checked="" type="checkbox"/> ss_rfmos_18
crn65lp_v1d0.scs	<input checked="" type="checkbox"/> tt_mos_cap_25	<input checked="" type="checkbox"/> ff_mos_cap_25	<input checked="" type="checkbox"/> ss_mos_cap_25
crn65lp_v1d0.scs	<input checked="" type="checkbox"/> tt_dio_33	<input checked="" type="checkbox"/> ff_dio_33	<input checked="" type="checkbox"/> ss_dio_33
crn65lp_v1d0.scs	<input checked="" type="checkbox"/> tt_rfind	<input checked="" type="checkbox"/> ff_rfind	<input checked="" type="checkbox"/> ss_rfind
crn65lp_v1d0.scs	<input checked="" type="checkbox"/> tt_25	<input checked="" type="checkbox"/> ff_25	<input checked="" type="checkbox"/> ss_25
crn65lp_v1d0.scs	<input checked="" type="checkbox"/> tt_dio_esd	<input checked="" type="checkbox"/> ff_dio_esd	<input checked="" type="checkbox"/> ss_dio_esd
crn65lp_v1d0.scs	<input checked="" type="checkbox"/> tt_33	<input checked="" type="checkbox"/> ff_33	<input checked="" type="checkbox"/> ss_33
crn65lp_v1d0.scs	<input checked="" type="checkbox"/> tt_bip	<input checked="" type="checkbox"/> ff_bip	<input checked="" type="checkbox"/> ss_bip
crn65lp_v1d0.scs	<input checked="" type="checkbox"/> tt_dio_mlv	<input checked="" type="checkbox"/> ff_dio_mlv	<input checked="" type="checkbox"/> ss_dio_mlv
crn65lp_v1d0.scs	<input checked="" type="checkbox"/> tt_dio_25	<input checked="" type="checkbox"/> ff_dio_25	<input checked="" type="checkbox"/> ss_dio_25
crn65lp_v1d0.scs	<input checked="" type="checkbox"/> tt_rfmos_lvt	<input checked="" type="checkbox"/> ff_rfmos_lvt	<input checked="" type="checkbox"/> ss_rfmos_lvt
crn65lp_v1d0.scs	<input checked="" type="checkbox"/> tt_dio_18	<input checked="" type="checkbox"/> ff_dio_18	<input checked="" type="checkbox"/> ss_dio_18
crn65lp_v1d0.scs	<input checked="" type="checkbox"/> tt_dio_hvt	<input checked="" type="checkbox"/> ff_dio_hvt	<input checked="" type="checkbox"/> ss_dio_hvt
crn65lp_v1d0.scs	<input checked="" type="checkbox"/> tt_rtmom	<input checked="" type="checkbox"/> ff_rtmom	<input checked="" type="checkbox"/> ss_rtmom
crn65lp_v1d0.scs	<input checked="" type="checkbox"/> tt_rfmos_33	<input checked="" type="checkbox"/> ff_rfmos_33	<input checked="" type="checkbox"/> ss_rfmos_33
crn65lp_v1d0.scs	<input checked="" type="checkbox"/> tt_rfmvar_25	<input checked="" type="checkbox"/> ff_rfmvar_25	<input checked="" type="checkbox"/> ss_rfmvar_25
crn65lp_v1d0.scs	<input checked="" type="checkbox"/> tt_disres	<input checked="" type="checkbox"/> ff_disres	<input checked="" type="checkbox"/> ss_disres
crn65lp_v1d0.scs	<input checked="" type="checkbox"/> tt_dio_na	<input checked="" type="checkbox"/> ff_dio_na	<input checked="" type="checkbox"/> ss_dio_na

Load Save Import PCF/DCF

OK Cancel Apply Help



# Corner simulation result



# Contents

- ☐ Chapter 1 Introduction
- ☐ Chapter 2 Schematic Capture
- ☐ Chapter 3 Pre-layout Simulation
- ☒ **Chapter 4 Layout Creation**
- ☐ Chapter 5 Physical Verification
- ☐ Chapter 6 Post-layout Simulation



# Layout Creation

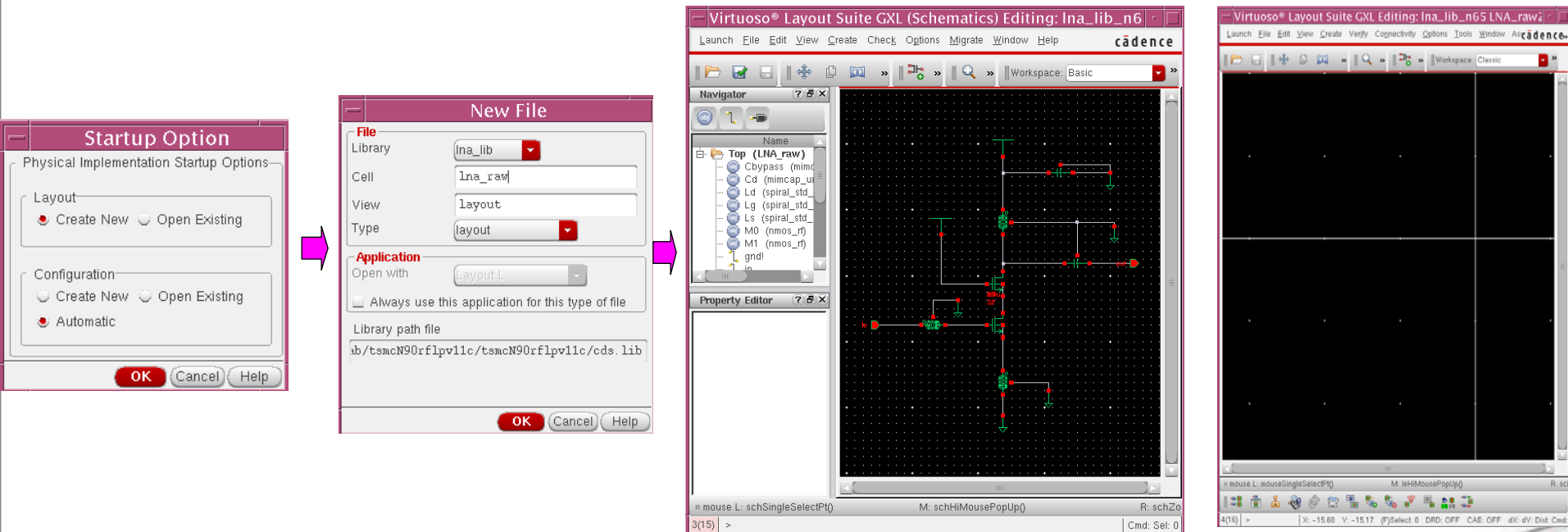
After completed the pre-layout simulation and make sure the functionality and the circuit performance are all correct and in the design specifications, we can now start to create the corresponding layout for our design. The layout creation procedures are partitioned into three parts: “Schematic-Driven-Layout”,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

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

# Schematic-driven-layout

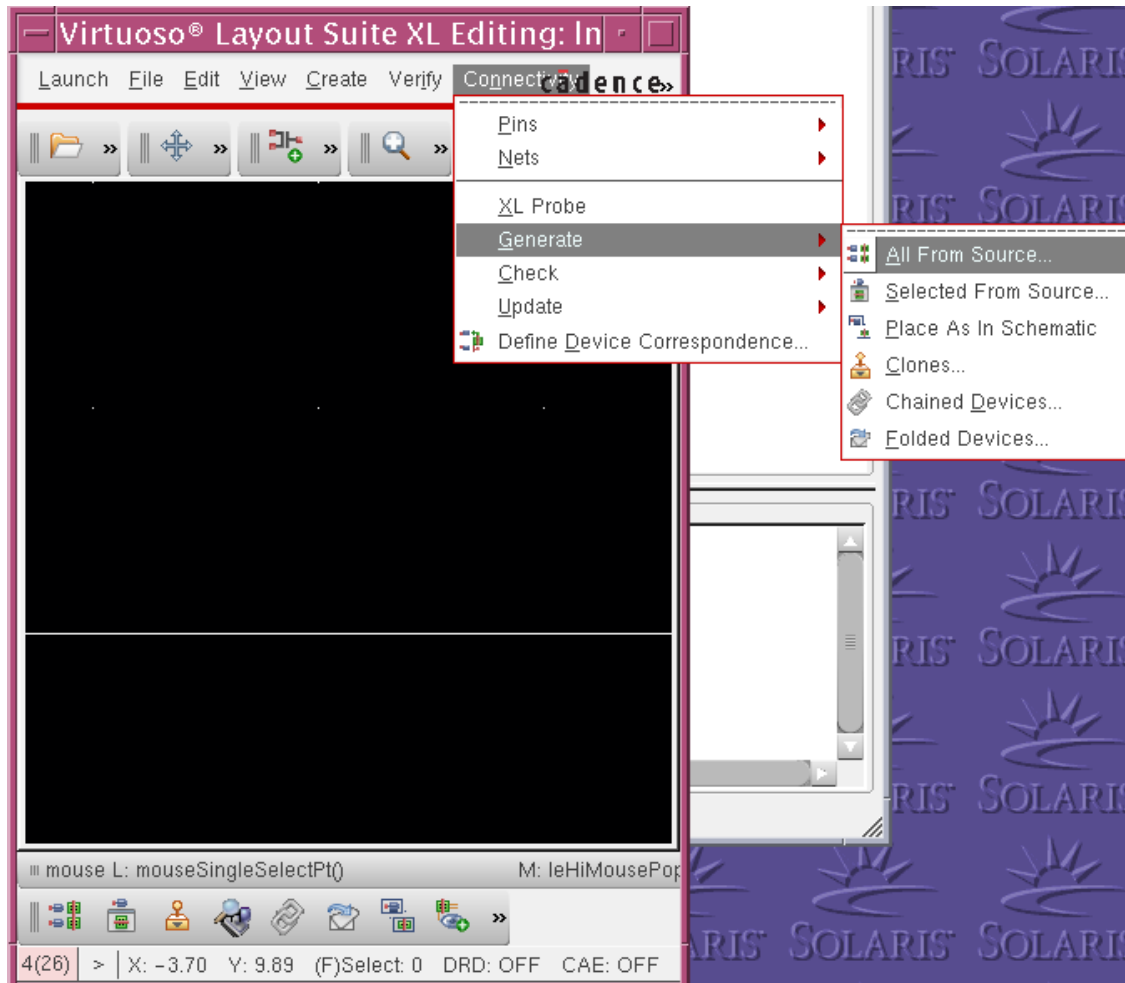
Steps for Schematic-Driven-Layout method:

1. Open the schematic view of our design (Ina\_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.



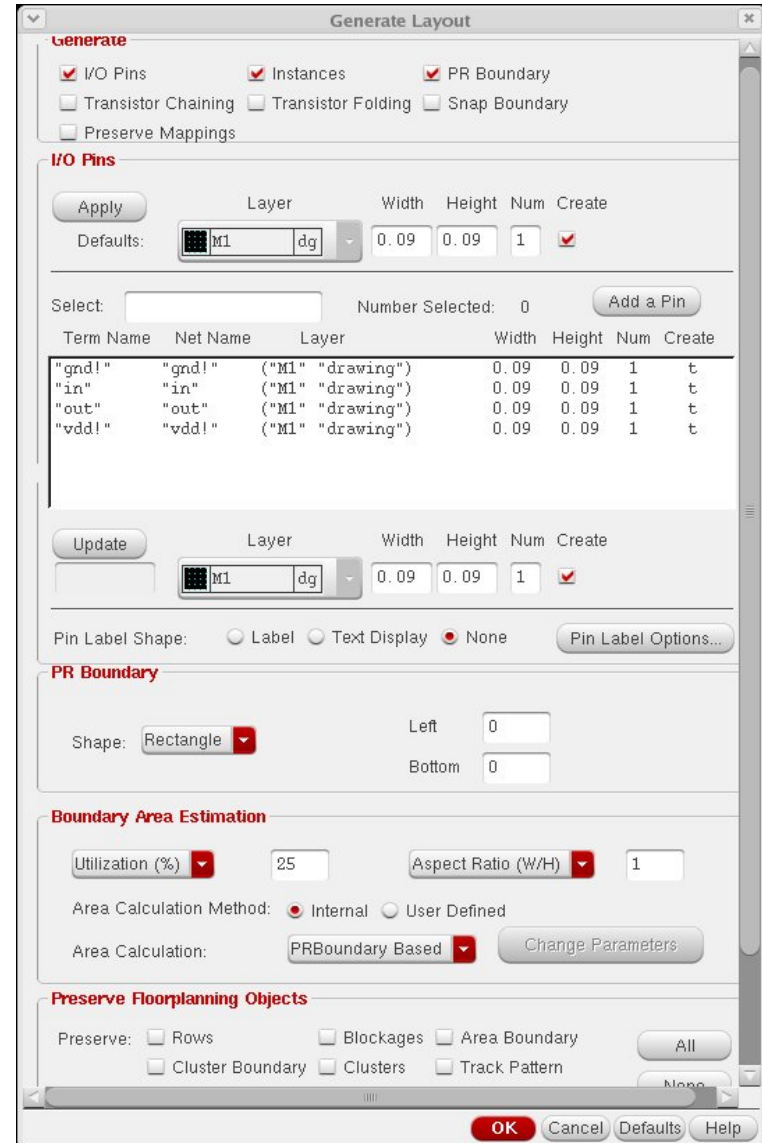
# Schematic-driven-layout

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.



**Generate Layout**

☒ I/O Pins    ☒ Instances    ☒ PR Boundary  
☐ Transistor Chaining    ☐ Transistor Folding    ☐ Snap Boundary  
☐ Preserve Mappings

**I/O Pins**

Apply    Layer: M1 dg    Width: 0.09    Height: 0.09    Num: 1    Create: ☒

Defaults: ☐ M1 dg    0.09    0.09    1    ☒

Select:    Number Selected: 0    Add a Pin

Term Name	Net Name	Layer	Width	Height	Num	Create
"gnd!"	"gnd!"	("M1" "drawing")	0.09	0.09	1	t
"in"	"in"	("M1" "drawing")	0.09	0.09	1	t
"out"	"out"	("M1" "drawing")	0.09	0.09	1	t
"vdd!"	"vdd!"	("M1" "drawing")	0.09	0.09	1	t

Update    Layer: M1 dg    Width: 0.09    Height: 0.09    Num: 1    Create: ☒

Pin Label Shape: ☐ Label ☐ Text Display ☒ None    Pin Label Options...

**PR Boundary**

Shape: Rectangle    Left: 0    Bottom: 0

**Boundary Area Estimation**

Utilization (%): 25    Aspect Ratio (W/H): 1

Area Calculation Method: ☒ Internal ☐ User Defined

Area Calculation: PRBoundary Based    Change Parameters

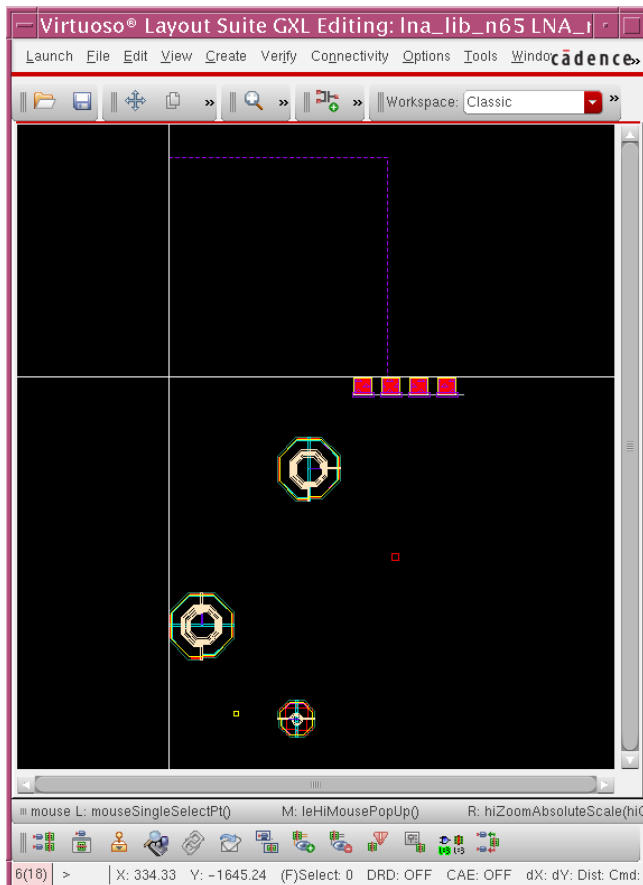
**Preserve Floorplanning Objects**

Preserve: ☐ Rows    ☐ Blockages    ☐ Area Boundary    All  
☐ Cluster Boundary    ☐ Clusters    ☐ Track Pattern    None

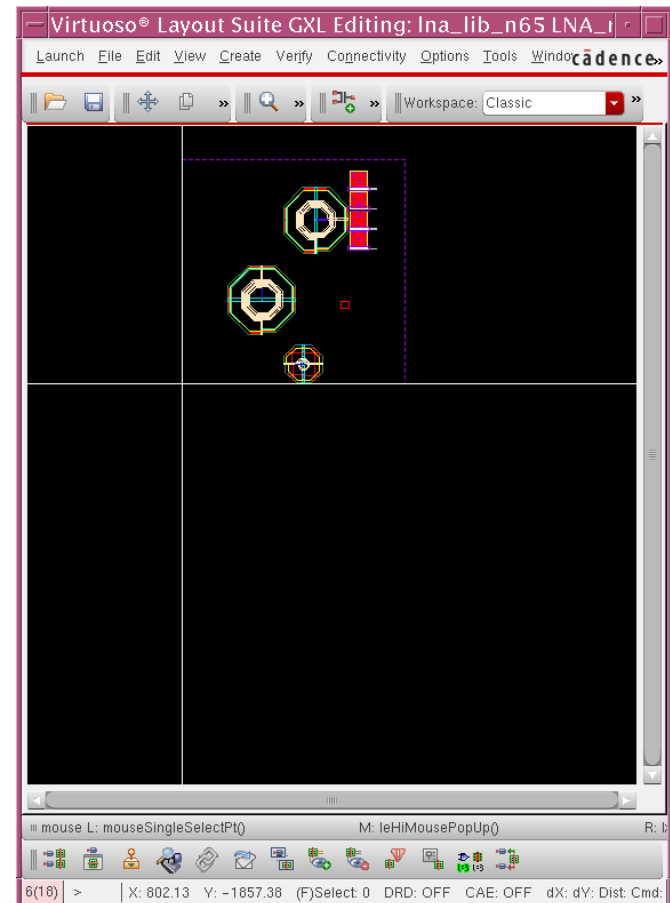
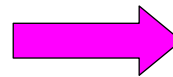
OK    Cancel    Defaults    Help

# 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.

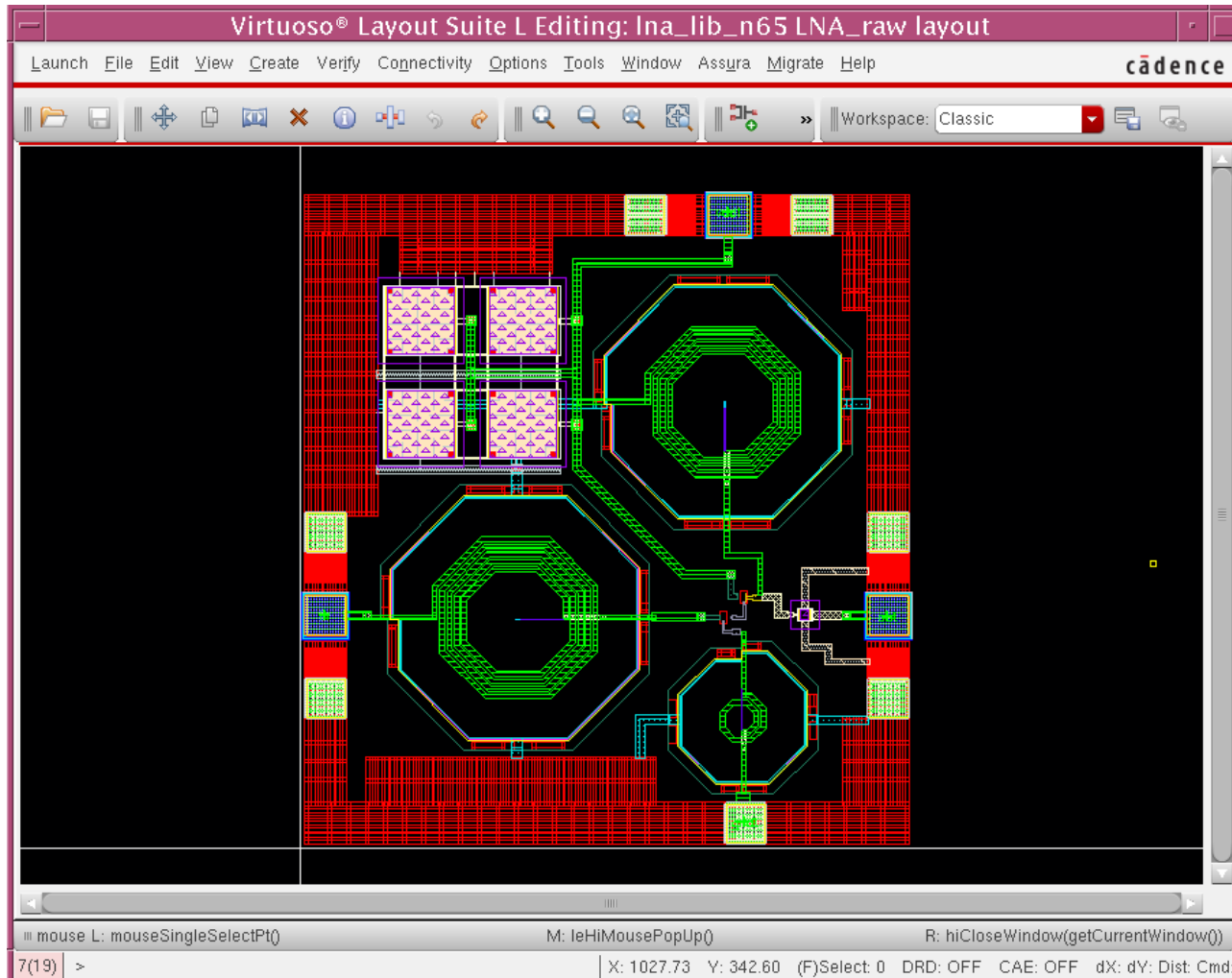


Place As In  
Schematic



# 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

- ❑ Chapter 1 Introduction
- ❑ Chapter 2 Schematic Capture
- ❑ Chapter 3 Pre-layout Simulation
- ❑ Chapter 4 Layout Creation
- ❑ Chapter 5 Physical Verification**
- ❑ Chapter 6 Post-layout Simulation

# 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

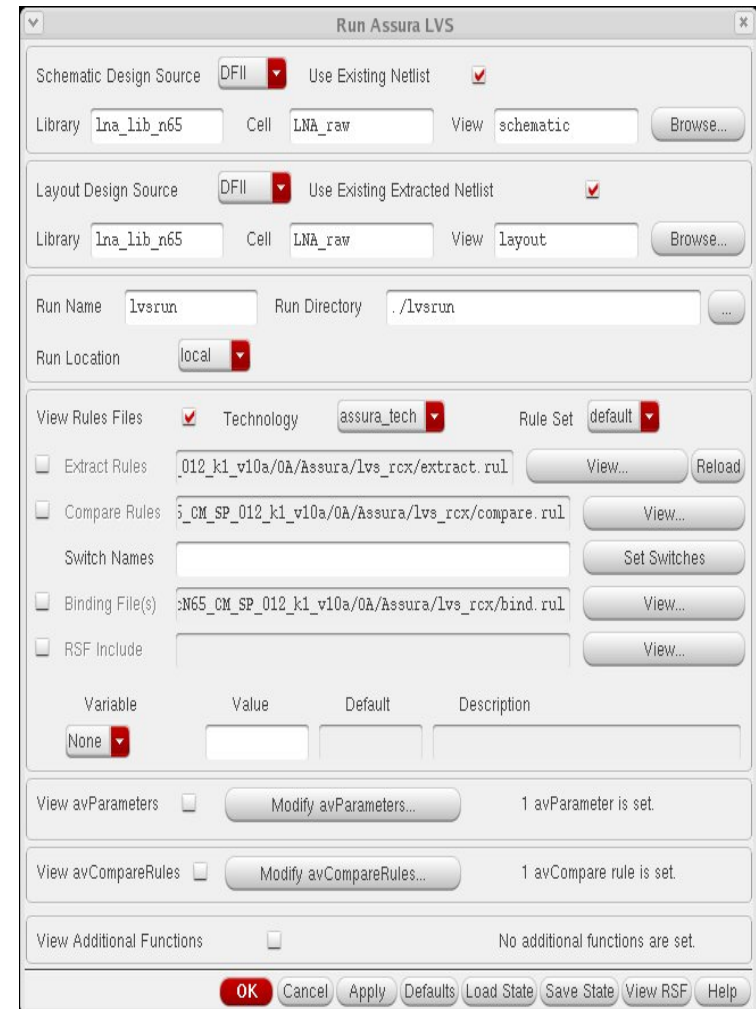
## Assura DRC Flow

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.

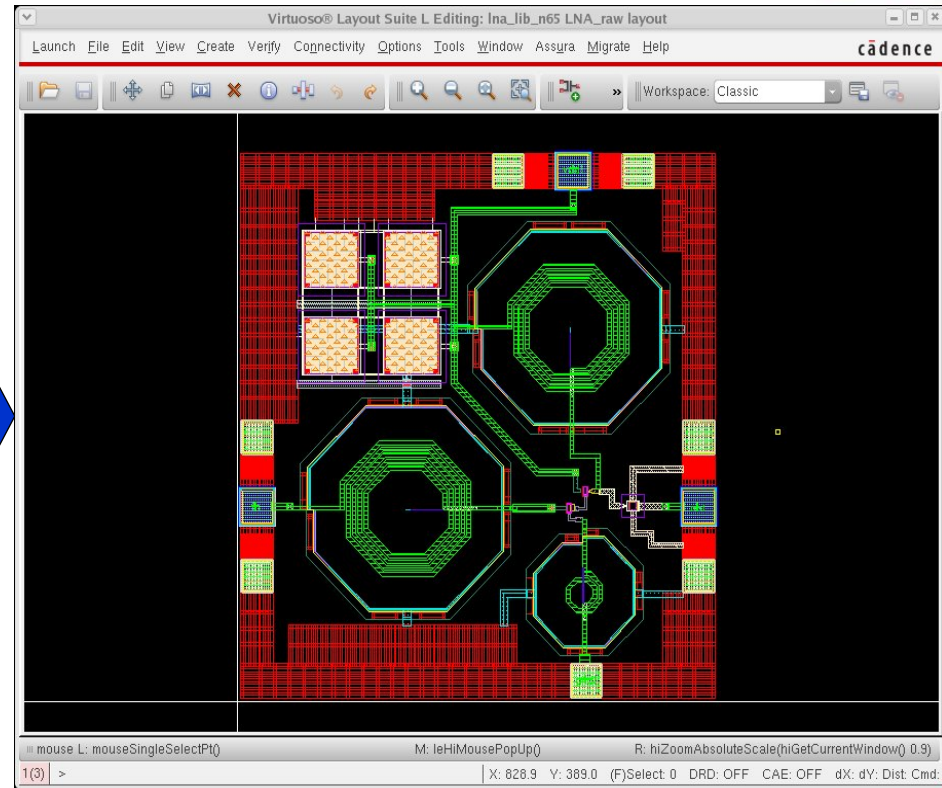
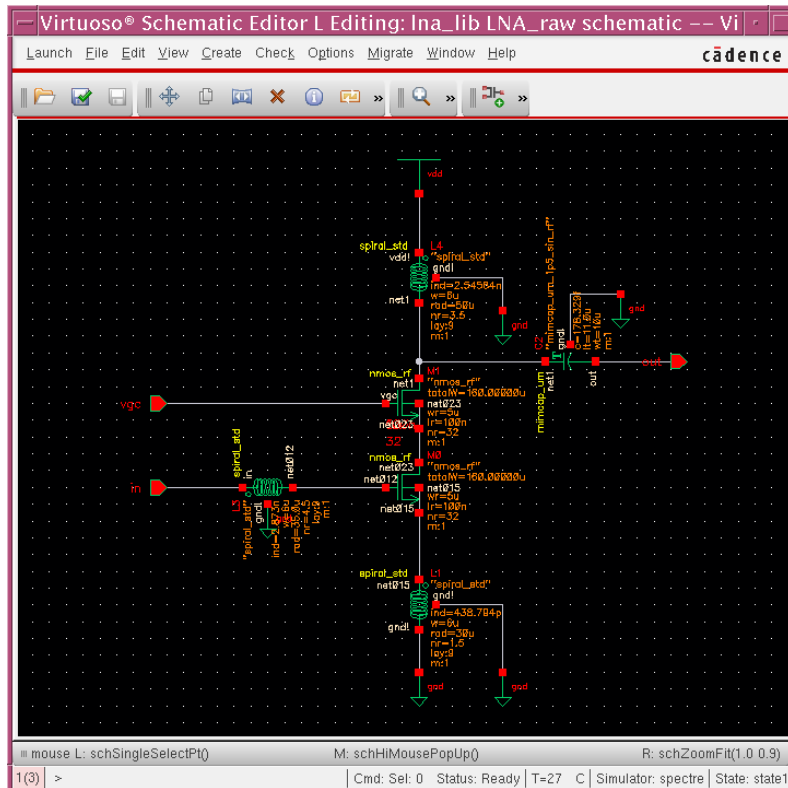
1. Click “Assura-> Technology” to load the file, “assura\_tech.lib” in the PDK install directory.
2. Click “Assura-> Run LVS..” in layout window to invoke Assura LVS graphic user interface.
3. Fill in the “Assura run directory” and select the “Technology” field to “assura” in Assura LVS window.
4. Click “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.



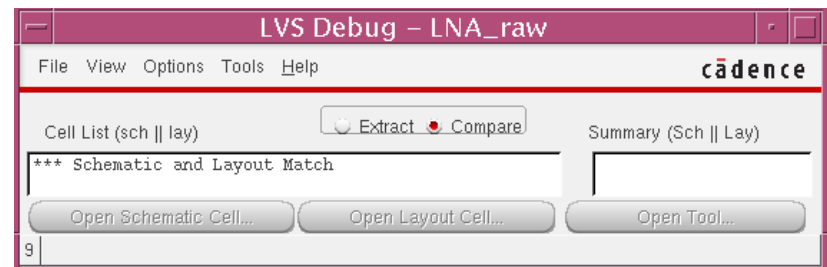
The image shows the 'Run Assura LVS' dialog box with the following settings:

- Schematic Design Source:** DFile, Use Existing Netlist (checked). Library: lna\_lib\_n65, Cell: LNA\_raw, View: schematic.
- Layout Design Source:** DFile, Use Existing Extracted Netlist (checked). Library: lna\_lib\_n65, Cell: LNA\_raw, View: layout.
- Run Name:** lvsrun, **Run Directory:** ./lvsrun.
- Run Location:** local.
- View Rules Files:** checked. Technology: assura\_tech, Rule Set: default.
- Extract Rules:** unchecked. Path: \_012\_k1\_v10a/0A/Assura/lvs\_rcx/extract.rul. Buttons: View..., Reload.
- Compare Rules:** unchecked. Path: \_012\_k1\_v10a/0A/Assura/lvs\_rcx/compare.rul. Button: View...
- Switch Names:** empty text field. Button: Set Switches.
- Binding File(s):** unchecked. Path: \_012\_k1\_v10a/0A/Assura/lvs\_rcx/bind.rul. Button: View...
- RSF Include:** unchecked. Button: View...
- Variable:** dropdown menu showing 'None'.
- Value:** empty text field.
- Default:** empty text field.
- Description:** empty text field.
- View avParameters:** unchecked. Button: Modify avParameters... 1 avParameter is set.
- View avCompareRules:** unchecked. Button: Modify avCompareRules... 1 avCompare rule is set.
- View Additional Functions:** unchecked. No additional functions are set.
- Buttons:** OK, Cancel, Apply, Defaults, Load State, Save State, View RSF, Help.

# Assura LVS Flow (con't)



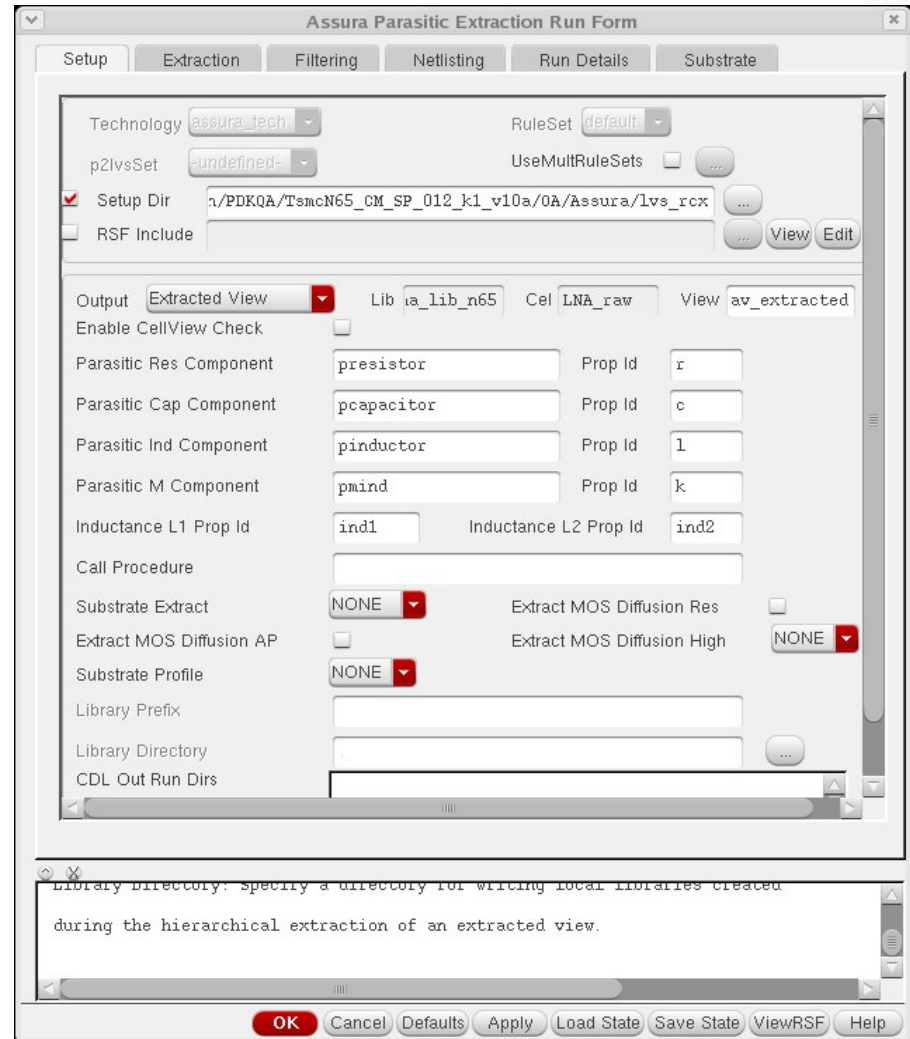
At this case, after running LVS, the result shows **“Schematic and Layout Match”**



# 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.

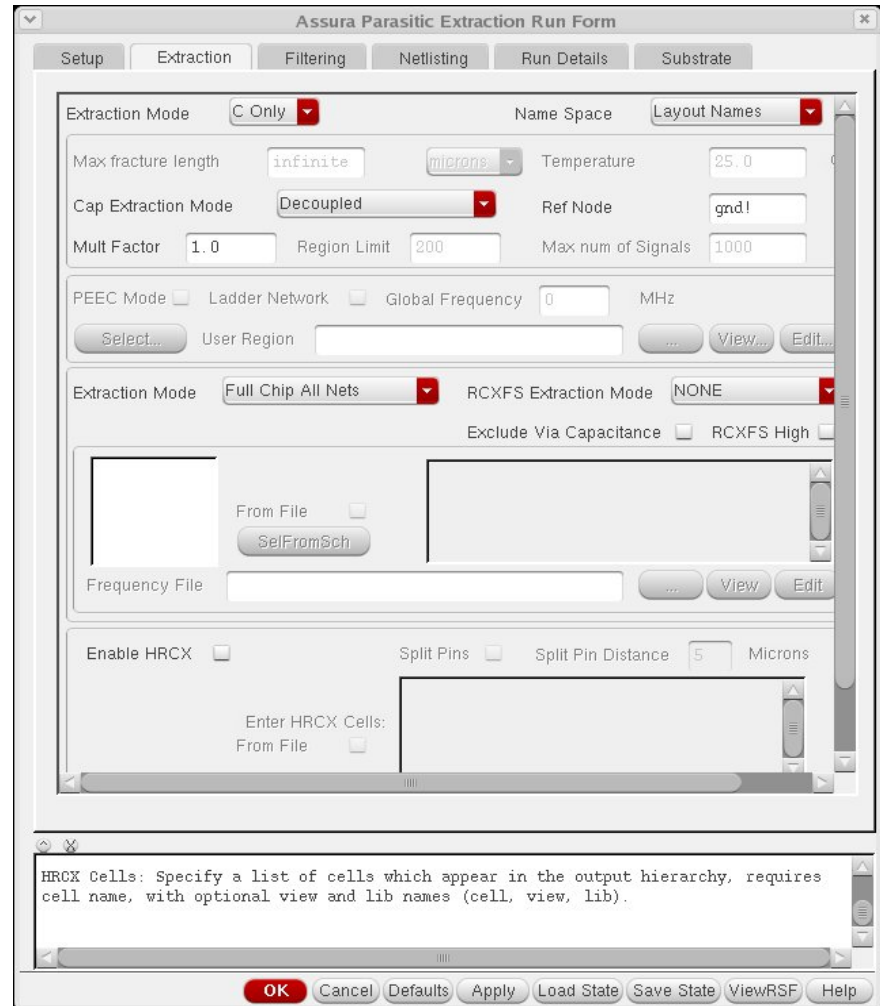
1. Click “Assura-> Run RCX...” in layout window to invoke Assura RCX graphic user interface.
2. Select “Output” to “Extracted View” in “setup” folder of Assura RCX window to output extract result to “av\_extracted” view.





# 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.



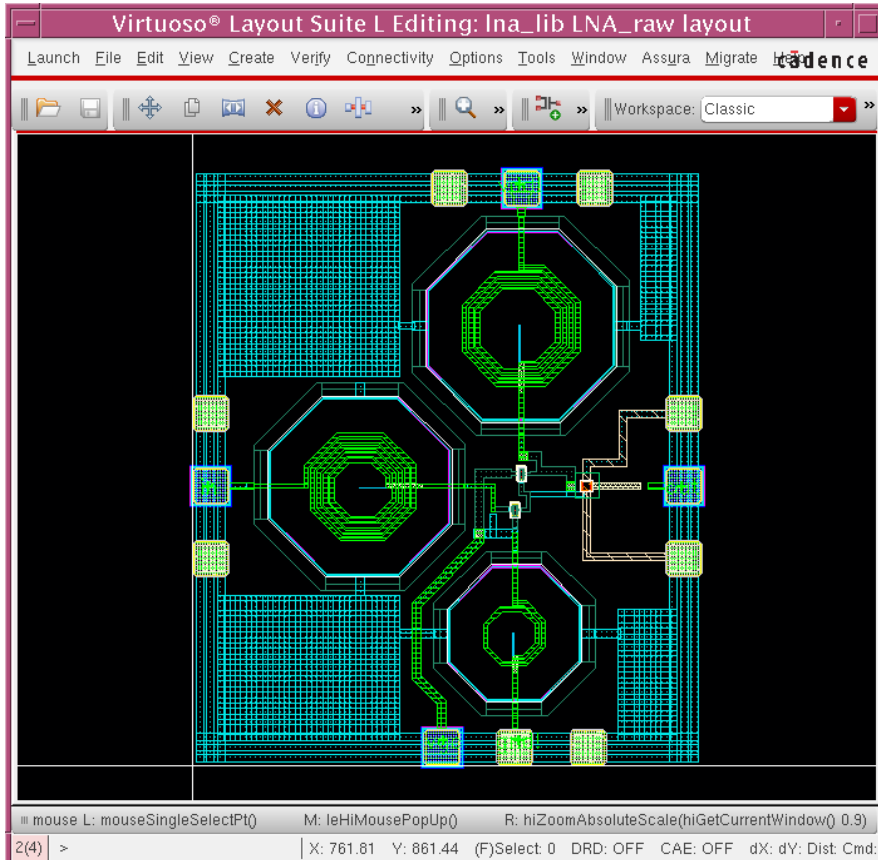
The image shows the 'Assura Parasitic Extraction Run Form' dialog box. It has several tabs: Setup, Extraction (selected), Filtering, Netlisting, Run Details, and Substrate. The 'Extraction' tab contains the following settings:

- Extraction Mode:** C Only (dropdown)
- Name Space:** Layout Names (dropdown)
- Max fracture length:** infinite (text field)
- Cap Extraction Mode:** Decoupled (dropdown)
- Ref Node:** gnd! (text field)
- Mult Factor:** 1.0 (text field)
- Region Limit:** 200 (text field)
- Max num of Signals:** 1000 (text field)
- Temperature:** 25.0 (text field)
- PEEC Mode:** ☐ Ladder Network ☐ Global Frequency 0 MHz
- Extraction Mode:** Full Chip All Nets (dropdown)
- RCXFS Extraction Mode:** NONE (dropdown)
- Exclude Via Capacitance:** ☐ RCXFS High ☐
- From File:** ☐ **SelfFromSch:** ☐
- Frequency File:** (text field)
- Enable HRCX:** ☐ **Split Pins:** ☐ **Split Pin Distance:** 5 Microns
- Enter HRCX Cells:** (text field)
- From File:** ☐

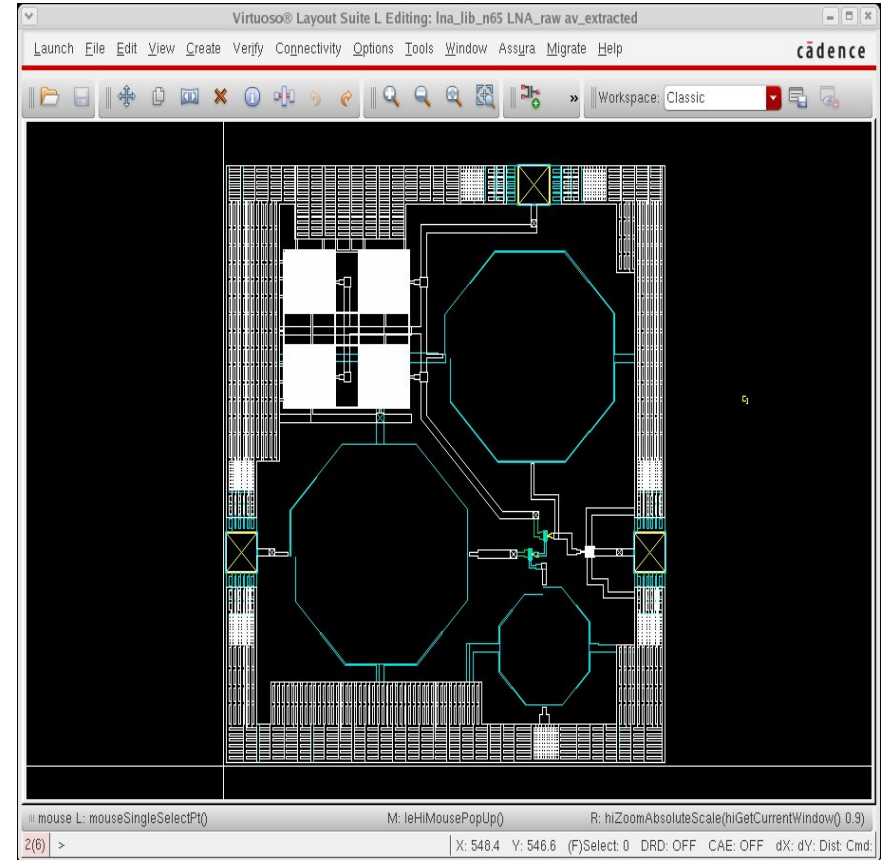
At the bottom, there is a text area for 'HRCX Cells' with the instruction: 'Specify a list of cells which appear in the output hierarchy, requires cell name, with optional view and lib names (cell, view, lib)'. Below the text area are buttons: OK, Cancel, Defaults, Apply, Load State, Save State, ViewRSF, and Help.

# 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

- ❑ Chapter 1 Introduction
- ❑ Chapter 2 Schematic Capture
- ❑ Chapter 3 Pre-layout Simulation
- ❑ Chapter 4 Layout Creation
- ❑ Chapter 5 Physical Verification
- ❑ Chapter 6 Post-layout Simulation**

# 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

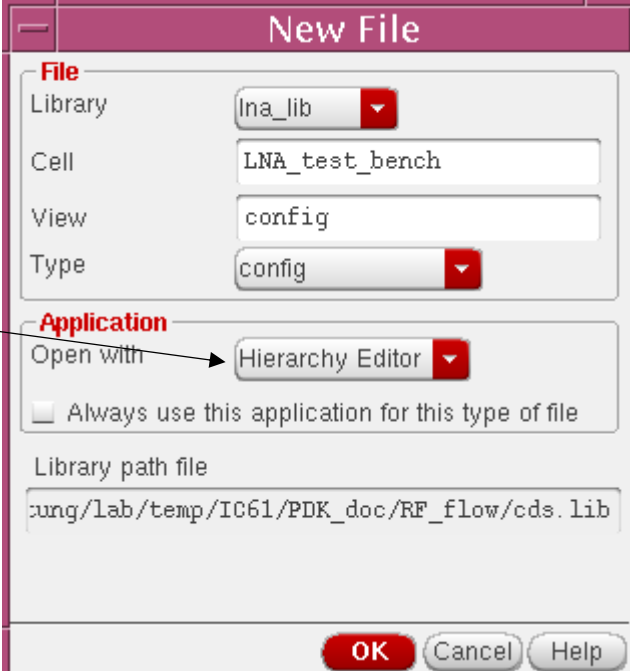
# Post-layout Simulation

## •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”



**New File**

**File**

Library: lna\_lib

Cell: LNA\_test\_bench

View: config

Type: config

**Application**

Open with: Hierarchy Editor

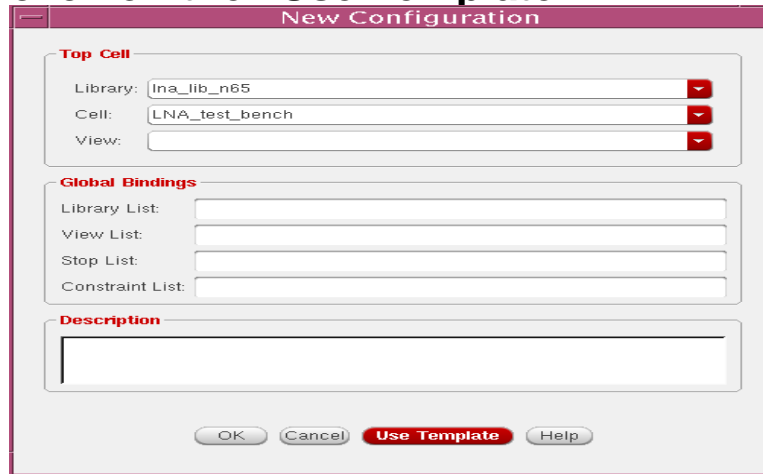
☐ Always use this application for this type of file

Library path file: c:/ung/lab/temp/IC61/PDK\_doc/RF\_flow/cds.lib

OK Cancel Help

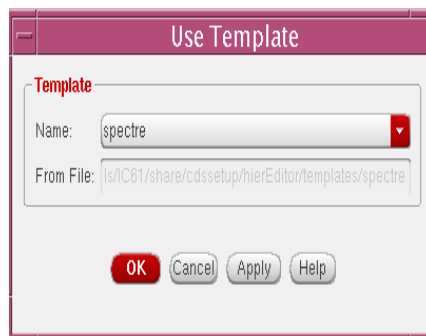
# Post-layout Simulation

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

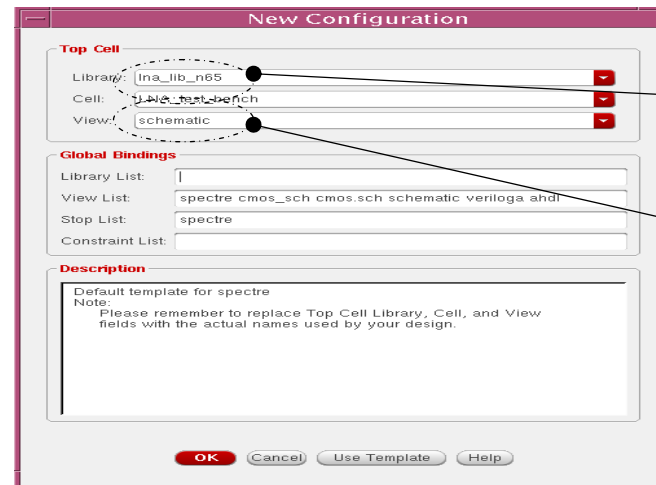


The 'New Configuration' dialog box is shown. It has three main sections: 'Top Cell', 'Global Bindings', and 'Description'. In the 'Top Cell' section, 'Library' is set to 'Ina\_lib\_n65', 'Cell' is 'LNA\_test\_bench', and 'View' is empty. The 'Global Bindings' section has empty fields for 'Library List', 'View List', 'Stop List', and 'Constraint List'. The 'Description' section has an empty text area. At the bottom are buttons for 'OK', 'Cancel', 'Use Template' (highlighted in red), and 'Help'.

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



The 'Use Template' dialog box is shown. It has a 'Template' section with 'Name' set to 'spectre' and 'From File' set to 'Is/IC61/share/cdssetup/hierEditor/templates/spectre'. At the bottom are buttons for 'OK' (highlighted in red), 'Cancel', 'Apply', and 'Help'.



The 'New Configuration' dialog box is shown after using the template. The 'Top Cell' section now has 'View' set to 'schematic'. The 'Global Bindings' section has 'View List' set to 'spectre cmos\_sch cmos.sch schematic veriloga andi' and 'Stop List' set to 'spectre'. The 'Description' section contains text: 'Default template for spectre' and 'Note: Please remember to replace Top Cell Library, Cell, and View fields with the actual names used by your design.' At the bottom are buttons for 'OK' (highlighted in red), 'Cancel', 'Use Template', and 'Help'.

Ina\_lib\_n65

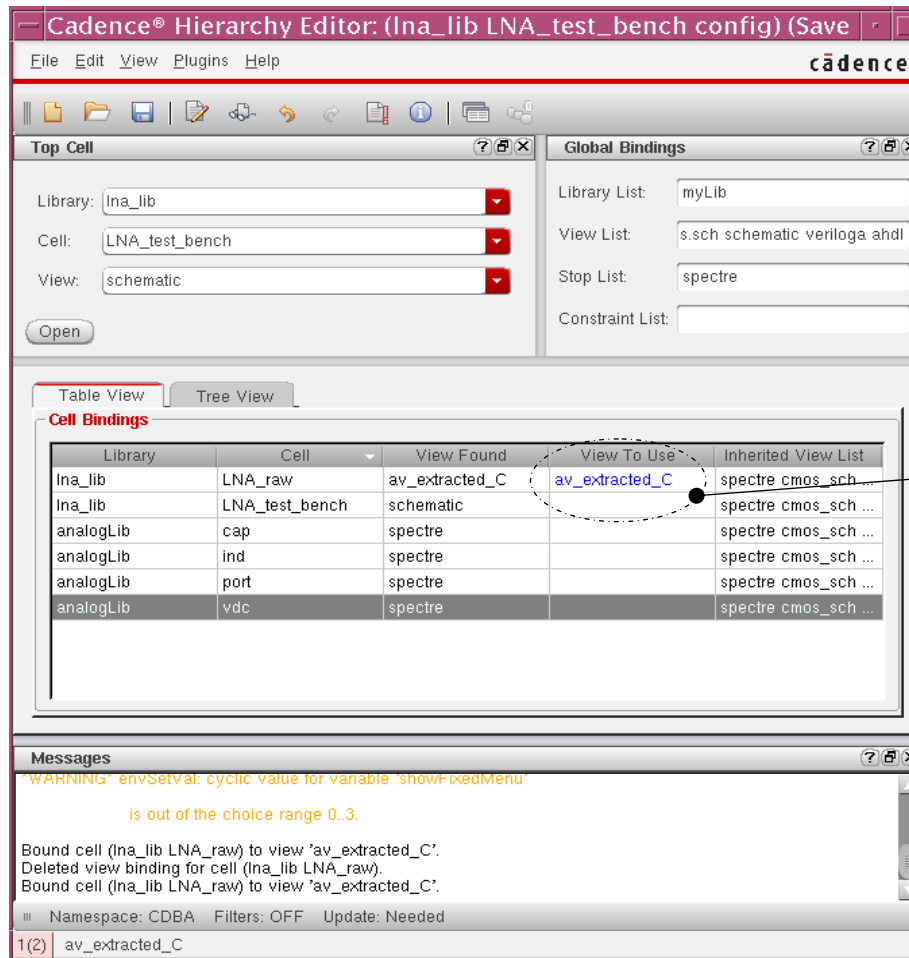
schematic



# 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.

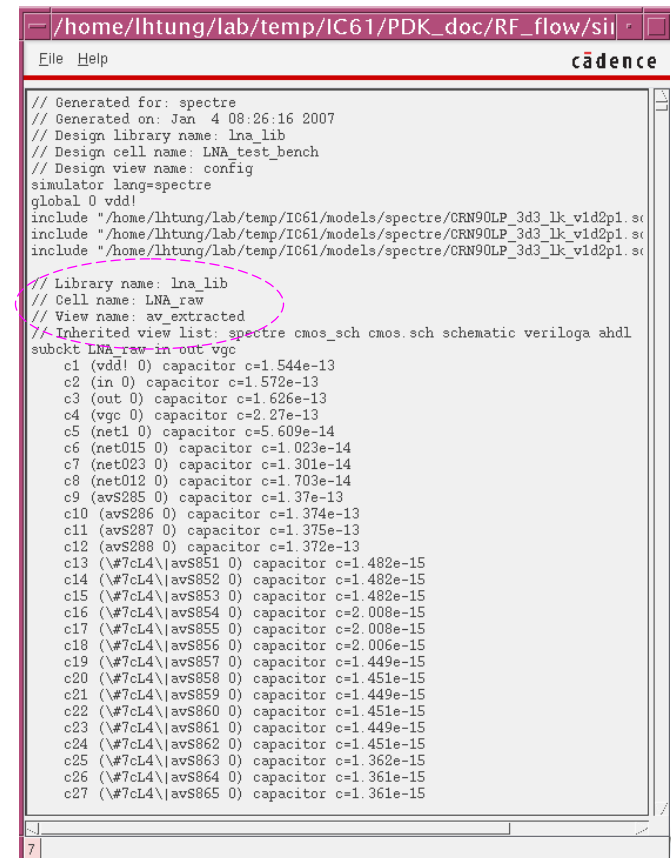
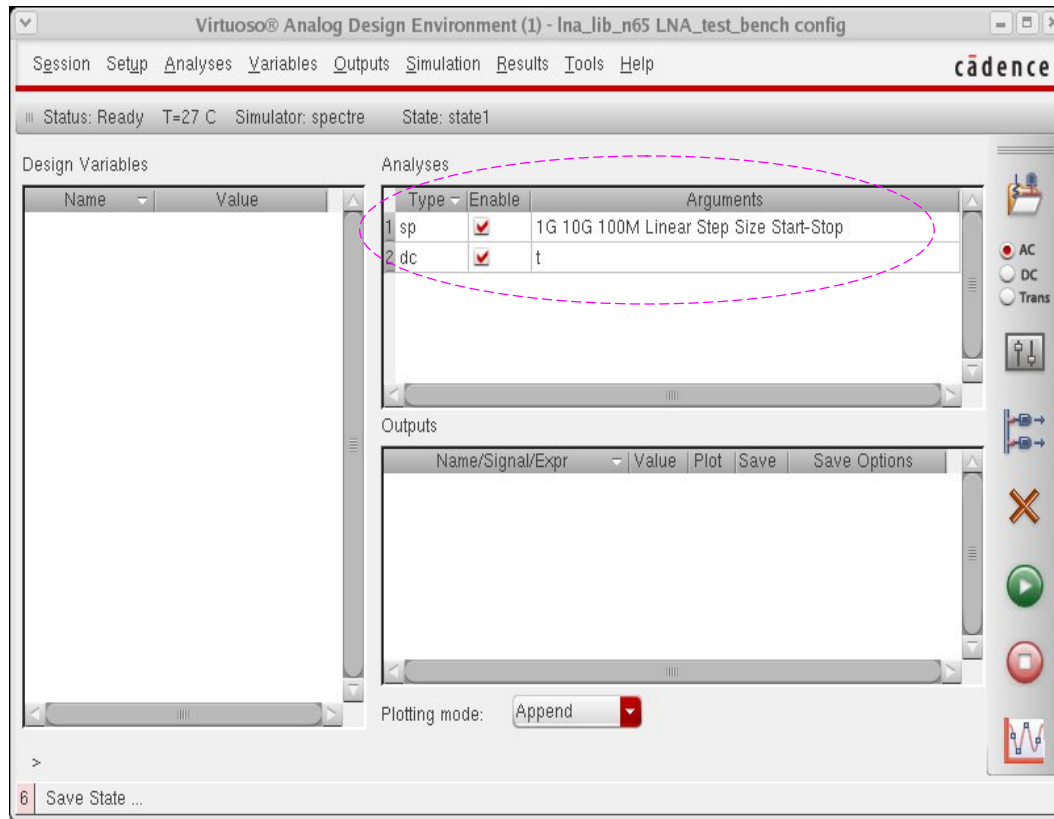


You can change the view that you have created by extraction tool.

# 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.



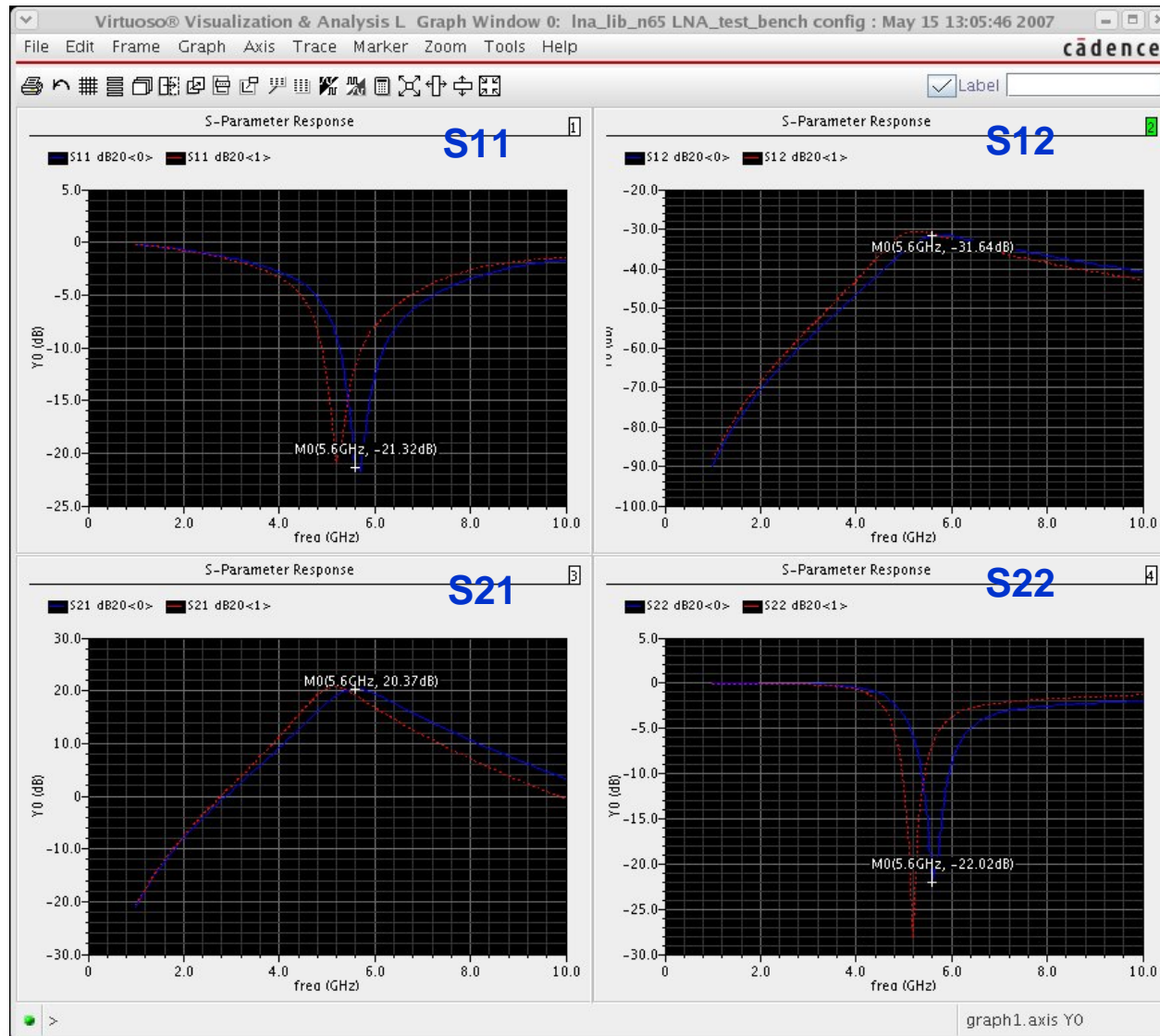
# Post-layout Simulation Result

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

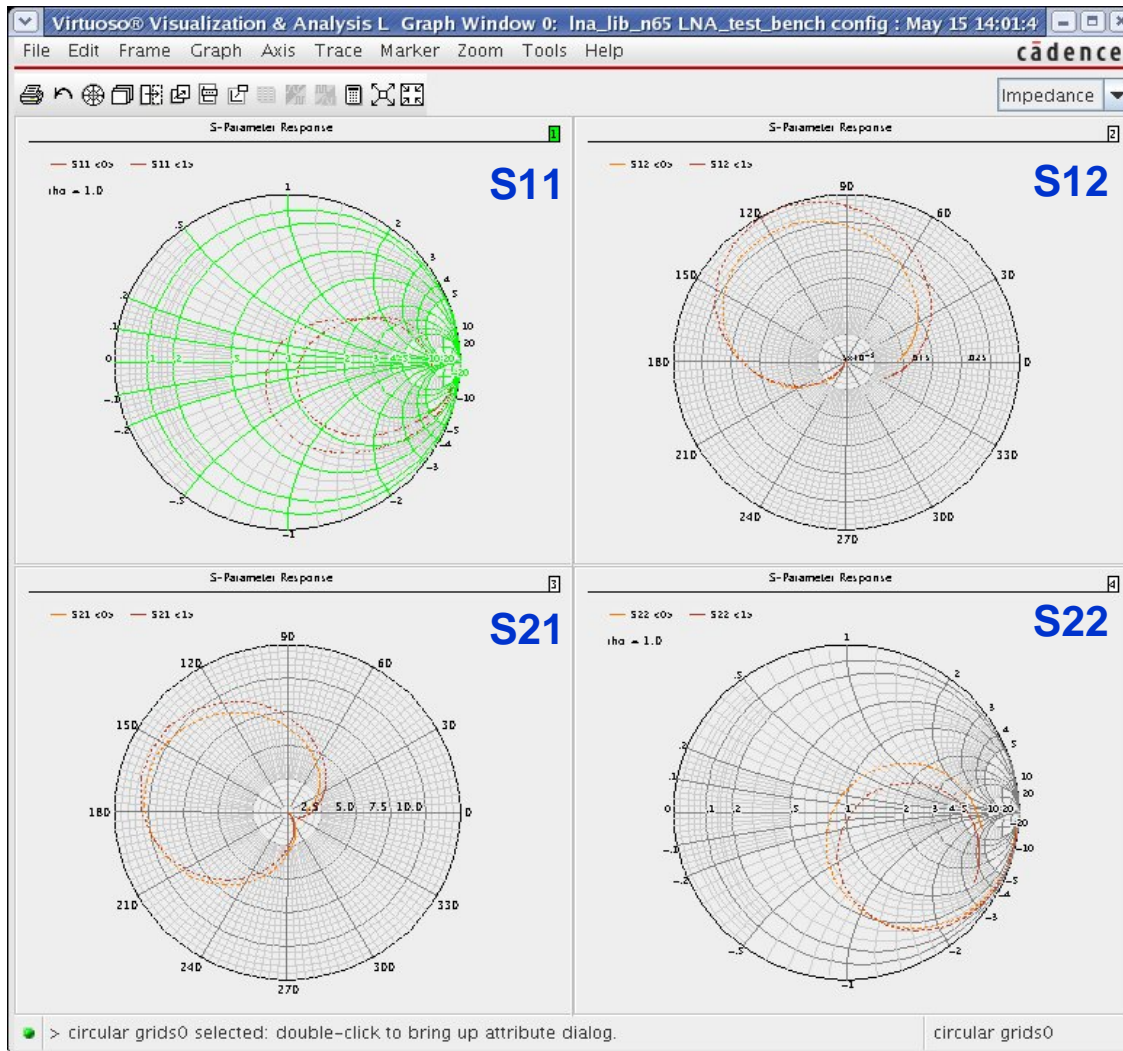
## S-parameter (20dB)



Solid line: pre-layout  
Dash line: C-only mode

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

## S-parameter

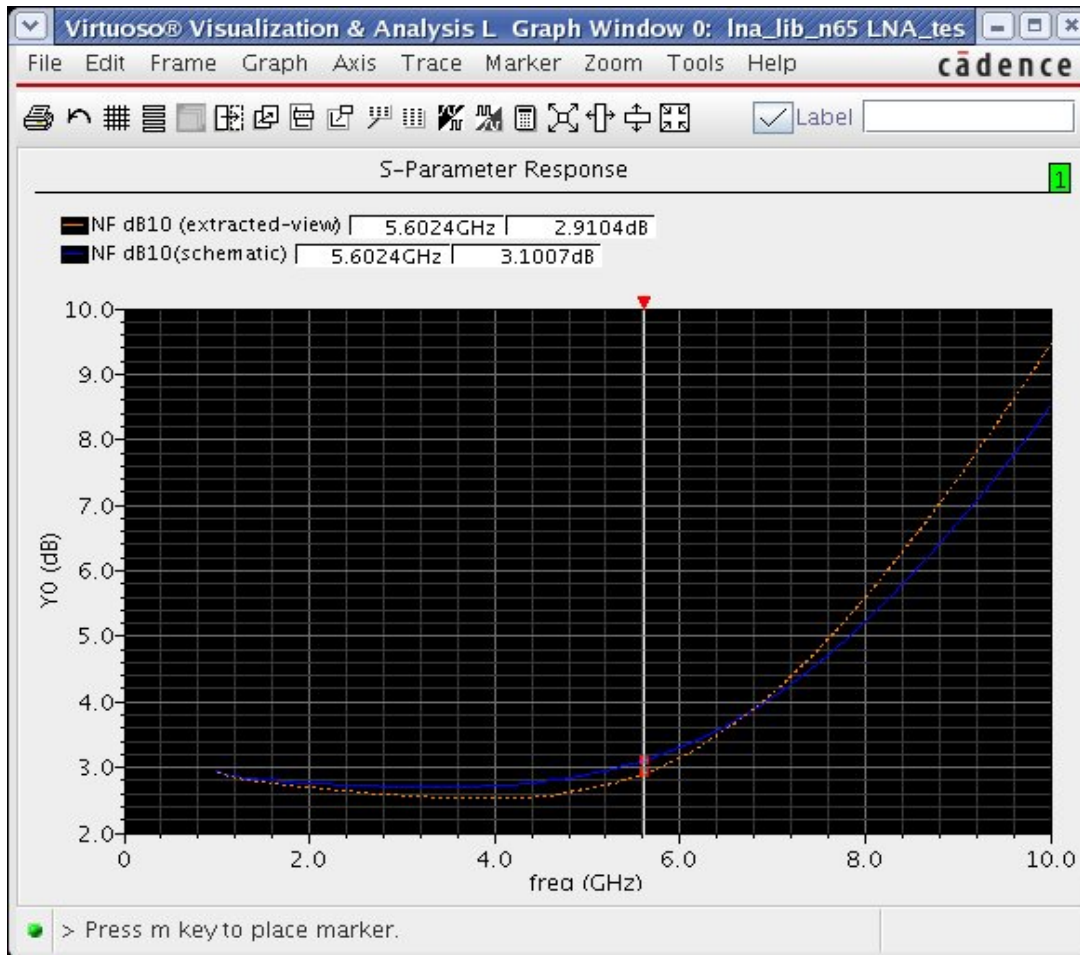


Red line: pre-layout  
 Brown line: C-only mode



# ■ Assura RC mode v.s. Pre-layout simulation

## Noise Figure



Solid line: pre-layout  
 Dash line: RC mode



**This slide is intentionally left blank**