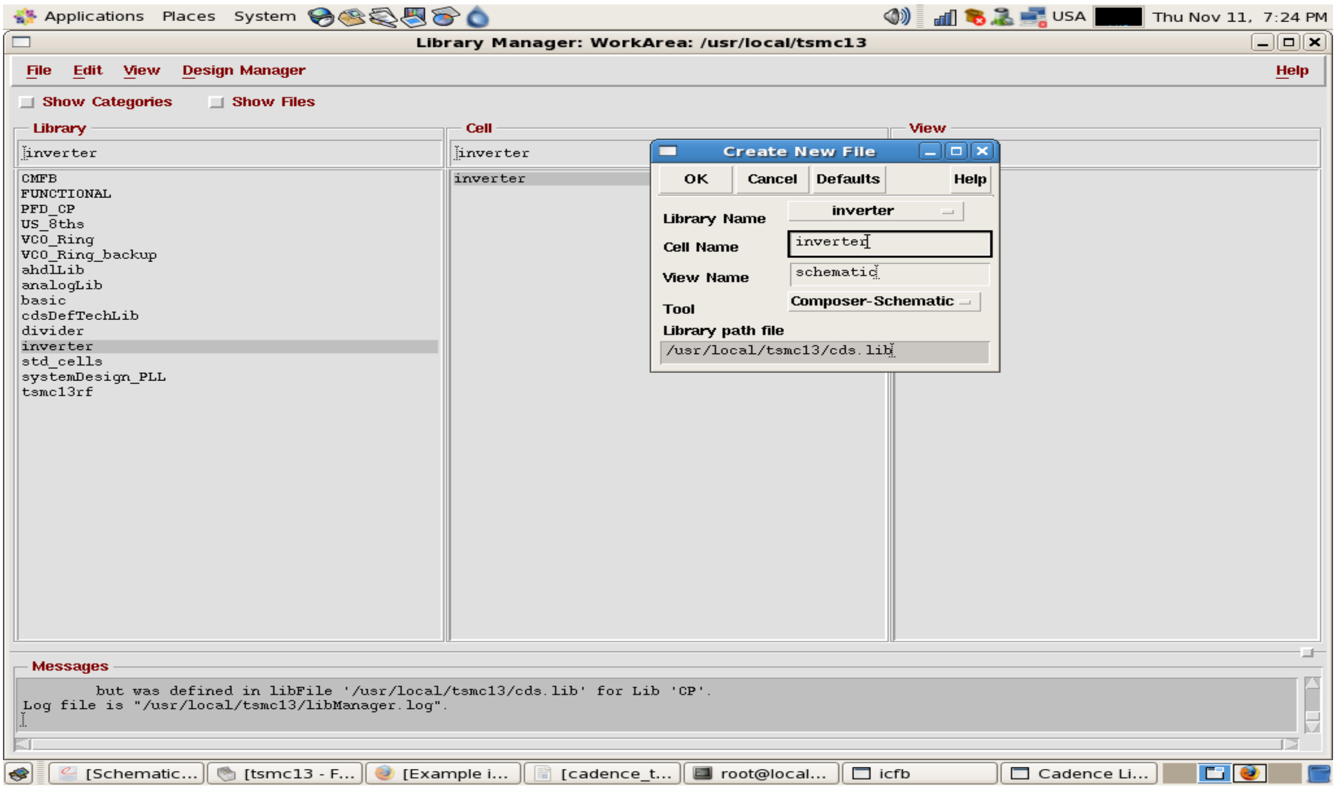


Library Manager: WorkArea: /usr/local/tsmc13

Library: inverter

```
CMFB
FUNCTIONAL
PFD_CP
US_8ths
VCO_Ring
VCO_Ring_backup
ahdlLib
analogLib
basic
cdsDefTechLib
divider
inverter
std_cells
systemDesign_PLL
tsmc13rf
```

Cell: inverter

Create New File

Library Name: inverter
Cell Name: inverter
View Name: schematic
Tool: Composer-Schematic
Library path file: /usr/local/tsmc13/cds.lib

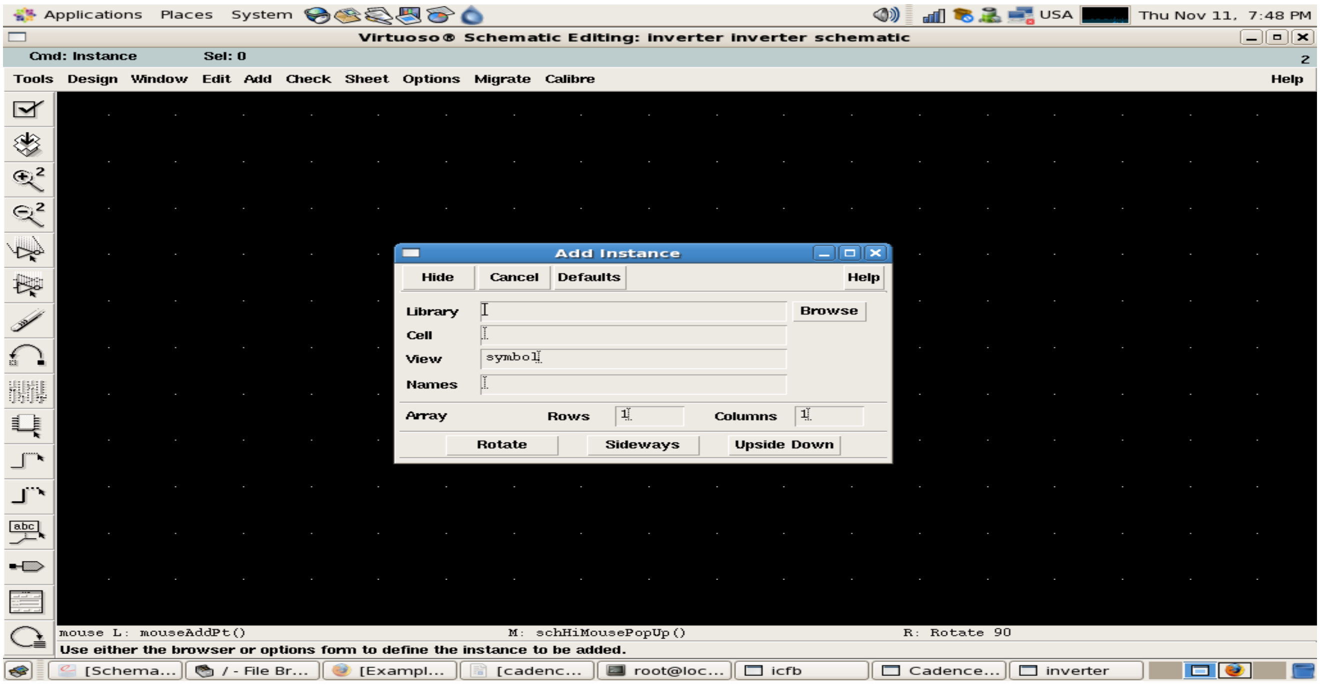
Messages

```
        but was defined in libFile '/usr/local/tsmc13/cds.lib' for Lib 'CP'.
Log file is "/usr/local/tsmc13/libManager.log".
```

Applications Places System
USA Thu Nov 11, 7:24 PM
File Edit View Design Manager Help
Show Categories
Show Files
Library
Cell
View
OK Cancel Defaults Help
[Schematic... [tsmc13 - F... [Example i... [cadence_t... root@local... icfb Cadence Li...

4. Inverter Schematic

a) After you open the previously created Inverter schematic, an empty window appears where you should place your components. To place an NMOS, select **Add->Instance** or use shortcut "**i**":



Virtuoso® Schematic Editing: inverter inverter schematic

Cmd: Instance   Sel: 0

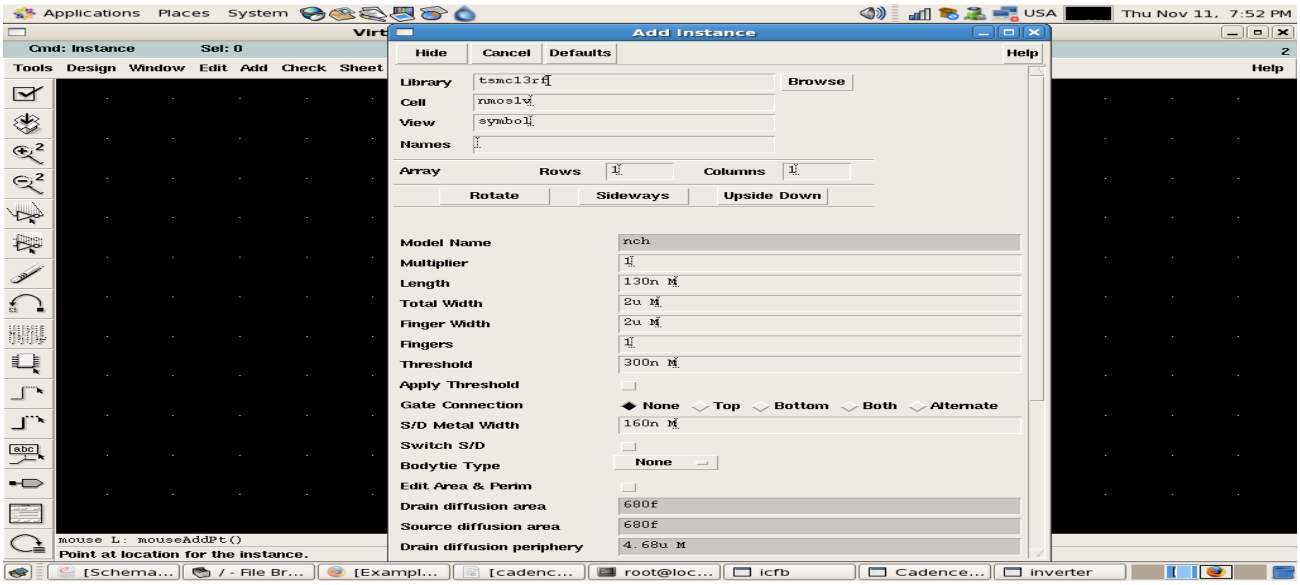
Add Instance

Library:
Cell:
View: symbol
Names:
Array   Rows 1   Columns 1

```
mouse L: mouseAddPt()                M: schHiMousePopUp()               R: Rotate 90
Use either the browser or options form to define the instance to be added.
```

Applications Places System
USA Thu Nov 11, 7:48 PM
Tools Design Window Edit Add Check Sheet Options Migrate Calibre Help
Hide Cancel Defaults Help
Browse
Rotate Sideways Upside Down
[Schema... / - File Br... [Exampl... [cadenc... root@loc... icfb Cadence... inverter

b) Click on **Browse->tsmc13rf->MOSFETs->nmos1v->symbol.** (Note: check the Show Categories button in the top left corner) Now, you can modify the transistor parameters, e.g. the width W, Length L, No. of fingers N, and Multipliers M for the NMOS.


Applications Places System
USA Thu Nov 11, 7:52 PM
Cmd: Instance Sel: 0
Tools Design Window Edit Add Check Sheet
Help
Add Instance
Hide Cancel Defaults Help
Browse
Rotate Sideways Upside Down
[Schema... [/ - File Br... [Exampl... [cadenc... root@loc... icfb Cadence... inverter

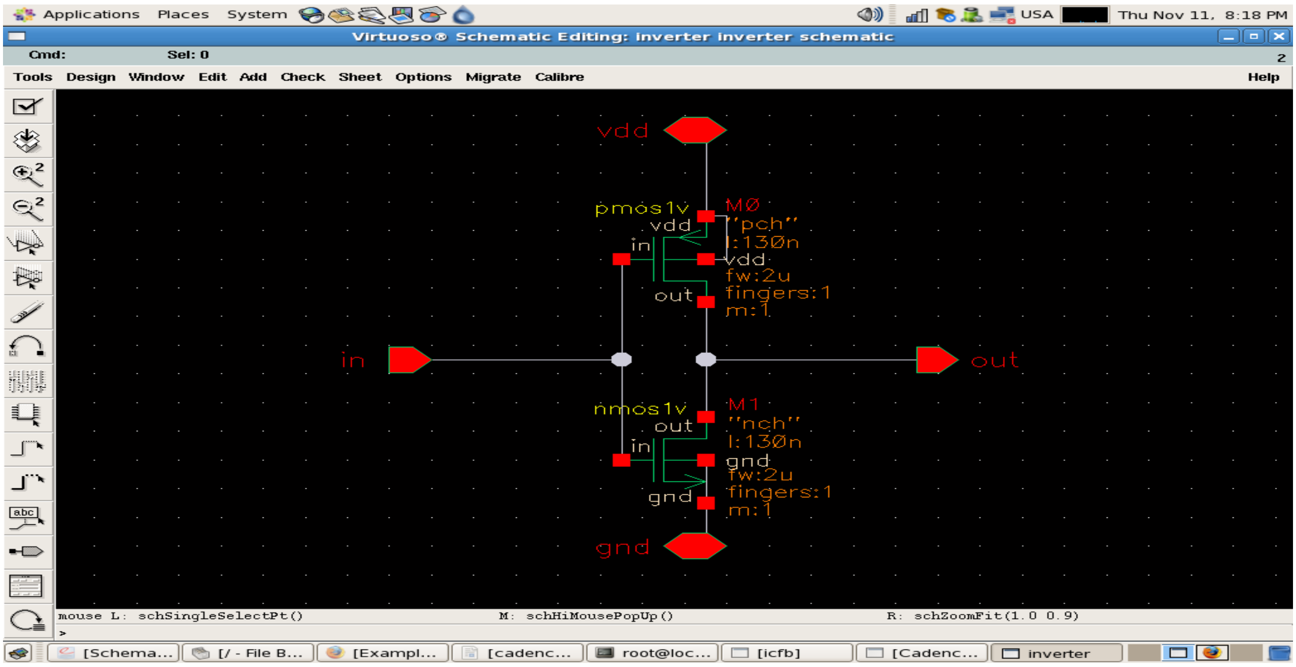
Library tsmc13rf
Cell nmos1v
View symbol
Names
Array Rows 1 Columns 1
Model Name nch
Multiplier 1
Length 130n M
Total Width 2u M
Finger Width 2u M
Fingers 1
Threshold 300n M
Apply Threshold □
Gate Connection ◆ None ◇ Top ◇ Bottom ◇ Both ◇ Alternate
S/D Metal Width 160n M
Switch S/D □
Bodytie Type None
Edit Area & Perim □
Drain diffusion area 680f
Source diffusion area 680f
Drain diffusion periphery 4.68u M

mouse L: mouseAddPt()
Point at location for the instance.

c) Repeat b) for the PMOS.

d) Then, to connect your components, you need wires. To add wires, select **add** ->**wires**, or simply use the shortcut "**w**".

e) Any circuit should have ports. Three types of ports exist: IN, OUT and INOUT. Our inverter schematic needs one input port IN, one output port OUT, and two input/output VDD and GND. To add a port: press **p**, then choose the name and the direction. The final schematic should be like this:


Applications Places System
USA Thu Nov 11, 8:18 PM
Virtuoso® Schematic Editing: inverter inverter schematic
Cmd: Sel: 0
Tools Design Window Edit Add Check Sheet Options Migrate Calibre
Help
vdd
pmos1v
vdd
in
out
M0
"pch"
l:130n
vdd
fw:2u
fingers:1
m:1
in
out
nmos1v
out
in
gnd
M1
"nch"
l:130n
gnd
fw:2u
fingers:1
m:1
gnd
[Schema... [/ - File B... [Exampl... [cadenc... root@loc... [icfb] [Cadenc... inverter

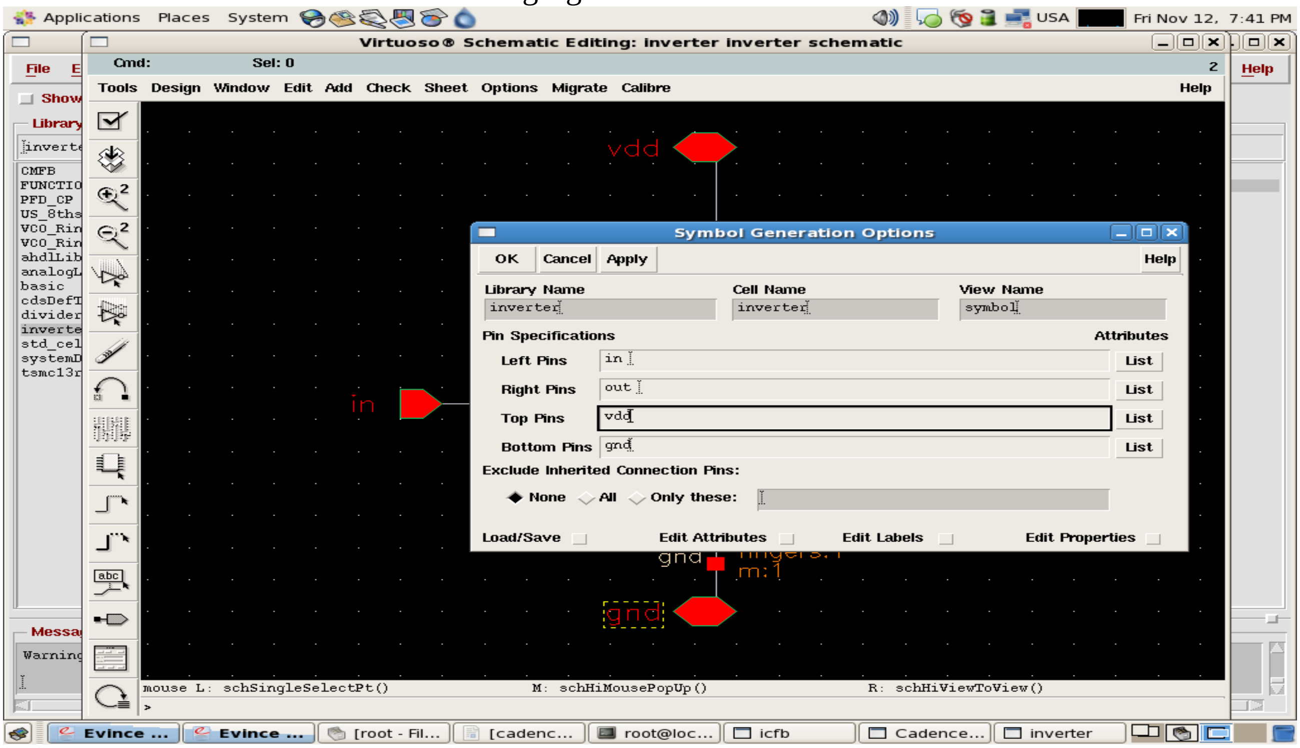
mouse L: schSingleSelectPt() M: schHiMousePopUp() R: schZoomFit(1.0 0.9)
>

5. Inverter Symbol

An auto-generation option for the symbol is integrated in Cadence's Virtuoso:

a) To do so, go to Inverter schematic, select **Design->Create Cellview->From Cellview** then press **OK** on the *Cellview From Cellview* window.

b) The *Symbol Generation Options* window appears where you can choose where you want to put your pins, to the left, right, top and bottom of the symbol and my choices are as in the following figure:



Virtuoso® Schematic Editing: inverter inverter schematic

Symbol Generation Options

Library Name: inverter
Cell Name: inverter
View Name: symbol

Pin Specifications
Left Pins: in
Right Pins: out
Top Pins: vdd
Bottom Pins: gnd

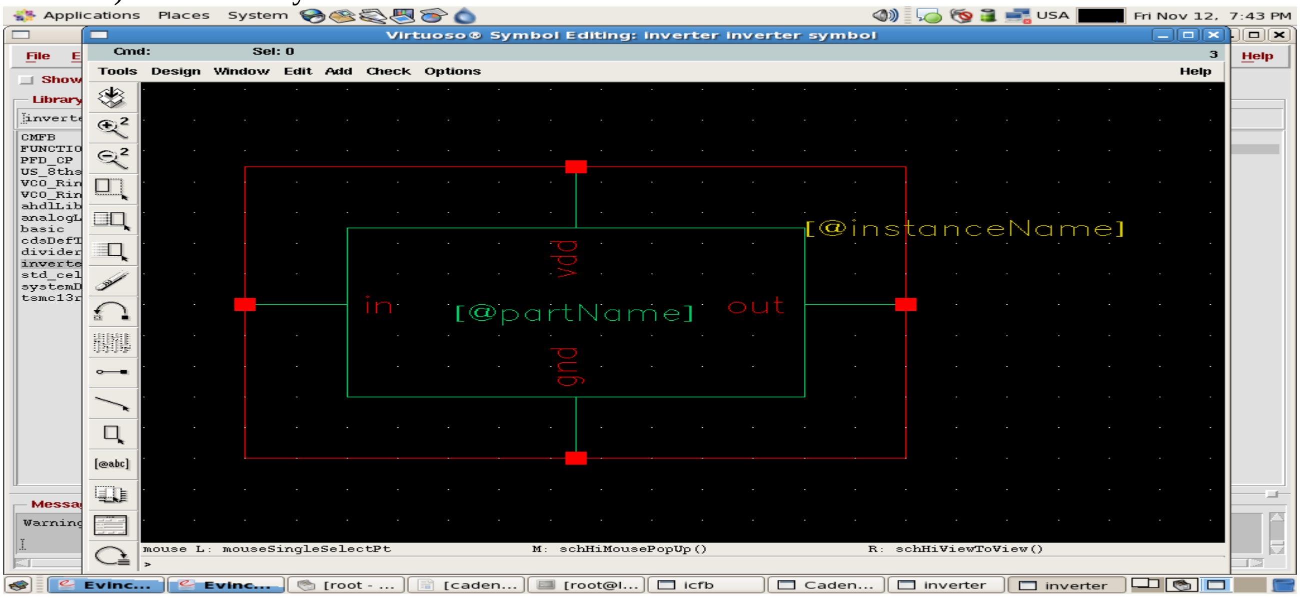
Exclude Inherited Connection Pins: ● None ○ All ○ Only these:

Load/Save □ Edit Attributes □ Edit Labels □ Edit Properties □

```
mouse L: schSingleSelectPt()          M: schHiMousePopUp()          R: schHiViewToView()
```

Applications Places System
Fri Nov 12, 7:41 PM
Cmd: Sel: 0
Tools Design Window Edit Add Check Sheet Options Migrate Calibre Help
OK Cancel Apply Help
List
vdd
in
gnd
m:1
Evince ... Evince ... [root - Fil... [cadenc... root@loc... icfb Cadence... inverter

c) The basic symbol is shown next:

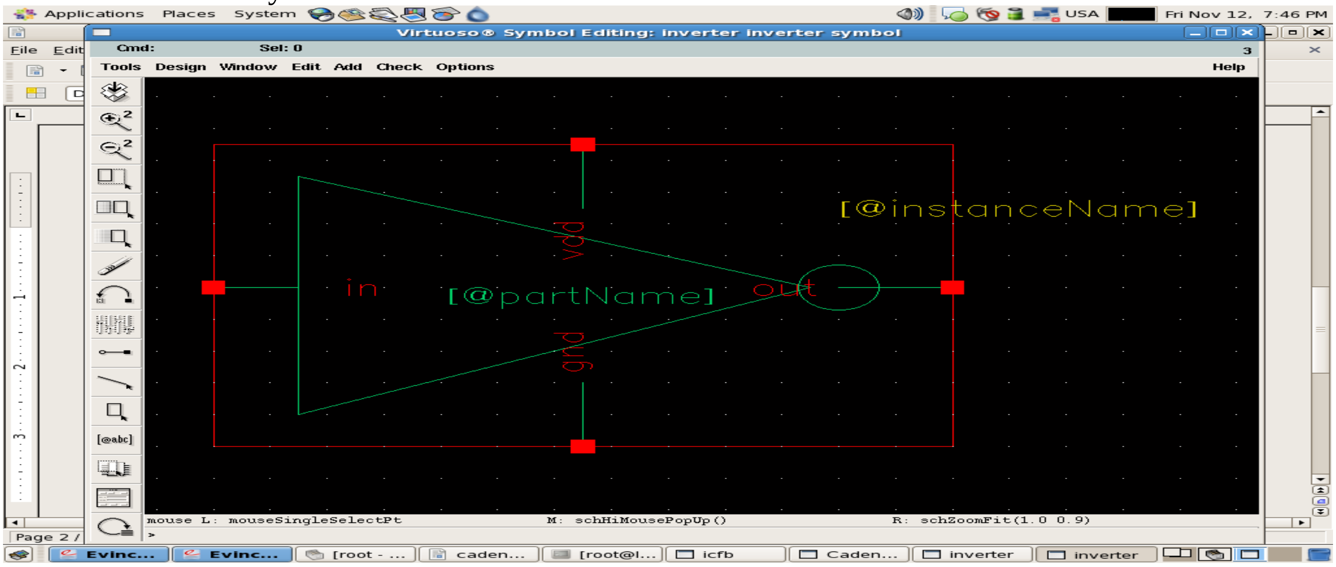


Virtuoso® Symbol Editing: inverter inverter symbol

```
mouse L: mouseSingleSelectPt()          M: schHiMousePopUp()          R: schHiViewToView()
```

Applications Places System
Fri Nov 12, 7:43 PM
Cmd: Sel: 0
Tools Design Window Edit Add Check Options Help
[@instanceName]
vdd
in
[@partName]
out
gnd
Evinc... Evinc... [root - ... [caden... [root@l... icfb Caden... inverter inverter

d) You can edit the symbol shape, select **Add->Shape->......** and draw the traditional inverter symbol as shown below:


Virtuoso® Symbol Editing: inverter inverter symbol
Tools Design Window Edit Add Check Options Help
[@instanceName]
[@partName]
in
out
vdd
gnd

Cmd: Sel: 0

mouse L: mouseSingleSelectPt M: schHiMousePopUp() R: schZoomFit(1.0 0.9)

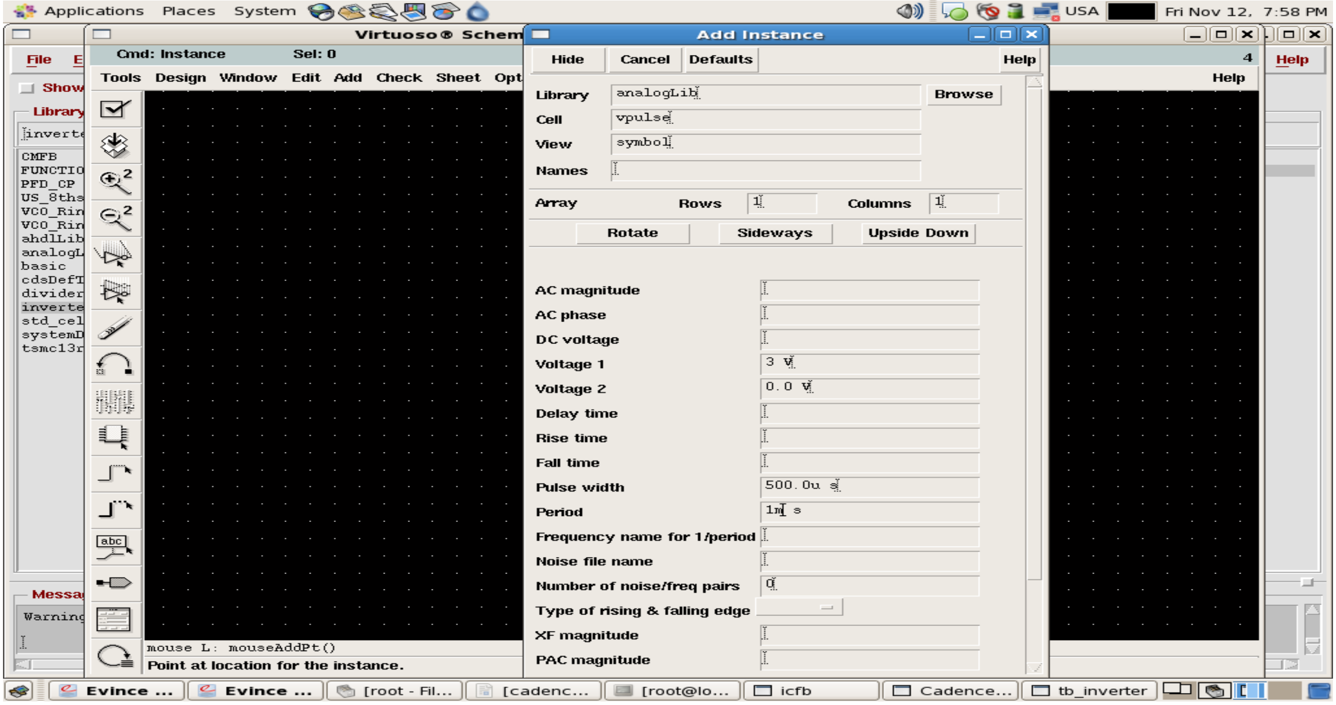
6. Test bench of inverter simulation

To simulate the inverter example, we simply need a DC source which will be connected between VDD and GND and a pulse source(square wave) that is connected to the input:

a) Create a new schematic cell view called “tb_inverter”.

b) Add our previously created inverter to it: press **i**, then choose your inverter and add it to the schematic.

c) To add the pulse source, press **i**, and click on **Browse** -> **analogLib** -> **Sources** -> **Independent** -> **vpulse** -> **Symbol**. In the following window, you can choose the characteristics of your square wave:



Cmd: Instance Sel: 0

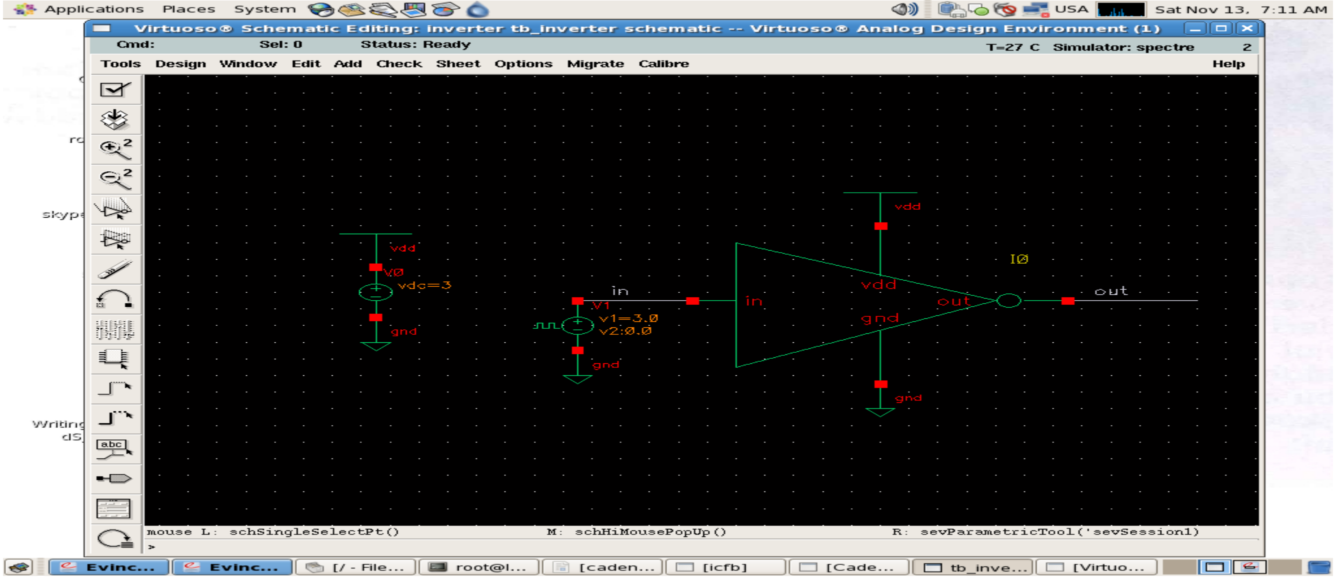
Library	analogLib
Cell	vpulse
View	symbol
Names	
Array Rows	1
Columns	1
AC magnitude	
AC phase	
DC voltage	
Voltage 1	3 V
Voltage 2	0.0 V
Delay time	
Rise time	
Fall time	
Pulse width	500.0u s
Period	1m s
Frequency name for 1/period	
Noise file name	
Number of noise/freq pairs	0
Type of rising & falling edge	
XF magnitude	
PAC magnitude	

mouse L: mouseAddPt()
Point at location for the instance.

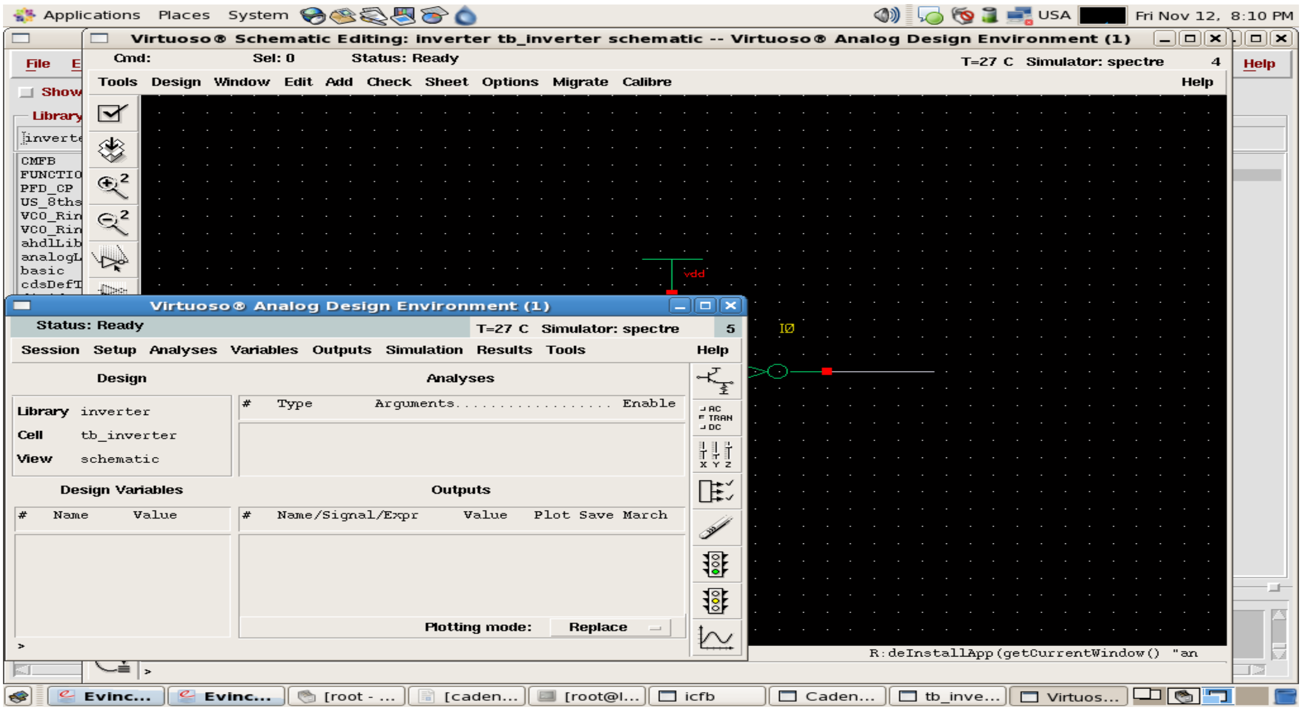
d) To add a DC source repeat 3 but with **vdc** from the **analogLib**. Don't forget to set its value to 3V

e) In electronics, we like to have VDD rails and GND rails which should be connected the DC source and these rails are connected to VDD and GND of each symbol. To add rails, press **i,** then, Click on **BROWSE** -> **analogLib** -> **Global** -> **vdd** -> **Symbol**, and the same for **gnd**.

f) You can label input and output nets by press **i**. You should end up with the following figure:


Applications Places System
USA Sat Nov 13, 7:11 AM
Virtuoso® Schematic Editing: inverter tb_inverter schematic -- Virtuoso® Analog Design Environment (1)
Cmd: Sel: 0 Status: Ready T=27 C Simulator: spectre 2
Tools Design Window Edit Add Check Sheet Options Migrate Calibre Help
vdd
V0
vdc=3
gnd
in
V1
v1=3.0
v2:0.0
gnd
vdd
I0
in
vdd
gnd
out
out
vdd
gnd
mouse L: schSingleSelectPt() M: schHiMousePopUp() R: sevParametricTool('sevSession1)
Evinc... Evinc... [/ - File... root@l... [caden... [icfb] [Cade... tb_inve... [Virtuo...

7. To begin simulation, select **Tools**->**Analog Environment**, the **ADE** window appears where we can handle simulations like below:


Applications Places System
USA Fri Nov 12, 8:10 PM
Virtuoso® Schematic Editing: inverter tb_inverter schematic -- Virtuoso® Analog Design Environment (1)
Cmd: Sel: 0 Status: Ready T=27 C Simulator: spectre 4
Tools Design Window Edit Add Check Sheet Options Migrate Calibre Help
Virtuoso® Analog Design Environment (1)
Status: Ready T=27 C Simulator: spectre 5
Session Setup Analyses Variables Outputs Simulation Results Tools Help
Design
Analyses
# Type Arguments.................. Enable
Design Variables
# Name Value
Outputs
# Name/Signal/Expr Value Plot Save March
AC TRAN DC
X Y Z
vdd
I0
R: deInstallApp(getCurrentWindow() "an
Evinc... Evinc... [root - ... [caden... [root@l... icfb Caden... tb_inve... Virtuos...

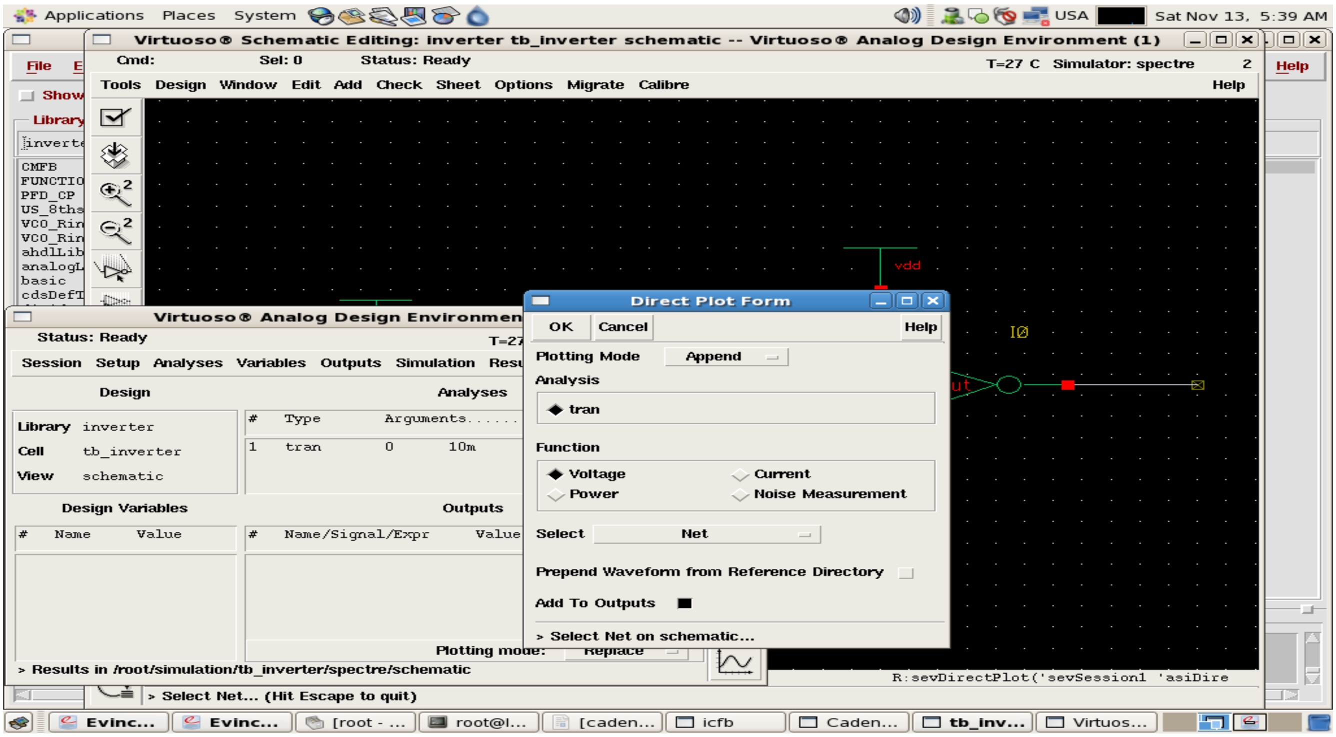
Library inverter
Cell tb_inverter
View schematic

Plotting mode: Replace

g) Click on **Choose Analysis** icon in the right panel, a window appears where you can make TRAN, AC, DC, PSS, PAC, ... analyses as you want, for the inverter exmaple, we'll use **TRAN** only and **Stop time** will be **10ms**.

h) To begin simulation, click on **Netlist and Run** icon (the green traffic light).

i) To see the waveform at input and output nets, you need to plot them. To do so, select **Results->Direct Plot->Plot Outputs** & click on the nets you wish to plot. Don't forget to check the **Add to outputs** option as in figure:


Applications Places System
Sat Nov 13, 5:39 AM
Virtuoso® Schematic Editing: inverter tb_inverter schematic -- Virtuoso® Analog Design Environment (1)
Cmd: Sel: 0 Status: Ready T=27 C Simulator: spectre 2
Tools Design Window Edit Add Check Sheet Options Migrate Calibre Help
vdd
Virtuoso® Analog Design Environment
Session Setup Analyses Variables Outputs Simulation Results
Direct Plot Form
OK Cancel Help

Status: Ready

Design	
Library	inverter
Cell	tb_inverter
View	schematic

#	Type	Arguments......
1	tran	0 10m

Plotting Mode: Append

Analysis: ◆ tran

Function: ◆ Voltage ◇ Current ◇ Power ◇ Noise Measurement

Select: Net

Prepend Waveform from Reference Directory □

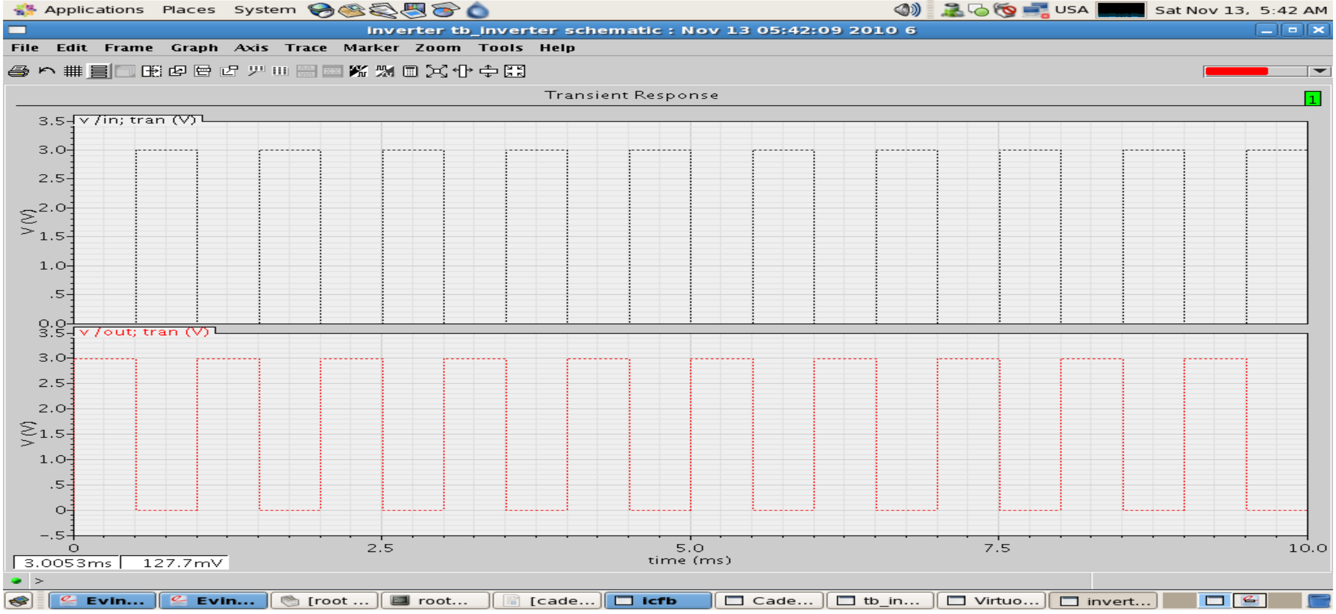
Add To Outputs ■

> Select Net on schematic...

> Results in /root/simulation/tb_inverter/spectre/schematic

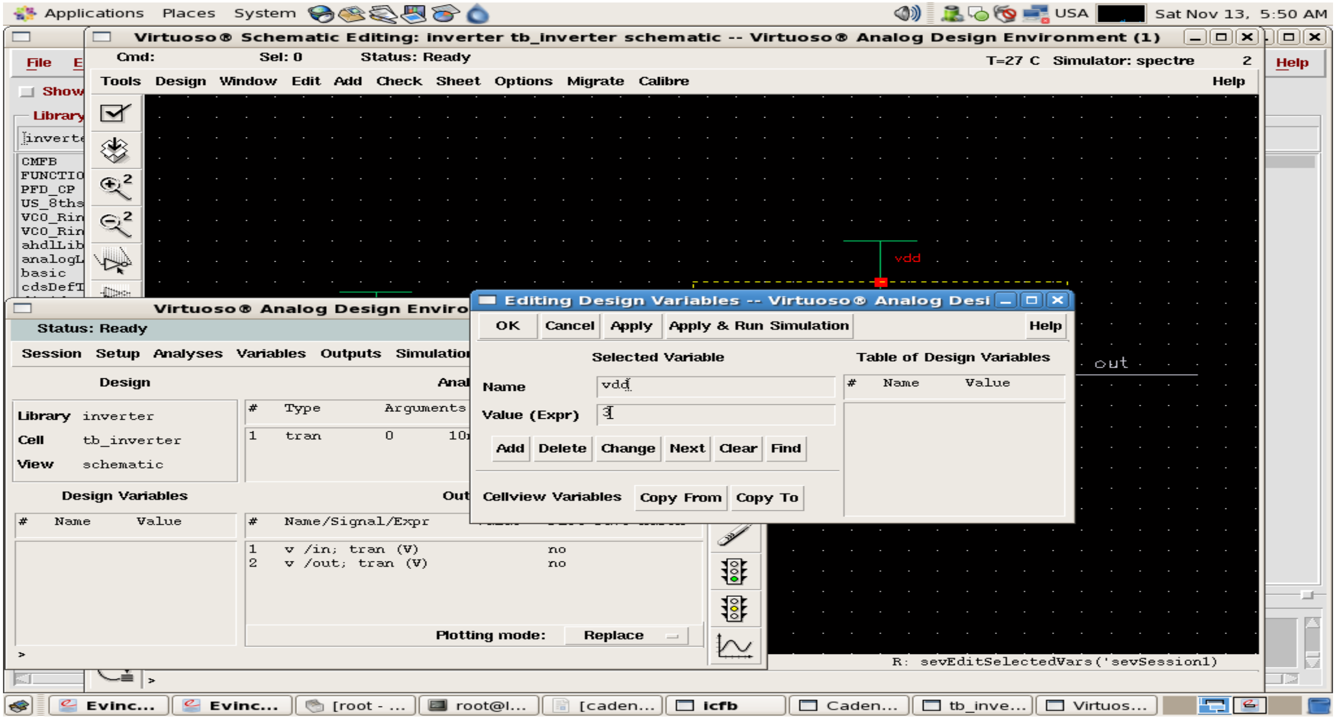
> Select Net... (Hit Escape to quit)

j) The plot of **in** and **out** nets is shown in the following figure:



7. More tips

a) You can declare a variable and use it to assign values to component parameters. To do so, in the ADE window, the simulation window, select **Variables**->**Edit** and the following window appears where you can declare a variable and set its value. My variable is **vdd** and its value equals 3 and then press **OK**. Then, in the tb_inverter schematic, set the DC source value to **vdd** and and re-simulate it. It should be ok.



Virtuoso® Schematic Editing: inverter tb_inverter schematic -- Virtuoso® Analog Design Environment (1)

Cmd: Sel: 0 Status: Ready T=27 C Simulator: spectre 2

Virtuoso® Analog Design Enviro…

Status: Ready

Library inverter
Cell tb_inverter
View schematic

#	Type	Arguments
1	tran	0 10…

#	Name/Signal/Expr		
1	v /in; tran (V)	no	
2	v /out; tran (V)	no	

Plotting mode: Replace

Editing Design Variables -- Virtuoso® Analog Desi…

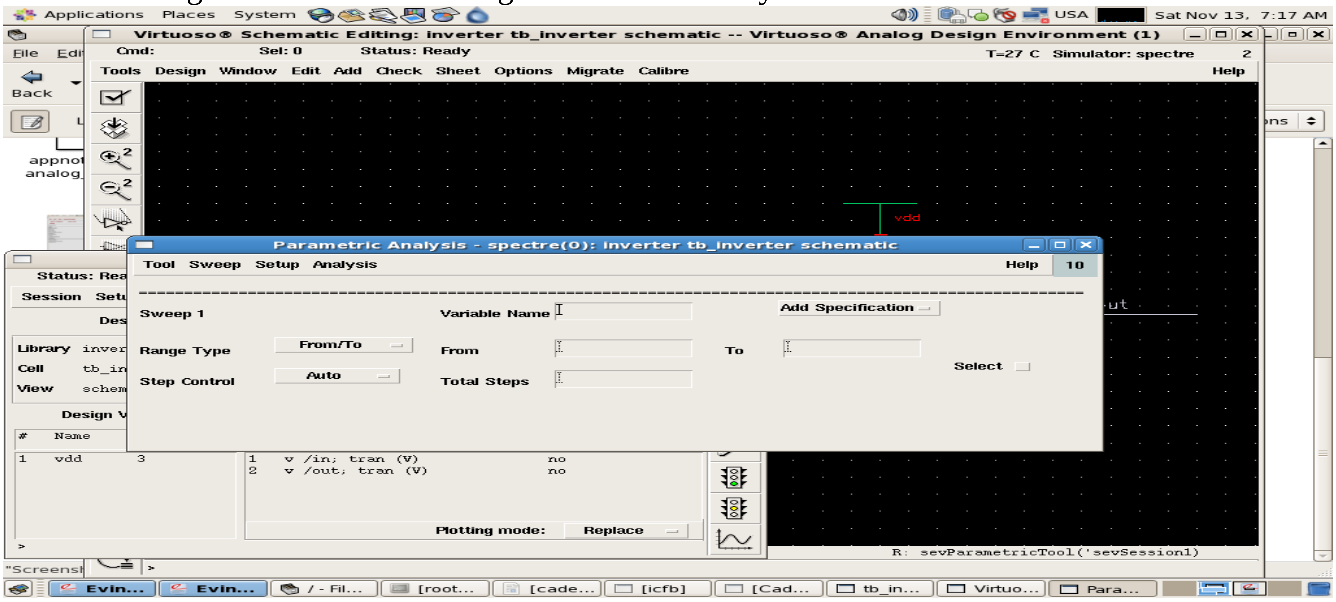
Selected Variable
Name vdd
Value (Expr) 3

R: sevEditSelectedVars('sevSession1')

Applications Places System
USA Sat Nov 13, 5:50 AM
File Edit
Show
Library
inverter
CMFB
FUNCTIO
PFD_CP
US_8ths
VCO_Rin
VCO_Rin
ahdlLib
analogL
basic
cdsDefT
Tools Design Window Edit Add Check Sheet Options Migrate Calibre Help
vdd
out
Session Setup Analyses Variables Outputs Simulation
Design
Design Variables
# Name Value
OK Cancel Apply Apply & Run Simulation Help
Table of Design Variables
# Name Value
Add Delete Change Next Clear Find
Cellview Variables Copy From Copy To
Evinc... Evinc... [root - ... root@l... [caden... icfb Caden... tb_inve... Virtuos...

b) The cadence simulator is called ***spectre***.

c) You can invoke **Tools**->**Parametric Analysis...** from the ADE window. The parametric analysis tool is a powerful tool that is used to do the same simulations with changing some specific variable. For example, you may need it to change the length of MOS and see the gain each time until you reach the desired value.



Virtuoso® Schematic Editing: inverter tb_inverter schematic -- Virtuoso® Analog Design Environment (1)

Cmd: Sel: 0 Status: Ready T=27 C Simulator: spectre 2

Parametric Analysis - spectre(0): inverter tb_inverter schematic

Sweep 1 Variable Name
Range Type From/To From To
Step Control Auto Total Steps

#	Name	Value
1	vdd	3

#	Name/Signal/Expr		
1	v /in; tran (V)	no	
2	v /out; tran (V)	no	

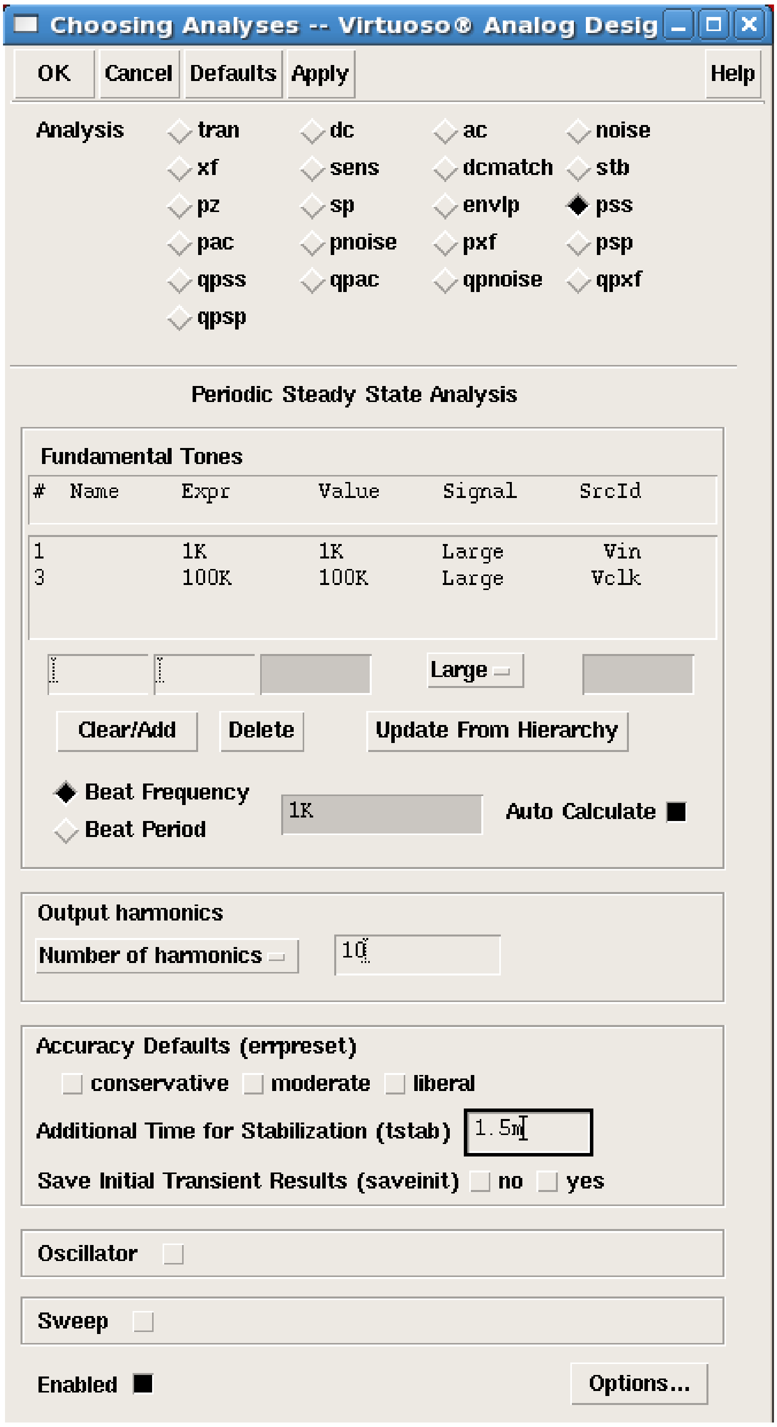
Plotting mode: Replace

R: sevParametricTool('sevSession1')

Applications Places System
USA Sat Nov 13, 7:17 AM
File Edit
Back
appnot
analog
Tools Design Window Edit Add Check Sheet Options Migrate Calibre Help
vdd
out
Tool Sweep Setup Analysis Help 10
Add Specification
Select
Status: Rea
Session Setu
Library inver
Cell tb_in
View schem
Design V
"Screensh
Evin... Evin... / - Fil... [root... [cade... [icfb] [Cad... tb_in... Virtuo... Para...

8. Periodic Steady-State (PSS) Analysis

This analysis computes the steady-state response of the circuit with only the clock and a transient input applied. This analysis is also a prerequisite to the periodic small-signal analyses that follow because it sets the periodic operating point.



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

Periodic Steady State Analysis

Fundamental Tones

#	Name	Expr	Value	Signal	SrcId
1		1K	1K	Large	Vin
3		100K	100K	Large	Vclk

Large

● Beat Frequency
○ Beat Period
1K
Auto Calculate ■

Output harmonics

Number of harmonics 10

Accuracy Defaults (errpreset)

□ conservative □ moderate □ liberal

Additional Time for Stabilization (tstab) 1.5m

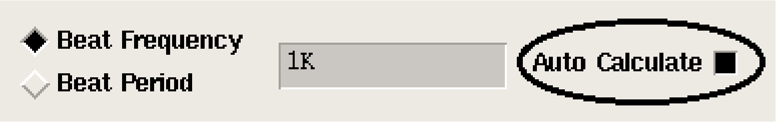
Save Initial Transient Results (saveinit) □ no □ yes

Oscillator □

Sweep □

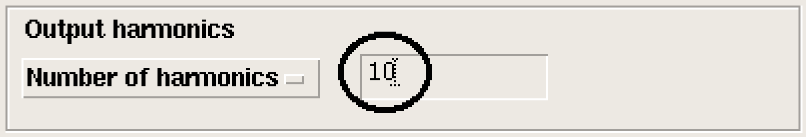
Enabled ■

The fundamental frequency of the steady-state response (called beat frequency) is usually the smallest frequency pertaining to all of the applied sources. For example, in a switched-capacitor filter, we may have two main sources with frequencies: f_{in} and f_{clk}. The output will have a fundamental frequency = $f_{in} < f_{clk}$. If you need to change the beat-frequency, uncheck the *Auto-Calculate* option: (This is not usually needed)



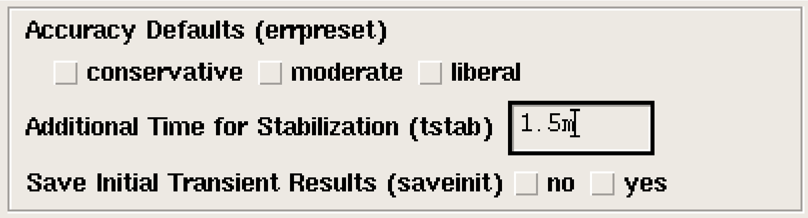
◆ Beat Frequency
◇ Beat Period
1K
Auto Calculate ■

The number of harmonics (multiples of fin) in the output spectrum can also be set using the corresponding parameter:



Output harmonics
Number of harmonics 10

The PSS analysis, like a conventional transient analysis, starts off from an initial condition. If you do not specify an initial condition, then Spectre uses a DC analysis to determine the initial condition. During a DC analysis, the clocks are not operating, so the opamps have no feedback, causing their outputs to get stuck at the rails. The initial condition computed during the DC analysis causes the filter to react wildly in transient analysis, bouncing off the rails several times before it settles. This can cause convergence difficulties during PSS analysis that could be resolved in several ways. One approach that works in this case is to simply specify zero initial conditions for all capacitors in the circuit. This avoids a bad starting point. Another approach that is widely applicable when confronting convergence problems in PSS analysis is to use the *tstab* parameter, which delays the start of the PSS shooting loop:



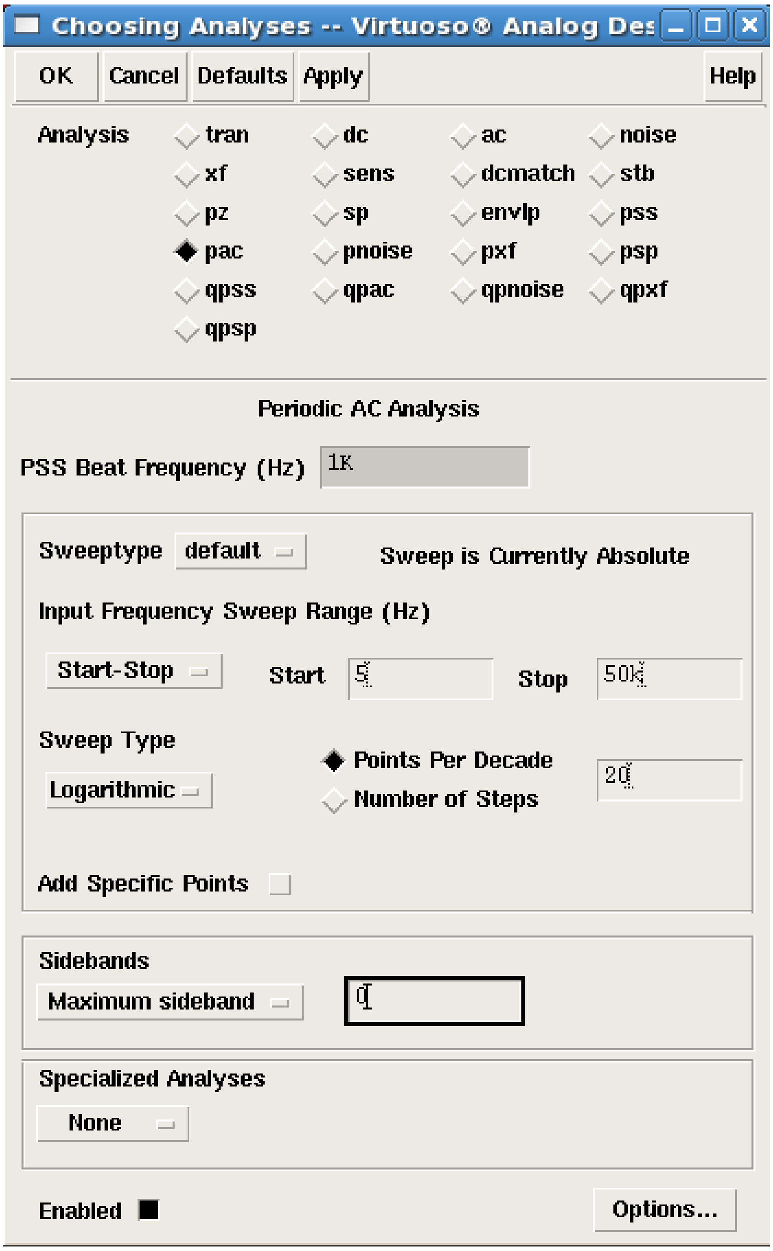
Accuracy Defaults (errpreset)
□ conservative □ moderate □ liberal
Additional Time for Stabilization (tstab) 1.5m
Save Initial Transient Results (saveinit) □ no □ yes

For example, if *tstab* = 1.5 ms, this causes the transient analysis to be performed to at least *t* = 1.5 ms before the PSS analysis attempts to compute the steady-state solution. Waiting until *t* = 1.5 ms allows the simulator to get beyond the difficult time when the filter is bouncing off the rails, which results in PSS analysis converging easily.

In order to further improve the efficiency of the PSS analysis, the phasing of the clock signals relative to the simulation interval is chosen carefully. It is best to have the simulation interval begin and end at points where the signals are not changing abruptly.

9. Periodic-AC (PAC) and Periodic-XF (PXF) Analyses

With Spectre there are two analyses commonly used to measure small-signal gain. They are referred to as the PAC and PXF analyses. We will be more concerned with the PAC analysis:



Choosing Analyses -- Virtuoso® Analog Des

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

Periodic AC Analysis

PSS Beat Frequency (Hz) 1K

Sweeptype default Sweep is Currently Absolute

Input Frequency Sweep Range (Hz)

Start-Stop Start 5 Stop 50K

Sweep Type
Logarithmic ◆ Points Per Decade 20
◇ Number of Steps

Add Specific Points □

Sidebands
Maximum sideband 0

Specialized Analyses
None

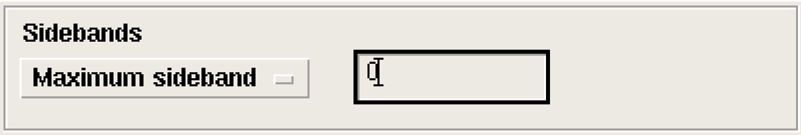
Enabled ■ Options...

The gain is measured by applying a small sinusoid at the input and measuring the amplitude of the signal at the output. This is somewhat complicated by two factors. First, the signal at the output may only approximate a sinusoid. Remember that it is being sampled at the clock rate. This means that care must be taken when defining what is meant by the output amplitude. With Spectre, the output amplitude is defined by first performing a Fourier analysis on the output signal and taking the amplitude to be the amplitude of the first harmonic. Second, the input and output sinusoids may be at different frequencies. If the input frequency is small relative to the clock frequency, then generally the input and output are at the same frequency. But it is also common for the input frequency to near or above the clock frequency, in which aliasing acts to down convert the input signal to baseband. In this case, designers are purposely exploiting aliasing to down-convert a signal. Spectre handles both cases.

The PAC (periodic AC) analysis is similar in concept to SPICE's AC analysis, with the difference being that Spectre's PAC analysis performs a small-signal analysis about a periodically-varying operating point whereas SPICE's AC analysis assumes a time-invariant operating point. Thus, with SPICE the clock is effectively disabled before running an AC analysis, whereas Spectre's PAC analysis effectively performs the AC analysis with the clock present. With the clock applied, the circuit naturally translates signals from one frequency to another. A signal applied at frequency f_{in} can be translated to any frequency $f_{\text{out}} = kf_{\text{clk}} + f_{\text{in}}$ (1) where k is any integer.

Similarly, a signal observed at a frequency f_{out} could have been injected into the circuit at any frequency $f_{\text{in}} = kf_{\text{clk}} + f_{\text{out}}$ (2). As such, the PAC and PXF analyses compute the transfer functions. In particular, they compute the transfer functions associated with a range of values of k as defined by the *Maximum Sideband* parameter, where PAC computes the transfer functions described by (1) and PXF computes the ones described by (2).

The circuit is linearized before performing the PAC analysis, so the output amplitude will scale linearly with the input amplitude, so generally one simply uses the most convenient value for the input amplitude, in this case 1 since it allows us to directly compute the gain at the output. The PAC magnitude option is adjusted by clicking on the source, viewing its properties (by pressing Q) and scrolling till you find *PAC Magnitude*. A sweep range is then set as in a conventional AC analysis. The last parameter specified on the PAC analysis is *Maximum Sideband*:



Sidebands

Maximum sideband 0

This parameter specifies how many output sidebands should be computed. Remember that the output frequency need not be the same as the input frequency. The output frequencies are given by $f_{\text{out}}(k) = kf_{\text{clk}} + f_{\text{in}}$ (3) where k is the sideband index. In the case where we are computing the direct gain, the output frequency is the same as the input frequency, so $k = 0$. The PAC analysis will compute all of the sidebands for $k \leq$ *Maximum Sideband* since we are only interested in the case where $k = 0$ the *Maximum Sideband* parameter is set to 0.

The result usually consists of two components, a direct feed through piece due to the signal in track mode, and a sin(*x*)/*x* piece due to the signal in hold mode. Often, one is only interested in the hold mode component. Spectre naturally considers the whole signal, to get it to ignore the track mode component one has to create a new signal that does not contain the track mode component. To do so, an idealized sample-and-hold is constructed with Verilog-A. It is used to sample the output of the track-and-hold during the hold phase.

The PAC analysis allows one to specify a particular signal at a particular frequency at the input and then to determine the response at the output at any sideband. If the opposite is occurring, i.e. the transfer functions from the various sidebands at the input to a single output sideband are desired. This would require running many PAC analyses, each with the input set to run over a different range of frequencies and the output produced for a different sideband. Instead, one can use a PXF analysis. Whereas the PAC analysis computes the output signal at every node and every sideband given a single input, the PXF analysis computes the transfer function from every input source at every sideband to a single output.

With the same analysis one could also determine the transfer function from each source. For example, the power supply rejection is available for free with a PXF analysis because it computes the transfer function from the supply source (at all sidebands) to the output.