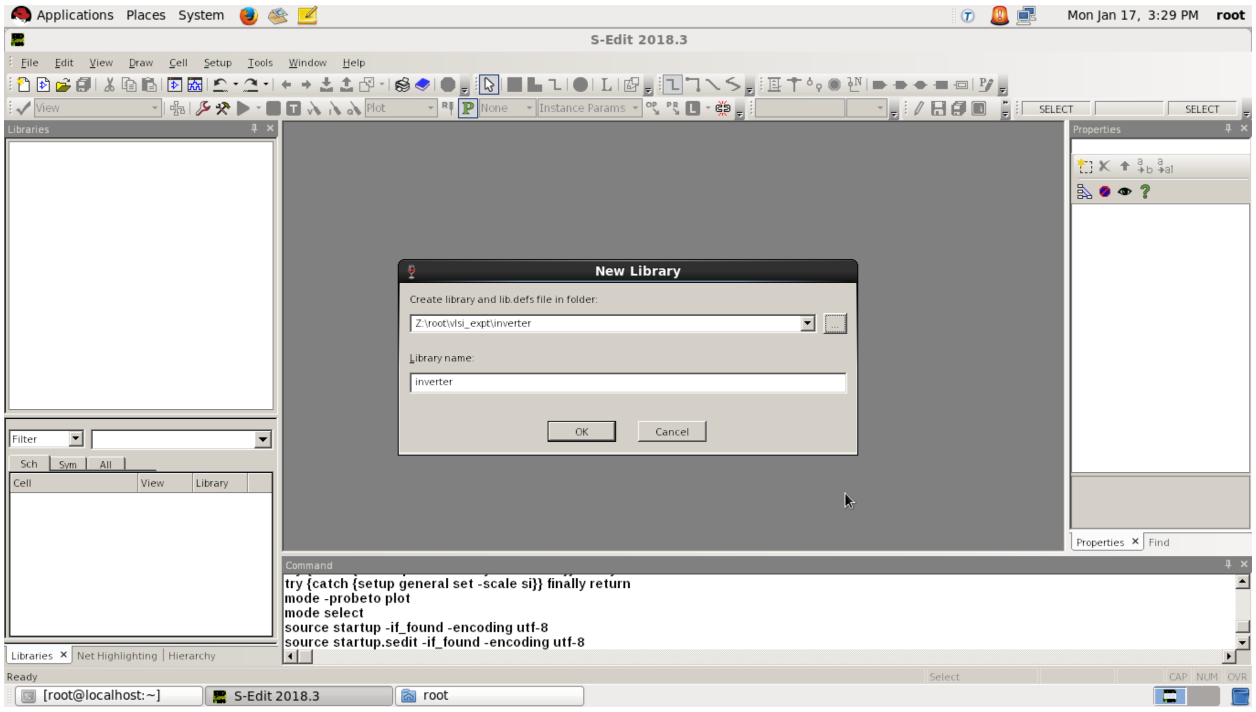
Simulation Steps for using Mentor Graphics

Schematic Design:

- ❖ Create a folder (to save your designs).
- ❖ Right click on **Desktop** (or inside your folder) → select **open in Terminal,** then type the following commands
    - **cd**
    - **csh**
    - **source /home/MentorGraphics/cshrc/hep.cshrc**
    - **sedit (to open Schematic editor)**
- ❖ **S-Edit** window will open, minimize **open in terminal** window (do not close it)
- ❖ Create new Library:
    - ➢ In **S-Edit window**, go to **File** → **New** → **New Library,** click on browse icon to select your **Folder** to save library → give library name (for example: inverter) → **OK**



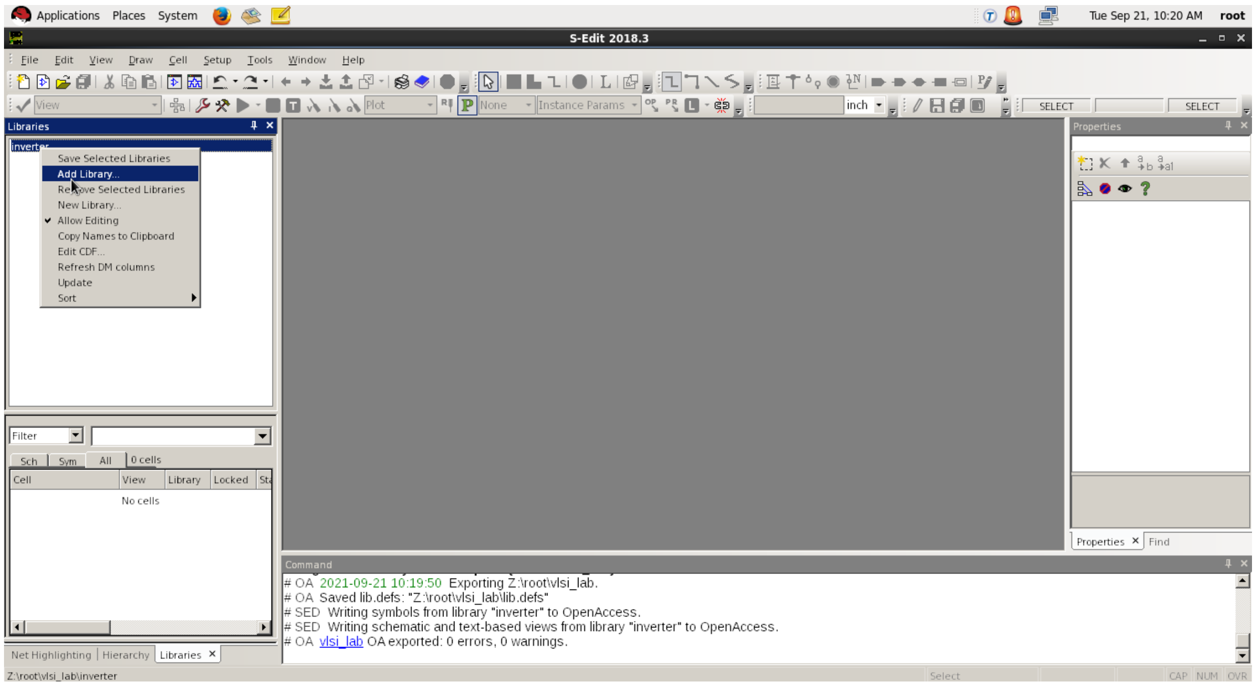
Create library and lib.defs file in folder: Z:\root\vlsi_expt\inverter

Library name: inverter

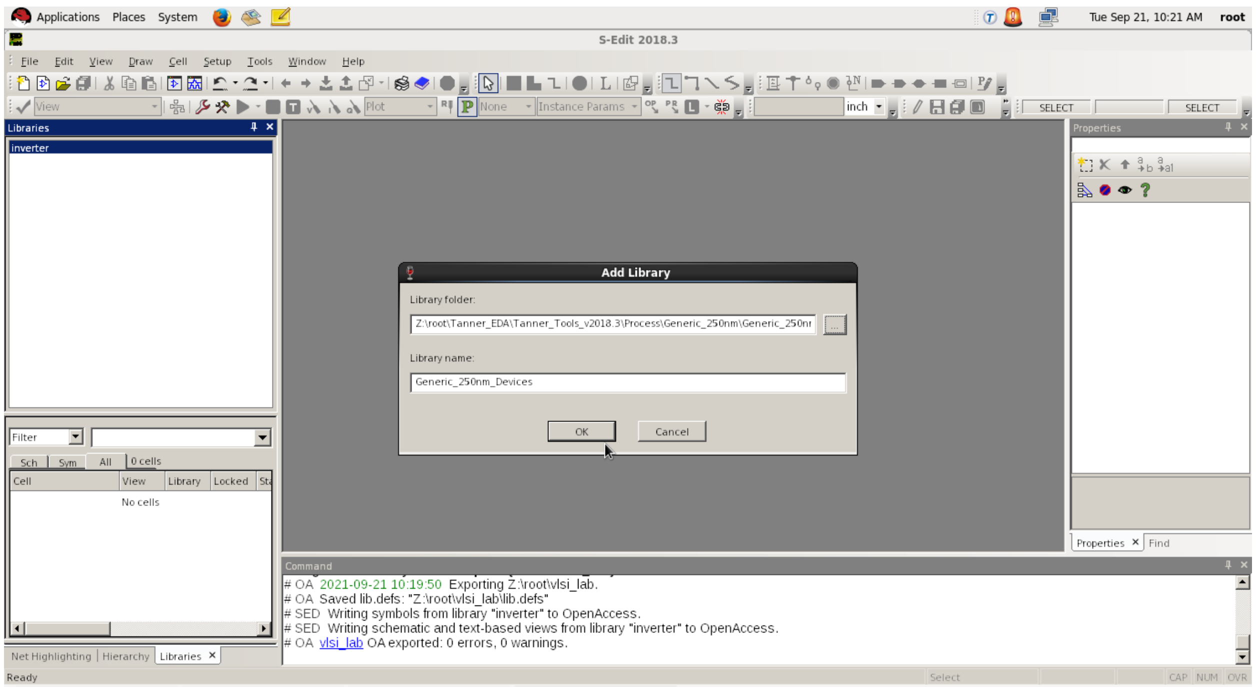
```
try {catch {setup general set -scale si}} finally return
mode -probeto plot
mode select
source startup -if_found -encoding utf-8
source startup.sedit -if_found -encoding utf-8
```

- ❖ Add following Libraries to your library
    i. **Generic_250nm_Devices**
    ii. **Misc**
    iii. **SPICE_Sources**
    iv. **SPICE_Measure**
    v. **SPICE_Plot**
- ❖ To add **Generic_250nm_Devices** library
    - ➢ Right click on your **Library Name** → click on **Add library,** click on browse icon to select **Generic_250nm_Devices** library from the following path.

root → Tanner_EDA → Tanner_Tools_v2018.3 → Process → Generic_250nm → Generic_ 250nm_Devices → OK



```
# OA  2021-09-21 10:19:50  Exporting Z:\root\vlsi_lab.
# OA  Saved lib.defs: "Z:\root\vlsi_lab\lib.defs"
# SED  Writing symbols from library "inverter" to OpenAccess.
# SED  Writing schematic and text-based views from library "inverter" to OpenAccess.
# OA  vlsi_lab OA exported: 0 errors, 0 warnings.
```



Library folder: Z:\root\Tanner_EDA\Tanner_Tools_v2018.3\Process\Generic_250nm\Generic_250nr

Library name: Generic_250nm_Devices

```
# OA  2021-09-21 10:19:50  Exporting Z:\root\vlsi_lab.
# OA  Saved lib.defs: "Z:\root\vlsi_lab\lib.defs"
# SED  Writing symbols from library "inverter" to OpenAccess.
# SED  Writing schematic and text-based views from library "inverter" to OpenAccess.
# OA  vlsi_lab OA exported: 0 errors, 0 warnings.
```

- ❖ To add **Misc** library
  - ➢ Right click on your **Library Name** → click on **Add library,** click on browse icon to select **Misc** library from the following path.

  root → Tanner_EDA → Tanner_Tools_v2018.3 → Process → Standard_Libraries → Misc → OK

- ❖ To add **SPICE_Sources** library
  - ➢ Right click on your **Library Name** → click on **Add library,** click on browse icon to select **SPICE_Sources** library from the following path.

  root → Tanner_EDA → Tanner_Tools_v2018.3 → Process → Standard_Libraries → SPICE_Sources → OK

- ❖ To add **SPICE_Plot** library
  - ➢ Right click on your **Library Name** → click on **Add library,** click on browse icon to select **SPICE_Plot** library from the following path.

  root → Tanner_EDA → Tanner_Tools_v2018.3 → Process → Standard_Libraries → SPICE_Plot → OK

- ❖ To add **SPICE_Measure** library
  - ➢ Right click on your **Library Name** → click on **Add library,** click on browse icon to select **SPICE_Measure** library from the following path.

root → Tanner_EDA → Tanner_Tools_v2018.3 → Process → Standard_Libraries → SPICE_Measure → OK

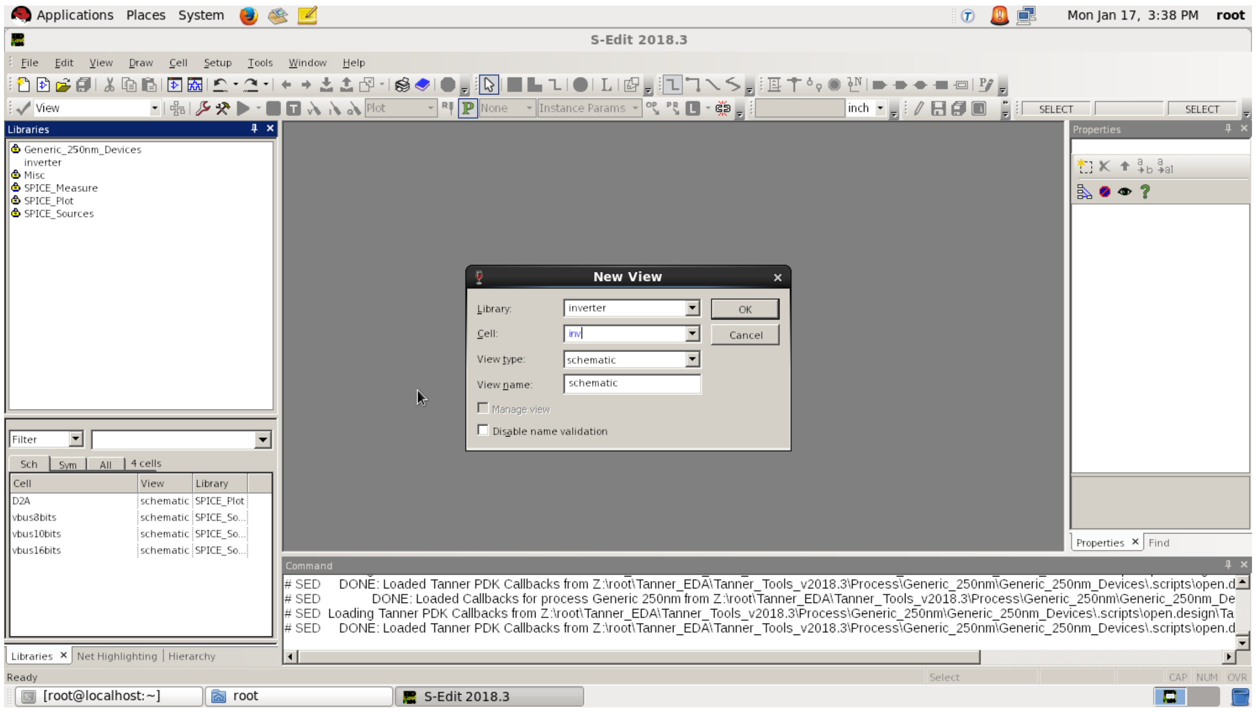
To create schematic:

❖ In **S-Edit window,** go to **Cell → New View (**press **n** from the keyboard**),** give cell name **→ OK**

Library → **inverter (for example)**

Cell → **inv (for example)**

View Type → **Schematic**



New View

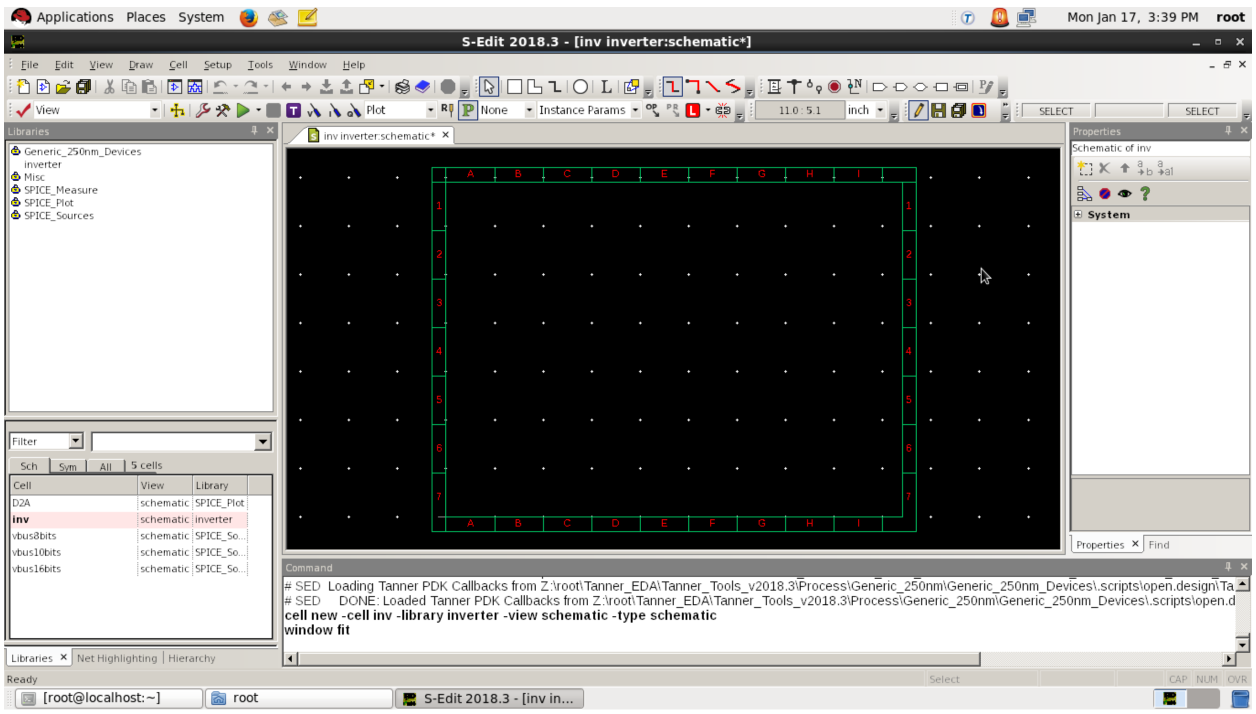
Library:	inverter
Cell:	inv
View type:	schematic
View name:	schematic

☐ Manage view
☐ Disable name validation

OK Cancel

Cell	View	Library
D2A	schematic	SPICE_Plot
vbus8bits	schematic	SPICE_So...
vbus10bits	schematic	SPICE_So...
vbus16bits	schematic	SPICE_So...

Command
```
# SED    DONE: Loaded Tanner PDK Callbacks from Z:\root\Tanner_EDA\Tanner_Tools_v2018.3\Process\Generic_250nm\Generic_250nm_Devices\.scripts\open.d
# SED          DONE: Loaded Callbacks for process Generic 250nm from Z:\root\Tanner_EDA\Tanner_Tools_v2018.3\Process\Generic_250nm\Generic_250nm_De
# SED  Loading Tanner PDK Callbacks from Z:\root\Tanner_EDA\Tanner_Tools_v2018.3\Process\Generic_250nm\Generic_250nm_Devices\.scripts\open.design\Ta
# SED    DONE: Loaded Tanner PDK Callbacks from Z:\root\Tanner_EDA\Tanner_Tools_v2018.3\Process\Generic_250nm\Generic_250nm_Devices\.scripts\open.d
```



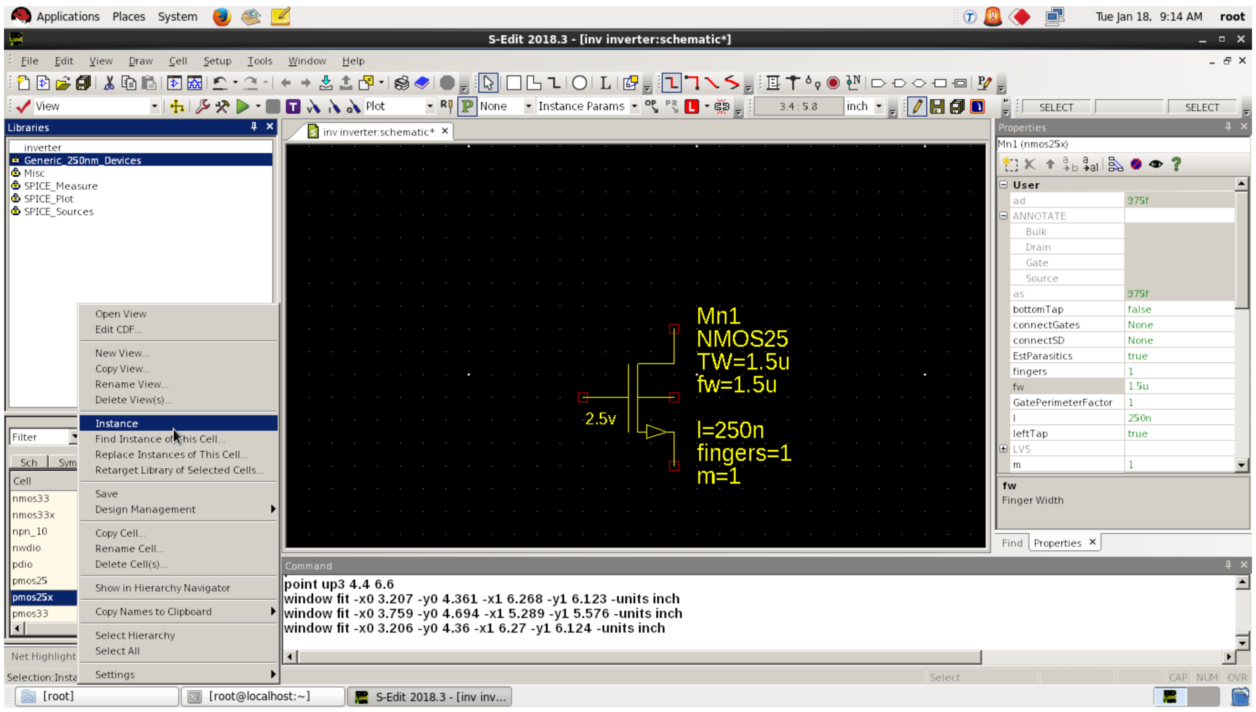
Cell	View	Library
D2A	schematic	SPICE_Plot
inv	schematic	inverter
vbus8bits	schematic	SPICE_So...
vbus10bits	schematic	SPICE_So...
vbus16bits	schematic	SPICE_So...

```
# SED  Loading Tanner PDK Callbacks from Z:\root\Tanner_EDA\Tanner_Tools_v2018.3\Process\Generic_250nm\Generic_250nm_Devices\.scripts\open.design\Ta
# SED    DONE: Loaded Tanner PDK Callbacks from Z:\root\Tanner_EDA\Tanner_Tools_v2018.3\Process\Generic_250nm\Generic_250nm_Devices\.scripts\open.d
cell new -cell inv -library inverter -view schematic -type schematic
window fit
```

- press **Home** key in the keyboard to **fit the design** into window.
- use **scroll** to **zoom in** and **zoom out** schematic window**.**

❖ **To place devices (pMOS, nMOS) on schematic window:**

- Select **Generic_250nm_Devices** library & then select devices (pMOS or nMOS) → right click on the device → select **Instance** → place it (left click) on the schematic window → click on done (or press **Esc** key in the keyboard)



ad	975f
ANNOTATE	
Bulk	
Drain	
Gate	
Source	
as	975f
bottomTap	false
connectGates	None
connectSD	None
EstParasitics	true
fingers	1
fw	1.5u
GatePerimeterFactor	1
l	250n
leftTap	true
LVS	
m	1

```
point up3 4.4 6.6
window fit -x0 3.207 -y0 4.361 -x1 6.268 -y1 6.123 -units inch
window fit -x0 3.759 -y0 4.694 -x1 5.289 -y1 5.576 -units inch
window fit -x0 3.206 -y0 4.36 -x1 6.27 -y1 6.124 -units inch
```

- press **r** key from the keyboard to rotate device
- press **v** key from the keyboard flip device vertically
- press **h** key from the keyboard to flip device horizontally

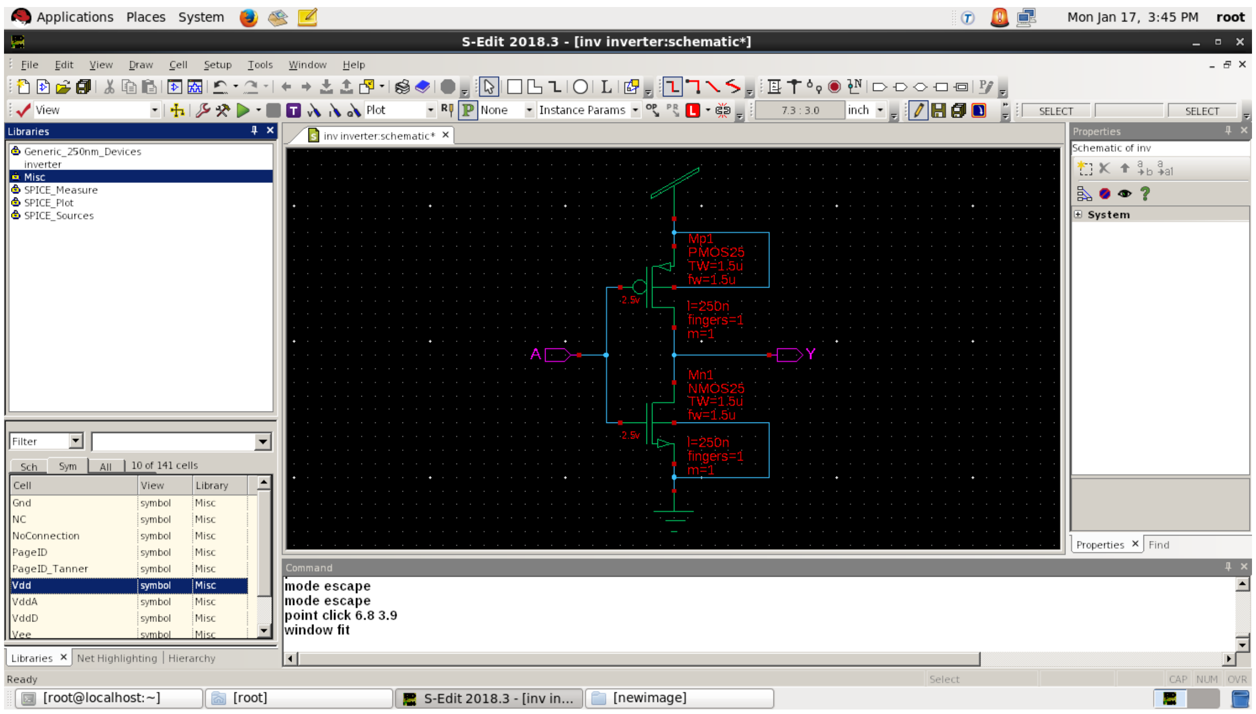
❖ **To place Vdd, GND on schematic window:**

- Select **Misc library** & then select **Vdd** → right click on the symbol → select **Instance** → place it (left click) on the schematic window → click on done (or press **Esc** key in the keyboard)
- Select **Misc library** & then select **GND** → right click on the symbol → select **Instance** → place it (left click) on the schematic window → click on done (press **Esc** key in the keyboard)

❖ **To place terminals on schematic window:**

- Select **In port** for input terminal
- Select **Out port** for output terminal

❖ Click on **Wire** & do the necessary connections.



```
mode escape
mode escape
point click 6.8 3.9
window fit
```

❖ To verify the design, go to **Tools** → **Design check** → **View**

Simulation:

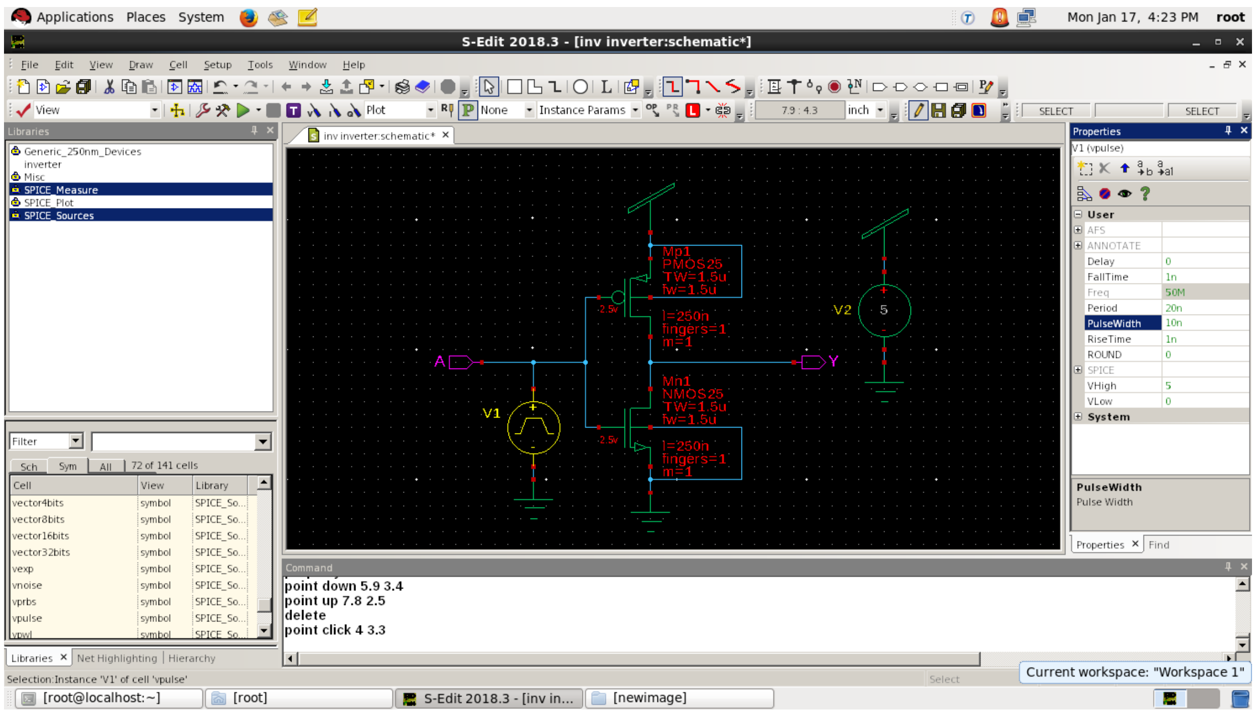
Transient Analysis:

❖ **For Schematic window add voltage sources**

➢ Add **Vpulse** to input terminal

- Select **SPICE_Sources** library → select **Vpulse** symbol → right click on the symbol → select **Instance** → place it (left click) on the schematic window → click on done (press **Esc** key in the keyboard)
- Select **Vpulse** source on the schematic window → Change its properties

Fall Time	1ns
Rise Time	1ns
Pulse Width	10ns
Period	20ns



72 of 141 cells

Cell	View	Library
vector4bits	symbol	SPICE_So...
vector8bits	symbol	SPICE_So...
vector16bits	symbol	SPICE_So...
vector32bits	symbol	SPICE_So...
vexp	symbol	SPICE_So...
vnoise	symbol	SPICE_So...
vprbs	symbol	SPICE_So...
vpulse	symbol	SPICE_So...
vpwl	symbol	SPICE_So...

Properties: V1 (vpulse)

User	
AFS	
ANNOTATE	
Delay	0
FallTime	1n
Freq	50M
Period	20n
PulseWidth	10n
RiseTime	1n
ROUND	0
SPICE	
VHigh	5
VLow	0
System	

PulseWidth
Pulse Width

```
point down 5.9 3.4
point up 7.8 2.5
delete
point click 4 3.3
```

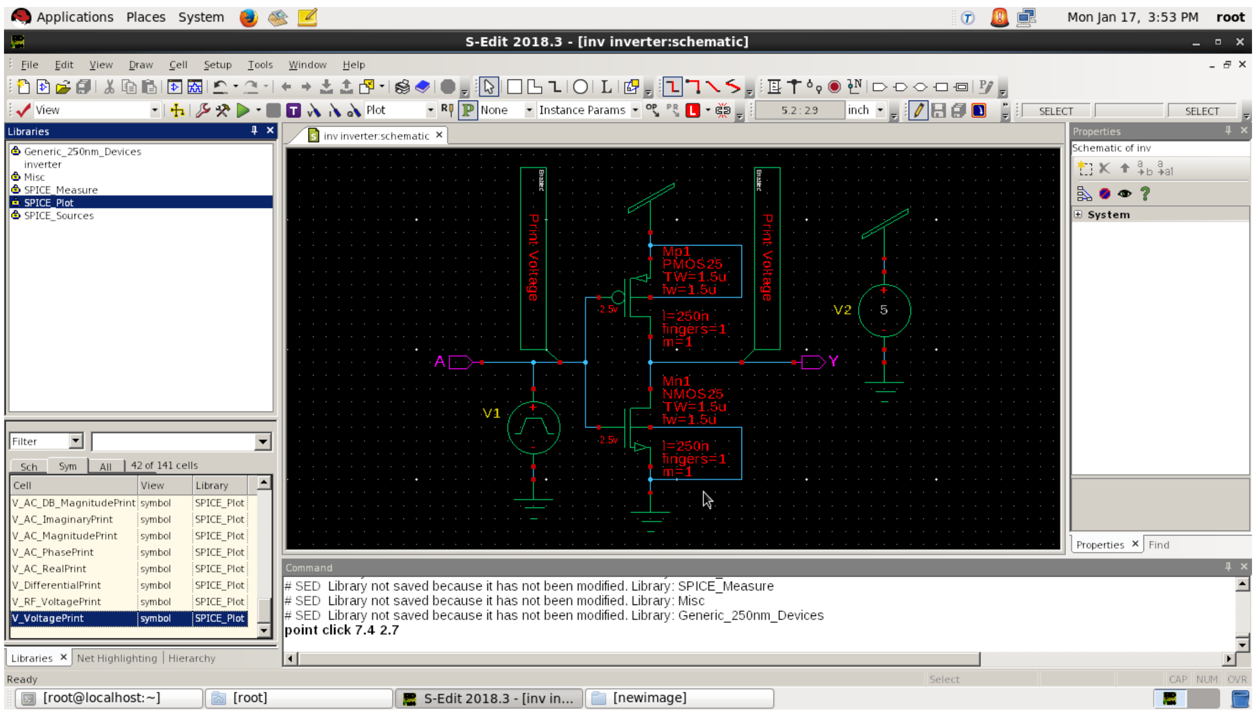
Selection:Instance 'V1' of cell 'vpulse'

- ➢ Add **Vdc** to Vdd terminal
  - ▪ Select **SPICE_Sources** library → select **vdc** symbol → right click on the symbol → select **Instance** → place it (left click) on the schematic window → click on done (press **Esc** key in the keyboard)

❖ Click on **Wire** & do the necessary connections.

❖ **For Schematic window add plot icon**

- ➢ Add **voltage plot** symbol
  - ▪ Select **SPICE_plot** library → select **V_VoltagePrint** symbol → right click on the symbol → select **Instance** → place it (left click) on the schematic window → click on done (press **Esc** key in the keyboard)



Cell	View	Library
V_AC_DB_MagnitudePrint	symbol	SPICE_Plot
V_AC_ImaginaryPrint	symbol	SPICE_Plot
V_AC_MagnitudePrint	symbol	SPICE_Plot
V_AC_PhasePrint	symbol	SPICE_Plot
V_AC_RealPrint	symbol	SPICE_Plot
V_DifferentialPrint	symbol	SPICE_Plot
V_RF_VoltagePrint	symbol	SPICE_Plot
V_VoltagePrint	symbol	SPICE_Plot

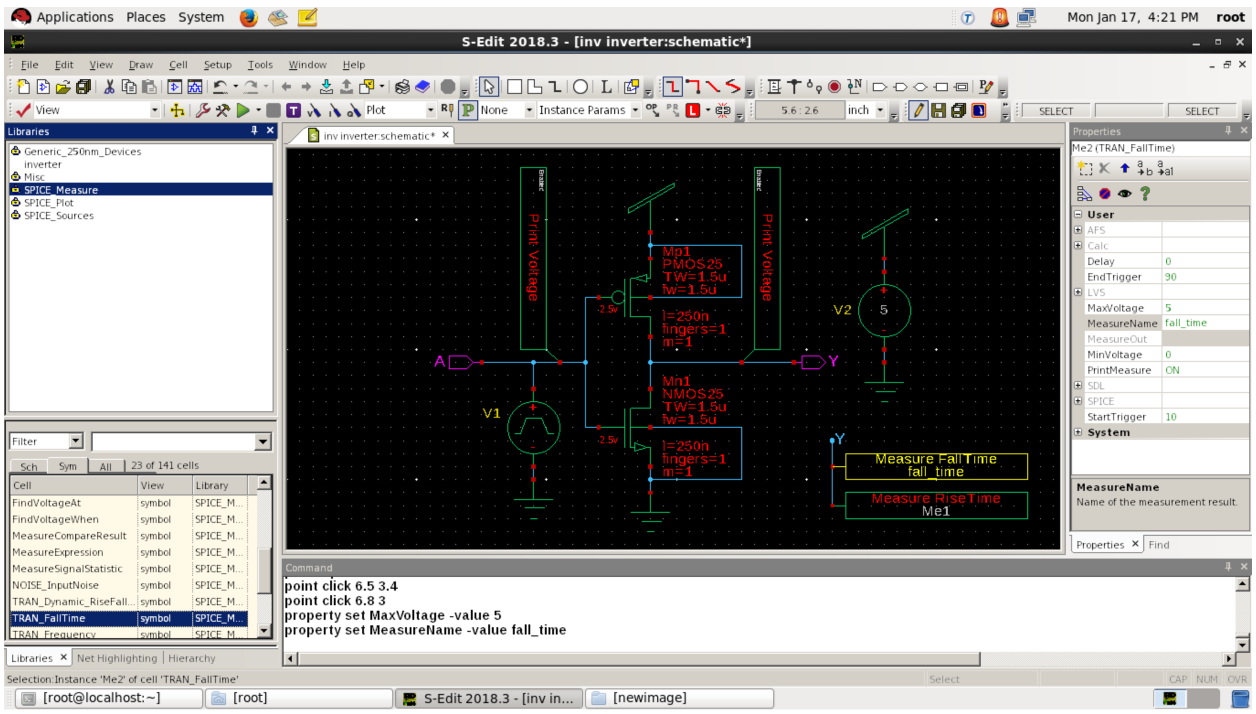
Command

```
# SED  Library not saved because it has not been modified. Library: SPICE_Measure
# SED  Library not saved because it has not been modified. Library: Misc
# SED  Library not saved because it has not been modified. Library: Generic_250nm_Devices
point click 7.4 2.7
```

❖ For Schematic window add measure icon

- ➢ Add **rise time and fall time** symbol
  - Select **SPICE_Measure** library → select **TRAN_FallTime** symbol → right click on the symbol → select **Instance** → place it (left click) on the schematic window → click on done (press **Esc** key in the keyboard)
  - Select **SPICE_Measure** library → select **TRAN_RiseTime** symbol → right click on the symbol → select **Instance** → place it (left click) on the schematic window → click on done (press **Esc** key in the keyboard)
  - Click on **Wire** & connect **TRAN_FallTime** and **TRAN_RiseTime** icon
  - Select net label → place it in the terminal
  - Change the properties of rise time and fall time symbol

MaxVoltage	5
MeasureName	Rise_time
MeasureName	fall_time



Cell	View	Library
FindVoltageAt	symbol	SPICE_M...
FindVoltageWhen	symbol	SPICE_M...
MeasureCompareResult	symbol	SPICE_M...
MeasureExpression	symbol	SPICE_M...
MeasureSignalStatistic	symbol	SPICE_M...
NOISE_InputNoise	symbol	SPICE_M...
TRAN_Dynamic_RiseFall...	symbol	SPICE_M...
TRAN_FallTime	symbol	SPICE_M...
TRAN_Frequency	symbol	SPICE_M...

Properties: Me2 (TRAN_FallTime)

Delay	0
EndTrigger	90
MaxVoltage	5
MeasureName	fall_time
MinVoltage	0
PrintMeasure	ON
StartTrigger	10

MeasureName
Name of the measurement result.

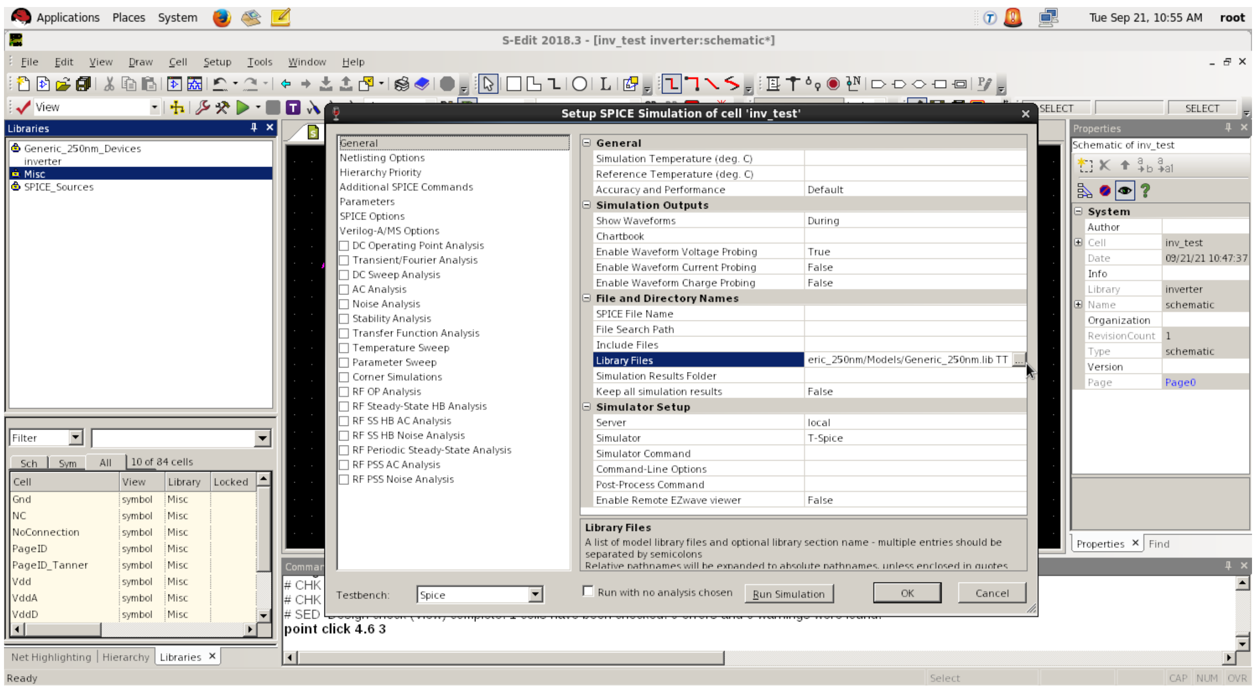
```
point click 6.5 3.4
point click 6.8 3
property set MaxVoltage -value 5
property set MeasureName -value fall_time
```

Selection:Instance 'Me2' of cell 'TRAN_FallTime'

- Go to **Tools** → **Design Check** → **View.**
- Go to **setup** → **SPICE Simulation**
  - Select **General**
    - Enable waveform voltage probing → **True**
    - Click on browse icon to select library from the following path.

root → **Tanner_EDA** → **Tanner_Tools_v2018.3** → **Process** → **Generic_250nm** → **Models** → **Generic_250nm.lib** → **OK**

Click on **the library path**, at the end giv**e space** & type **TT**

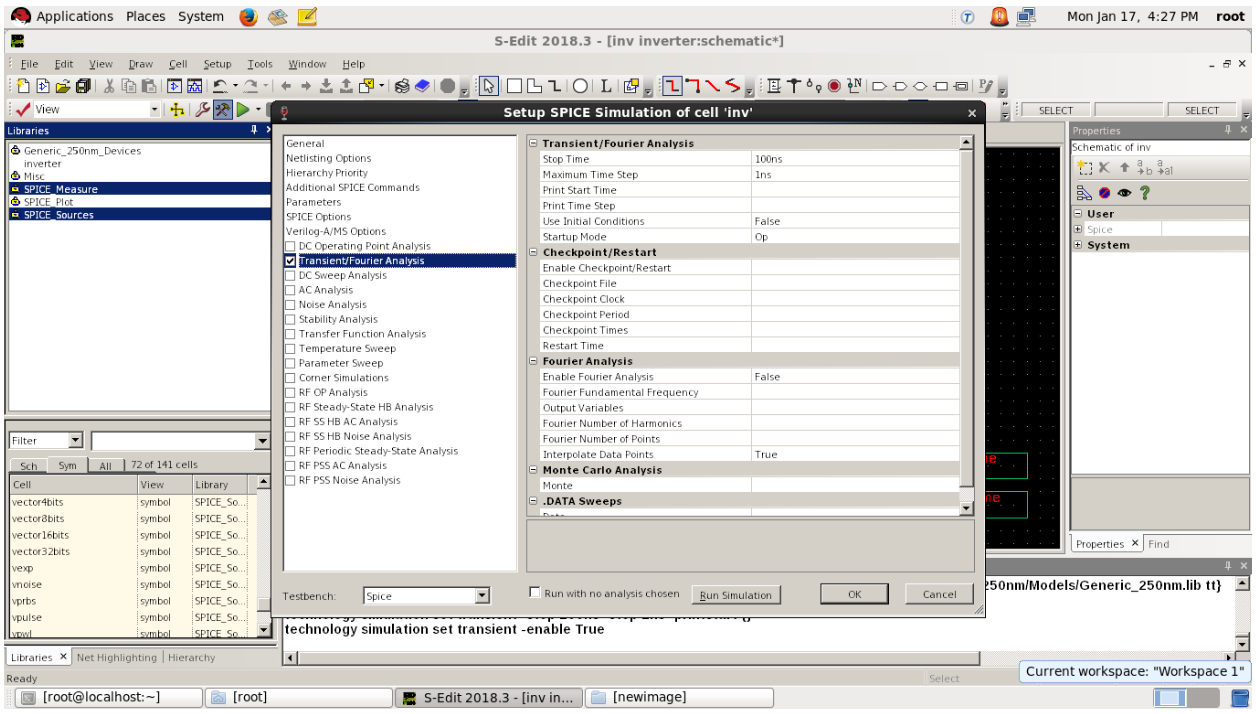


General	
Simulation Temperature (deg. C)	
Reference Temperature (deg. C)	
Accuracy and Performance	Default
Simulation Outputs	
Show Waveforms	During
Chartbook	
Enable Waveform Voltage Probing	True
Enable Waveform Current Probing	False
Enable Waveform Charge Probing	False
File and Directory Names	
SPICE File Name	
File Search Path	
Include Files	
Library Files	eric_250nm/Models/Generic_250nm.lib TT
Simulation Results Folder	
Keep all simulation results	False
Simulator Setup	
Server	local
Simulator	T-Spice
Simulator Command	
Command-Line Options	
Post-Process Command	
Enable Remote EZwave viewer	False

Library Files
A list of model library files and optional library section name - multiple entries should be separated by semicolons
Relative pathnames will be expanded to absolute pathnames, unless enclosed in quotes

❖ **Select Transient / Fourier Analysis** → click on the square to the left of Transient / Fourier Analysis and give following parameters.

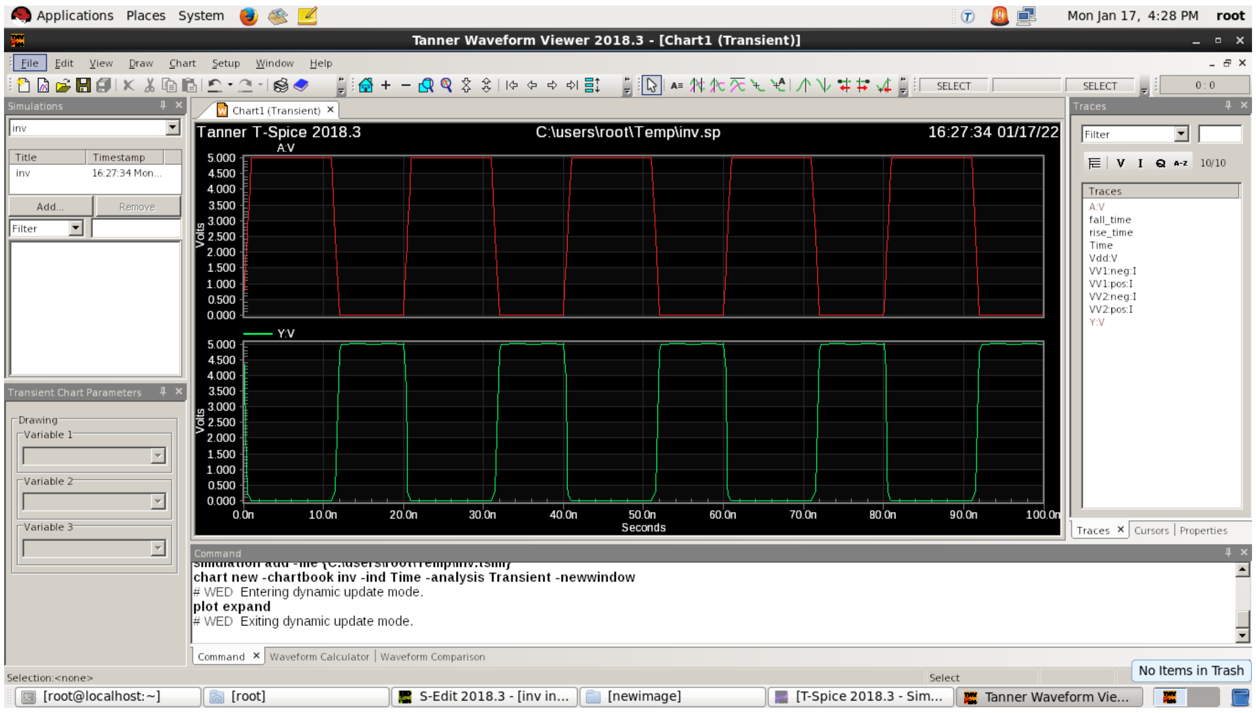
Stop Time	100ns
Maximum time step	1ns



Transient/Fourier Analysis	
Stop Time	100ns
Maximum Time Step	1ns
Print Start Time	
Print Time Step	
Use Initial Conditions	False
Startup Mode	Op
Checkpoint/Restart	
Enable Checkpoint/Restart	
Checkpoint File	
Checkpoint Clock	
Checkpoint Period	
Checkpoint Times	
Restart Time	
Fourier Analysis	
Enable Fourier Analysis	False
Fourier Fundamental Frequency	
Output Variables	
Fourier Number of Harmonics	
Fourier Number of Points	
Interpolate Data Points	True
Monte Carlo Analysis	
Monte	
.DATA Sweeps	

technology simulation set transient -enable True

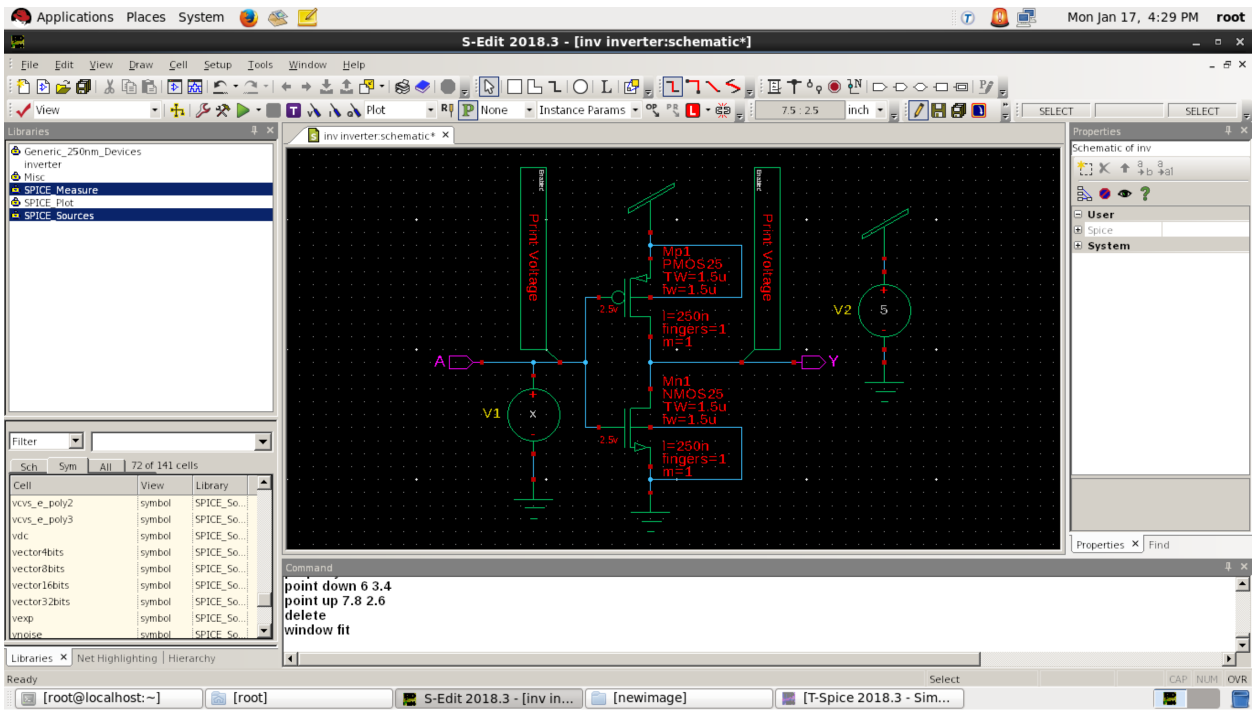
- ❖ Click **OK.**
- ❖ Click on **Start Simulation**, waveform window will be open.



```
simulation add -file {C:\users\root\Temp\inv.tsim}
chart new -chartbook inv -ind Time -analysis Transient -newwindow
# WED  Entering dynamic update mode.
plot expand
# WED  Exiting dynamic update mode.
```

DC Analysis:

- ❖ In Schematic window remove **Vpulse** from the input terminal and add **Vdc** to input terminal.
- ❖ Select **Vdc** and change its **properties** (remove voltage value and give parameter name **x)**.



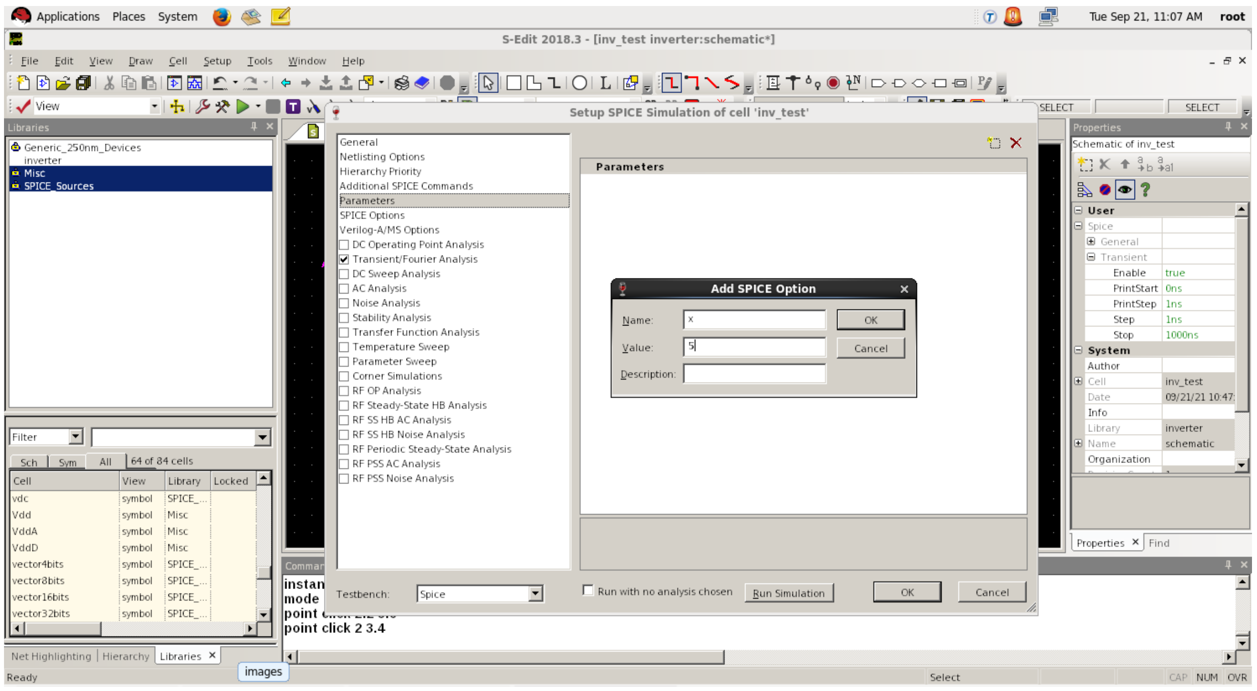
```
point down 6 3.4
point up 7.8 2.6
delete
window fit
```

- ❖ Go to **setup** → **SPICE Simulation**
- ❖ Select **Parameter** → click on Add SPICE Option (square icon on the right side) → give following parameters

  Name → **x (for example)**

  Value → **5**

  Click on **ok**


S-Edit 2018.3 - [inv_test inverter:schematic*]

Setup SPICE Simulation of cell 'inv_test'

Parameters

Add SPICE Option

Name: x

Value: 5

Description:

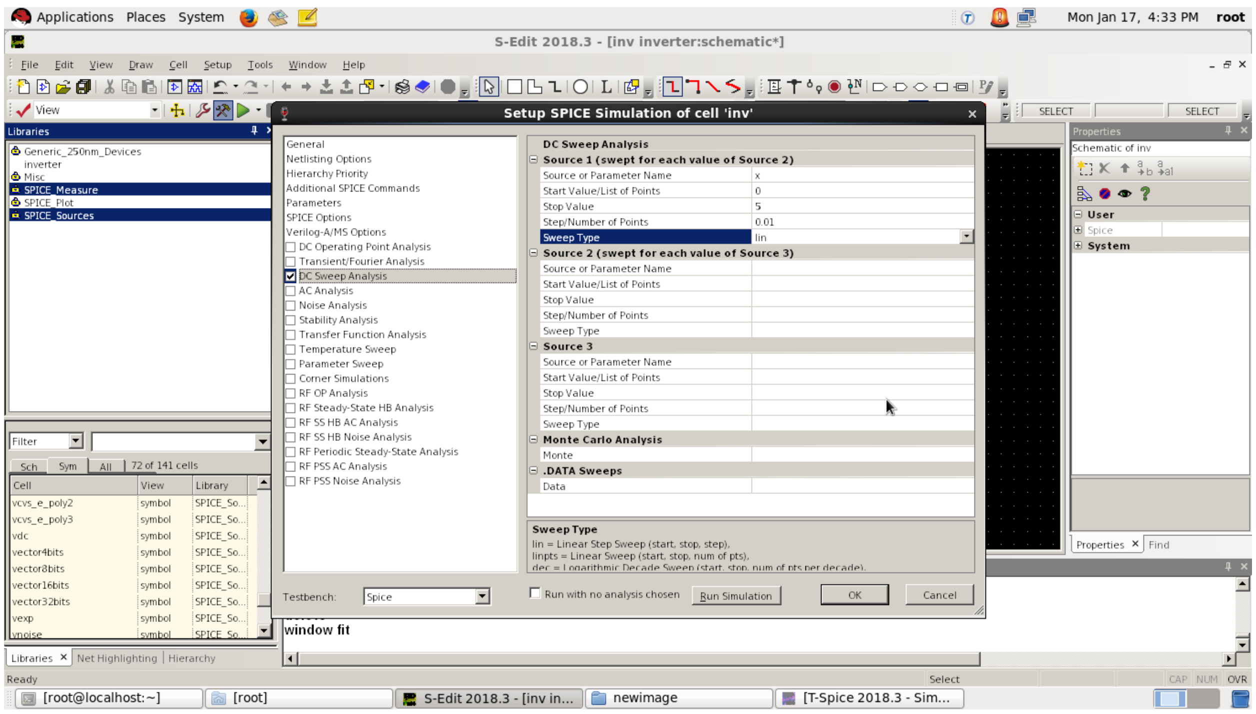
☑ Transient/Fourier Analysis

Testbench: Spice

Enable	true
PrintStart	0ns
PrintStep	1ns
Step	1ns
Stop	1000ns

❖ **Select DC Sweep Analysis** → click on the square to the left of DC Sweep Analysis **(de-select Transient / Fourier Analysis)** and give following parameters.

Source or Parameter Name	**x**
Start value / List of Points	**0**
Stop Value	**5**
Step/ No of Points	**0.01**
Sweep Type	**lin**



Setup SPICE Simulation of cell 'inv'

DC Sweep Analysis

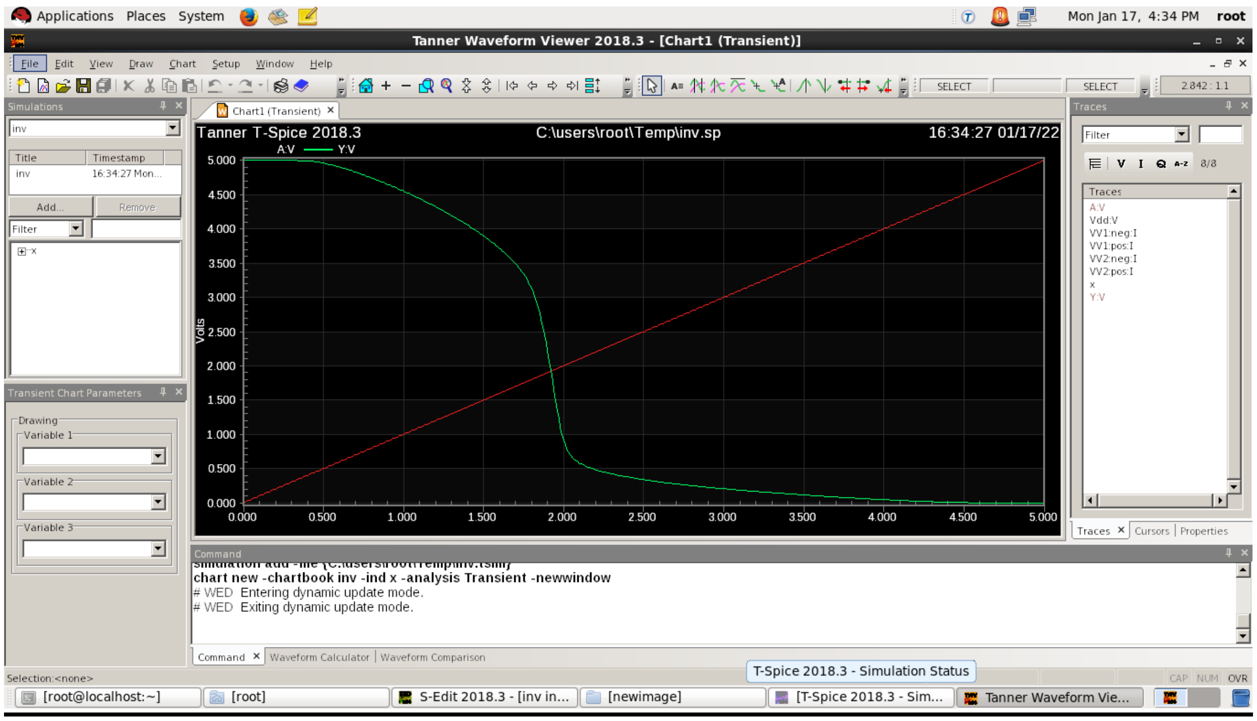
Source 1 (swept for each value of Source 2)

Source or Parameter Name	x
Start Value/List of Points	0
Stop Value	5
Step/Number of Points	0.01
Sweep Type	lin

Sweep Type
lin = Linear Step Sweep (start, stop, step),
linpts = Linear Sweep (start, stop, num of pts),
dec = Logarithmic Decade Sweep (start, stop, num of pts per decade)

Testbench: Spice

- ❖ Click **OK.**
- ❖ Click on **Start Simulation**, waveform window will be open.

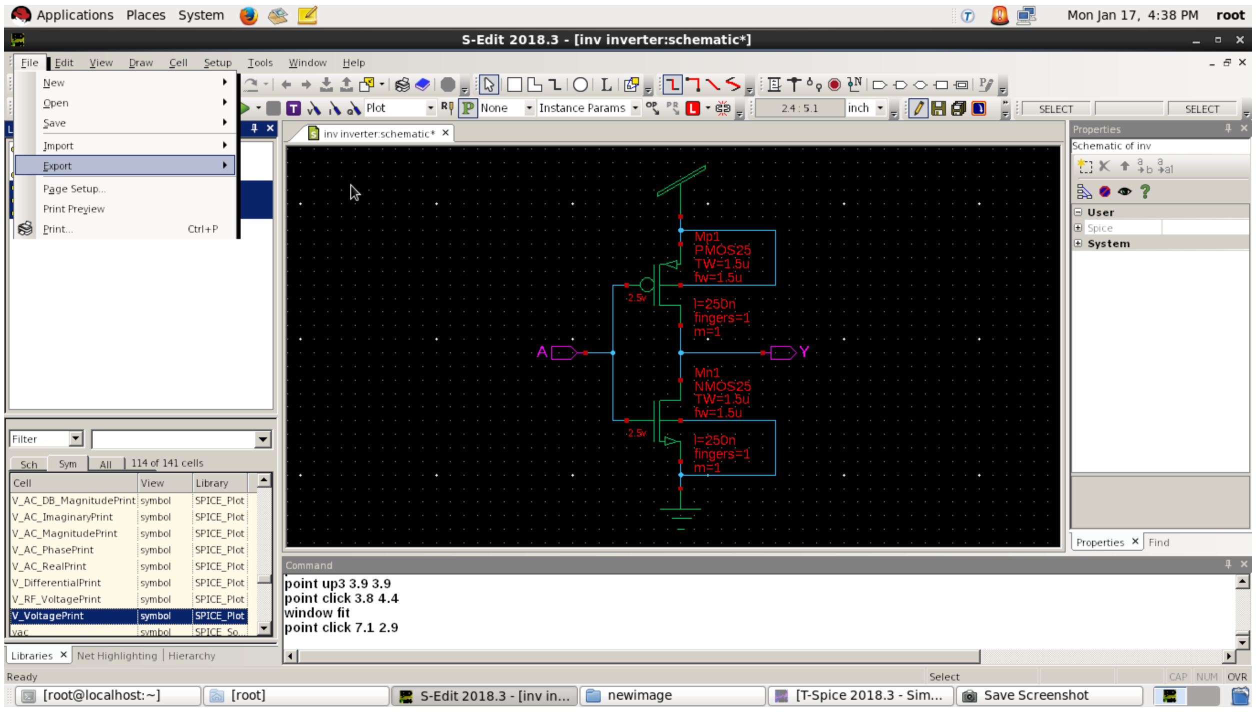


```
chart new -chartbook inv -ind x -analysis Transient -newwindow
# WED  Entering dynamic update mode.
# WED  Exiting dynamic update mode.
```

Layout Design:

- Go to **setup** → **SPICE Simulation**
- Select **General**

  Remove the library files → **OK**
- Go to S-Edit window (schematic circuit), **File** → **Export** → **Export SPICE**

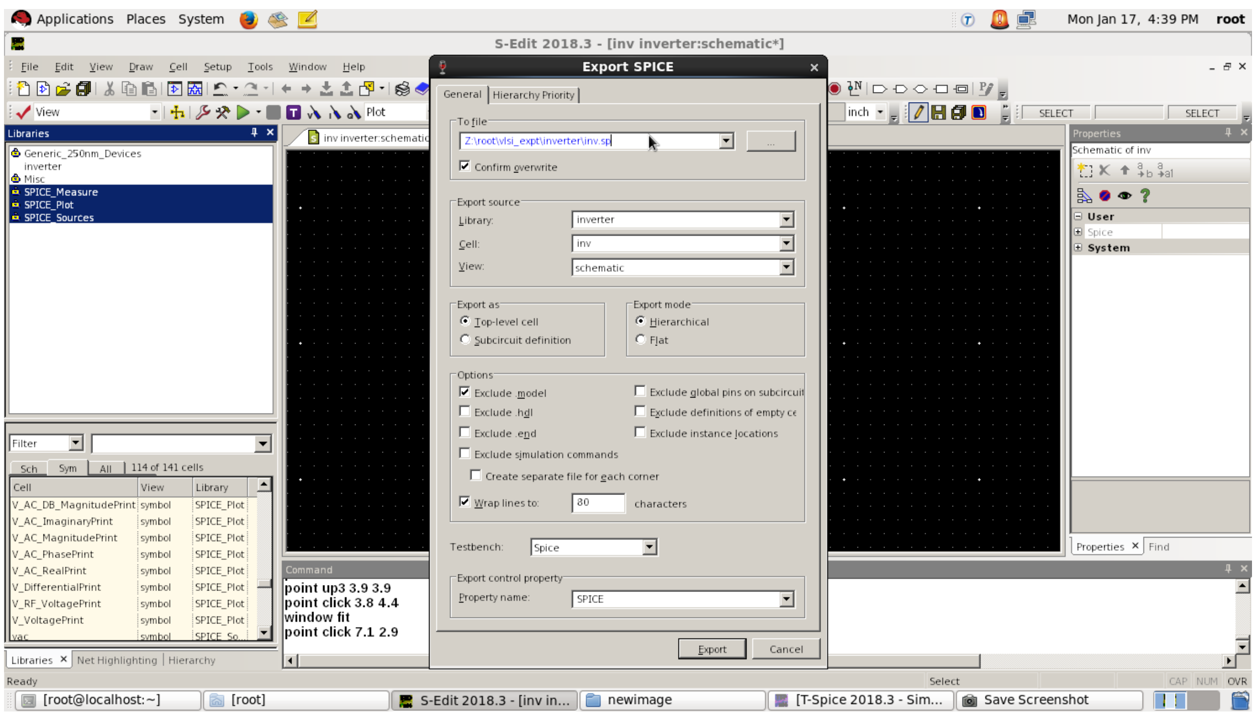


Cell	View	Library
V_AC_DB_MagnitudePrint	symbol	SPICE_Plot
V_AC_ImaginaryPrint	symbol	SPICE_Plot
V_AC_MagnitudePrint	symbol	SPICE_Plot
V_AC_PhasePrint	symbol	SPICE_Plot
V_AC_RealPrint	symbol	SPICE_Plot
V_DifferentialPrint	symbol	SPICE_Plot
V_RF_VoltagePrint	symbol	SPICE_Plot
V_VoltagePrint	symbol	SPICE_Plot
vac	symbol	SPICE_So

```
point up3 3.9 3.9
point click 3.8 4.4
window fit
point click 7.1 2.9
```

- **Browse your folder** → **save (**don't change any other parameters**)** → **Export**

  (File format should be **.sp** file)



Export SPICE

To file: Z:\root\vlsi_expt\inverter\inv.sp

[✓] Confirm overwrite

Export source
Library: inverter
Cell: inv
View: schematic

Export as: [✓ Top-level cell] Subcircuit definition

Export mode: [✓ Hierarchical] Flat

Options
[✓] Exclude model
[ ] Exclude .hdl
[ ] Exclude .end
[ ] Exclude simulation commands
[ ] Create separate file for each corner
[✓] Wrap lines to: 80 characters
[ ] Exclude global pins on subcircuit
[ ] Exclude definitions of empty ce
[ ] Exclude instance locations

Testbench: Spice

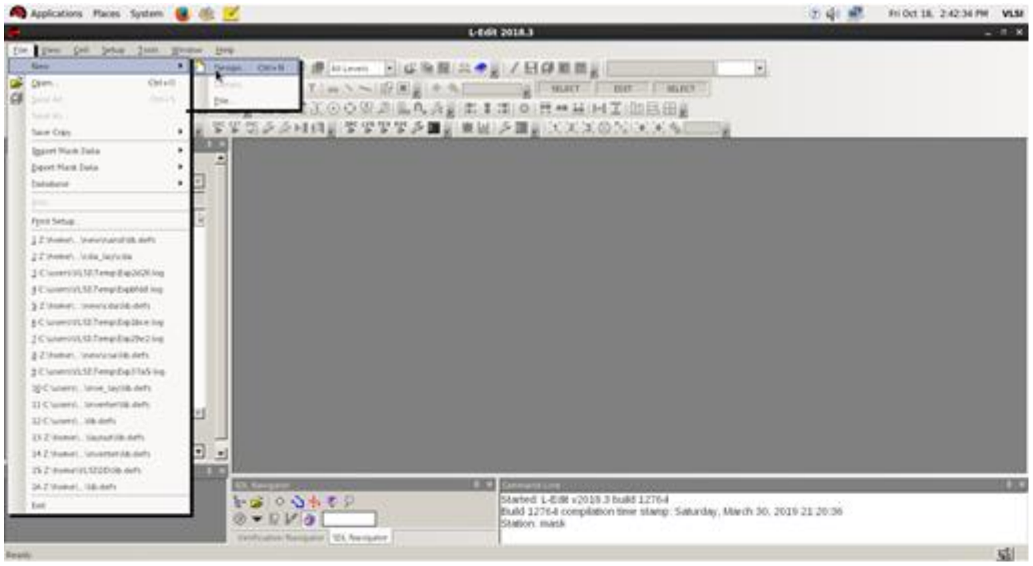
Export control property
Property name: SPICE

Cell	View	Library
V_AC_DB_MagnitudePrint	symbol	SPICE_Plot
V_AC_ImaginaryPrint	symbol	SPICE_Plot
V_AC_MagnitudePrint	symbol	SPICE_Plot
V_AC_PhasePrint	symbol	SPICE_Plot
V_AC_RealPrint	symbol	SPICE_Plot
V_DifferentialPrint	symbol	SPICE_Plot
V_RF_VoltagePrint	symbol	SPICE_Plot
V_VoltagePrint	symbol	SPICE_Plot
vac	symbol	SPICE_So

114 of 141 cells

Command
```
point up3 3.9 3.9
point click 3.8 4.4
window fit
point click 7.1 2.9
```

- ❖ Close S-Edit window
- ❖ Type ledit in **open in Terminal** window
- ❖ **L-Edit** window will open, minimize **open in terminal** window
- ❖ In L-Edit window, go to **File → New → Design**


Applications Places System
L-Edit 2018.3
File View Cell Setup Tools Window Help
New
Design
Open...
Ctrl+O
Exit
Ready

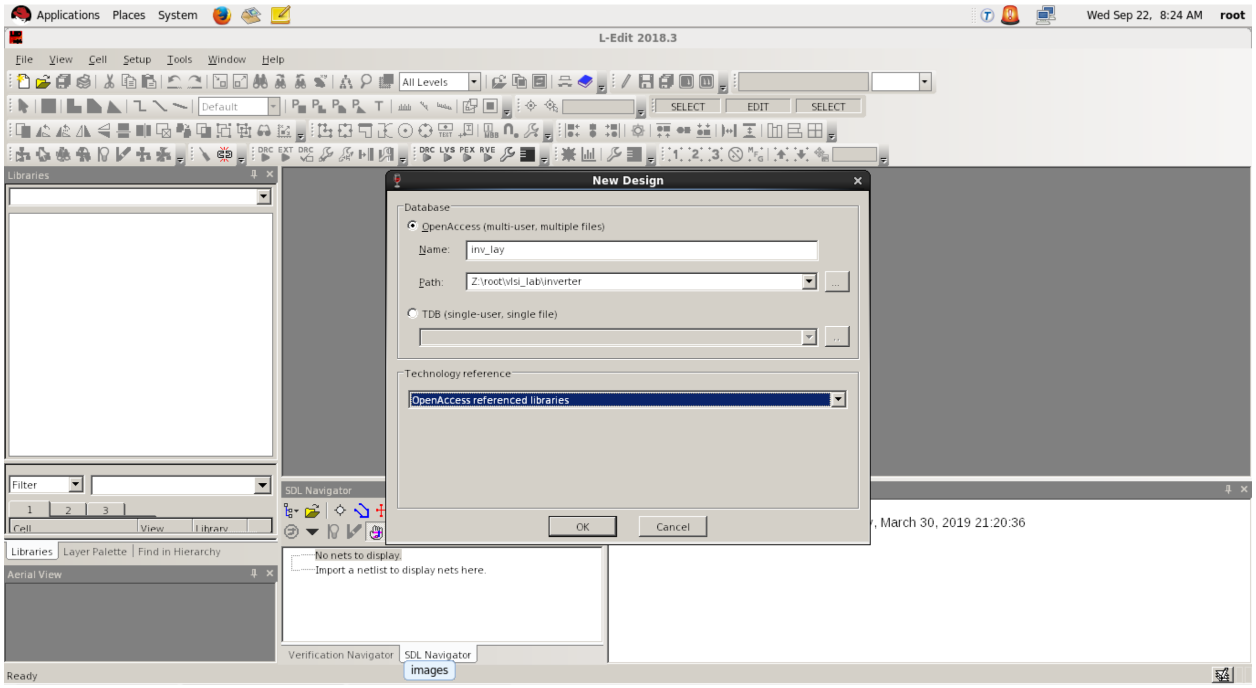
```
Started: L-Edit v2018.3 build 12764
Build 12764 compilation time stamp: Saturday, March 30, 2019 21:20:36
Station: mask
```

- ❖ In New Design window give the following parameters

  Name → **inv_lay (for example)**

  Path → browse your folder

  Technology reference → OpenAcess referenced libraries → ok

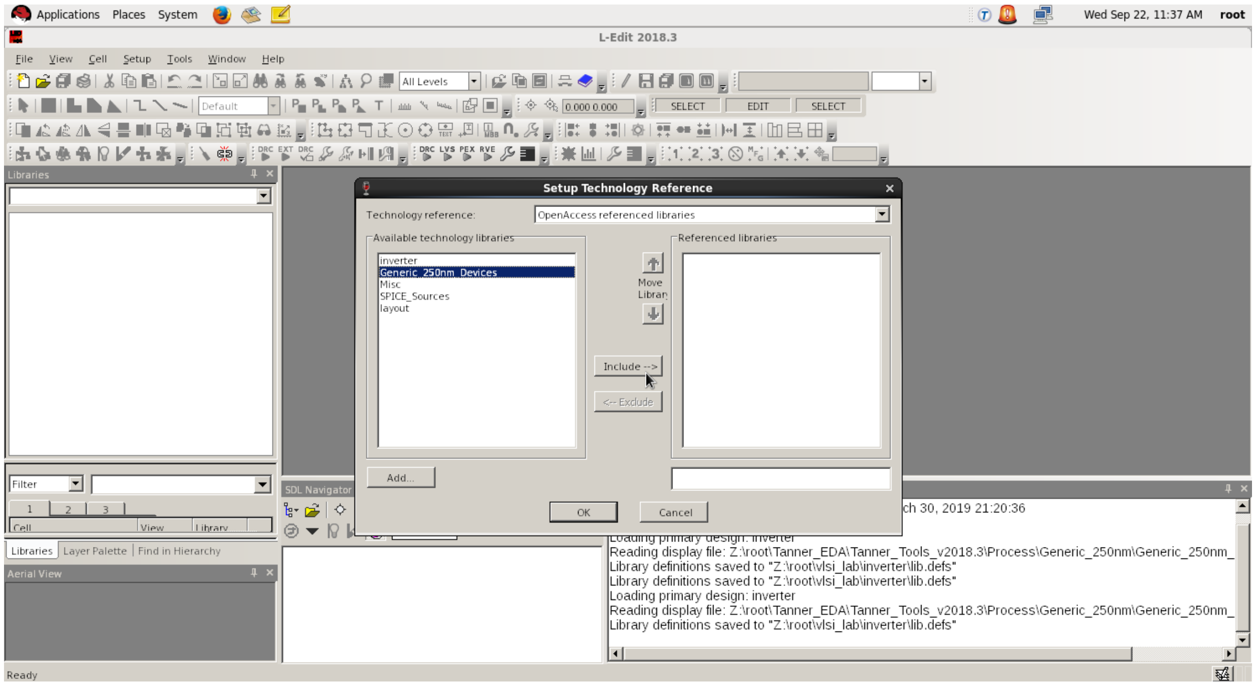


Database
(●) OpenAccess (multi-user, multiple files)
Name: inv_lay
Path: Z:\root\vlsi_lab\inverter
( ) TDB (single-user, single file)
Technology reference: OpenAccess referenced libraries

No nets to display
Import a netlist to display nets here.

, March 30, 2019 21:20:36

❖ Setup Technology Reference window will open, select **Generic_250nm_Devices → include → OK**


Applications Places System
Wed Sep 22, 11:37 AM root
L-Edit 2018.3
File View Cell Setup Tools Window Help
Setup Technology Reference
Move Library
Include -->
<-- Exclude
Add...
OK
Cancel
Ready

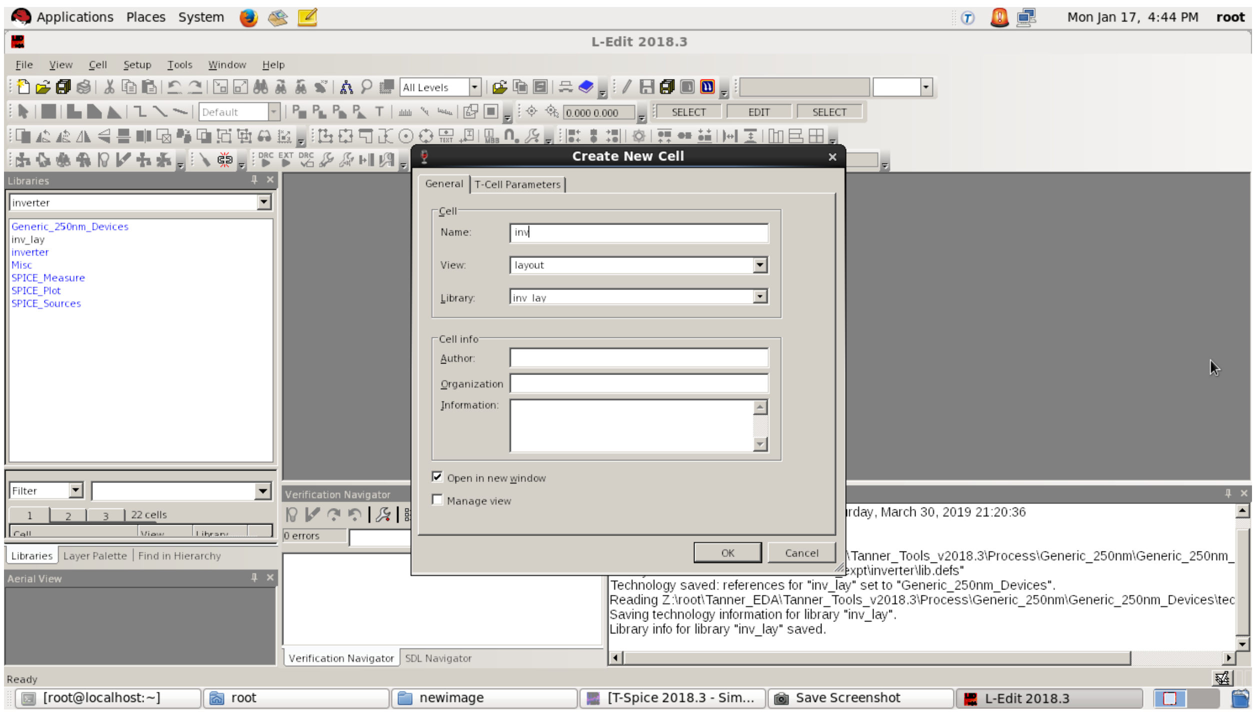
Technology reference: OpenAccess referenced libraries
Available technology libraries: inverter, Generic_250nm_Devices, Misc, SPICE_Sources, layout
Referenced libraries:

```
ch 30, 2019 21:20:36
Loading primary design: inverter
Reading display file: Z:\root\Tanner_EDA\Tanner_Tools_v2018.3\Process\Generic_250nm\Generic_250nm_
Library definitions saved to "Z:\root\vlsi_lab\inverter\lib.defs"
Library definitions saved to "Z:\root\vlsi_lab\inverter\lib.defs"
Loading primary design: inverter
Reading display file: Z:\root\Tanner_EDA\Tanner_Tools_v2018.3\Process\Generic_250nm\Generic_250nm_
Library definitions saved to "Z:\root\vlsi_lab\inverter\lib.defs"
```

❖ Go to **Cell → New,** give the following parameters

Cell name → **inv (for example) (**don’t change any other parameters**)**

(Provide cell name same as the schematic design cell name)


Applications Places System
Mon Jan 17, 4:44 PM root
L-Edit 2018.3
File View Cell Setup Tools Window Help
Create New Cell
General T-Cell Parameters
OK Cancel
Libraries Layer Palette Find in Hierarchy
Verification Navigator SDL Navigator
[root@localhost:~] root newimage [T-Spice 2018.3 - Sim... Save Screenshot L-Edit 2018.3

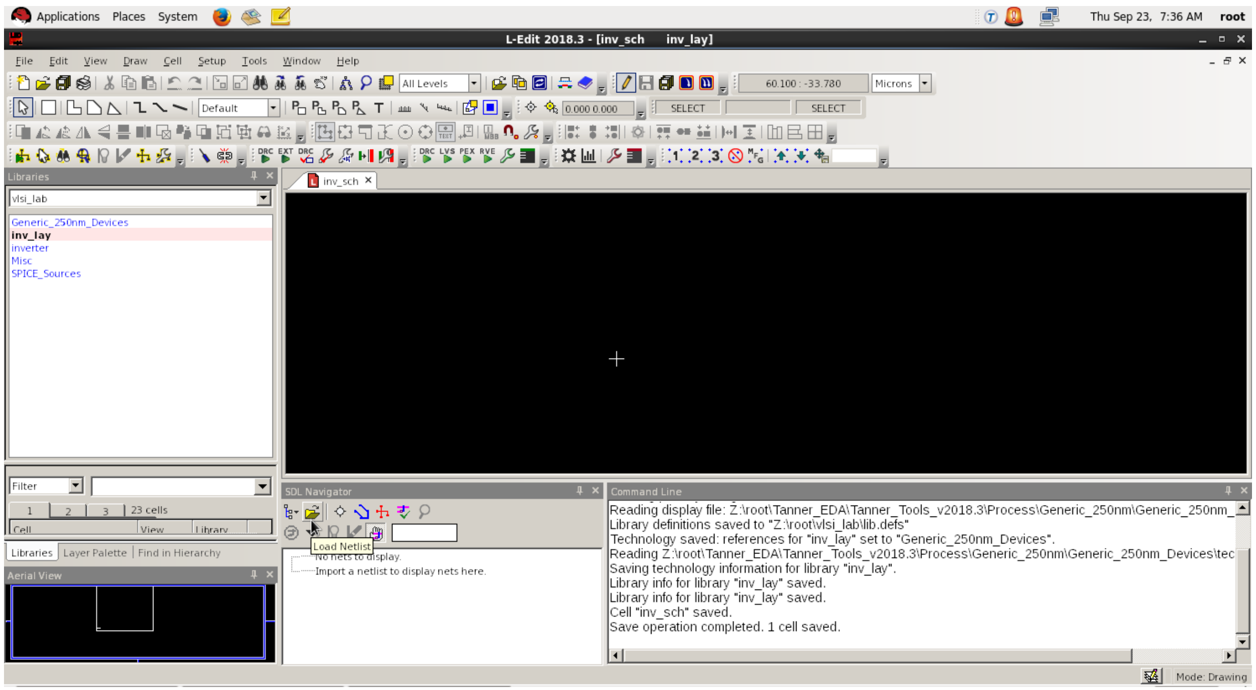
Cell
Name: inv
View: layout
Library: inv_lay

Cell info
Author:
Organization:
Information:

☑ Open in new window
☐ Manage view

```
rday, March 30, 2019 21:20:36
\Tanner_Tools_v2018.3\Process\Generic_250nm\Generic_250nm_
xpt\inverter\lib.defs"
Technology saved: references for "inv_lay" set to "Generic_250nm_Devices".
Reading Z:\root\Tanner_EDA\Tanner_Tools_v2018.3\Process\Generic_250nm\Generic_250nm_Devices\tec
Saving technology information for library "inv_lay".
Library info for library "inv_lay" saved.
```

- ❖ In SDL navigator window, click on **Load Netlist**

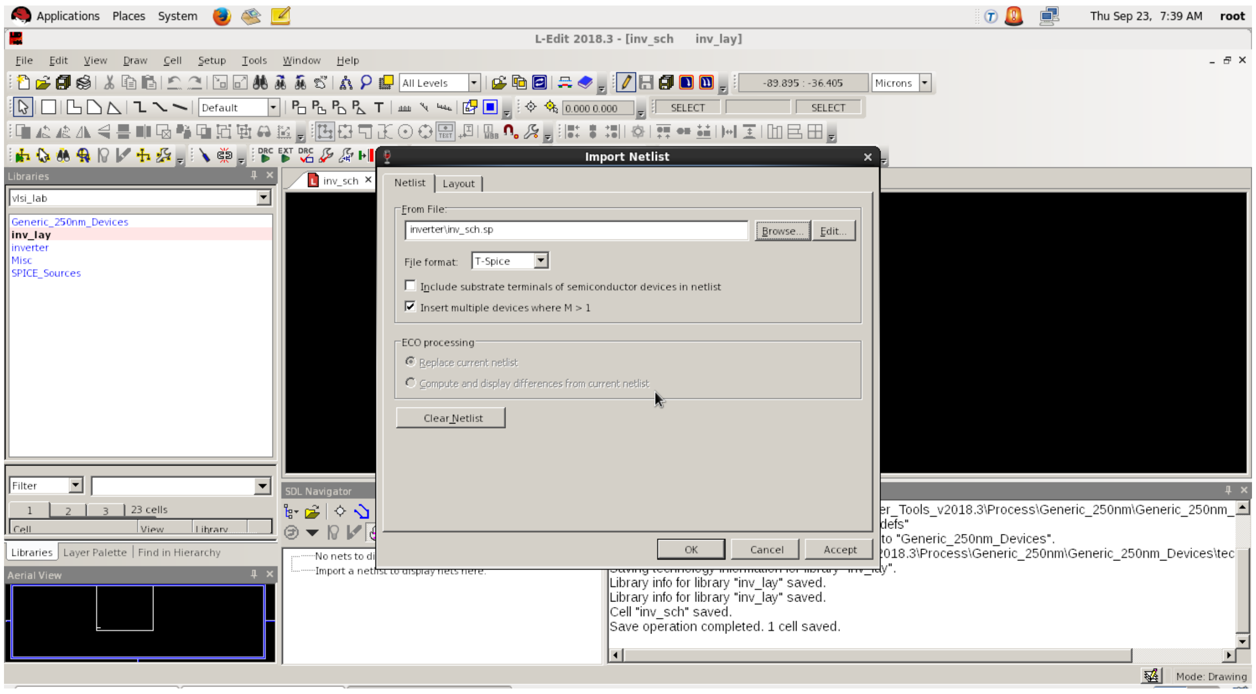


```
No nets to display.
Import a netlist to display nets here.
```

```
Reading display file: Z:\root\Tanner_EDA\Tanner_Tools_v2018.3\Process\Generic_250nm\Generic_250nm_
Library definitions saved to "Z:\root\vlsi_lab\lib.defs"
Technology saved: references for "inv_lay" set to "Generic_250nm_Devices".
Reading Z:\root\Tanner_EDA\Tanner_Tools_v2018.3\Process\Generic_250nm\Generic_250nm_Devices\tec
Saving technology information for library "inv_lay".
Library info for library "inv_lay" saved.
Library info for library "inv_lay" saved.
Cell "inv_sch" saved.
Save operation completed. 1 cell saved.
```

- ❖ Go to **Netlist** tab give the following parameters
  - ➢ Select **.sp** file from your folder
  - ➢ File format → T-Spice
- ❖ Go to **Layout** tab give the following parameters

- **Layer** → Metal1: drawing
- **Size** → 0.250 Microns → **OK.**



Import Netlist

Netlist | Layout

From File: inverter\inv_sch.sp Browse... Edit...

File format: T-Spice

☐ Include substrate terminals of semiconductor devices in netlist

☑ Insert multiple devices where M > 1

ECO processing

● Replace current netlist

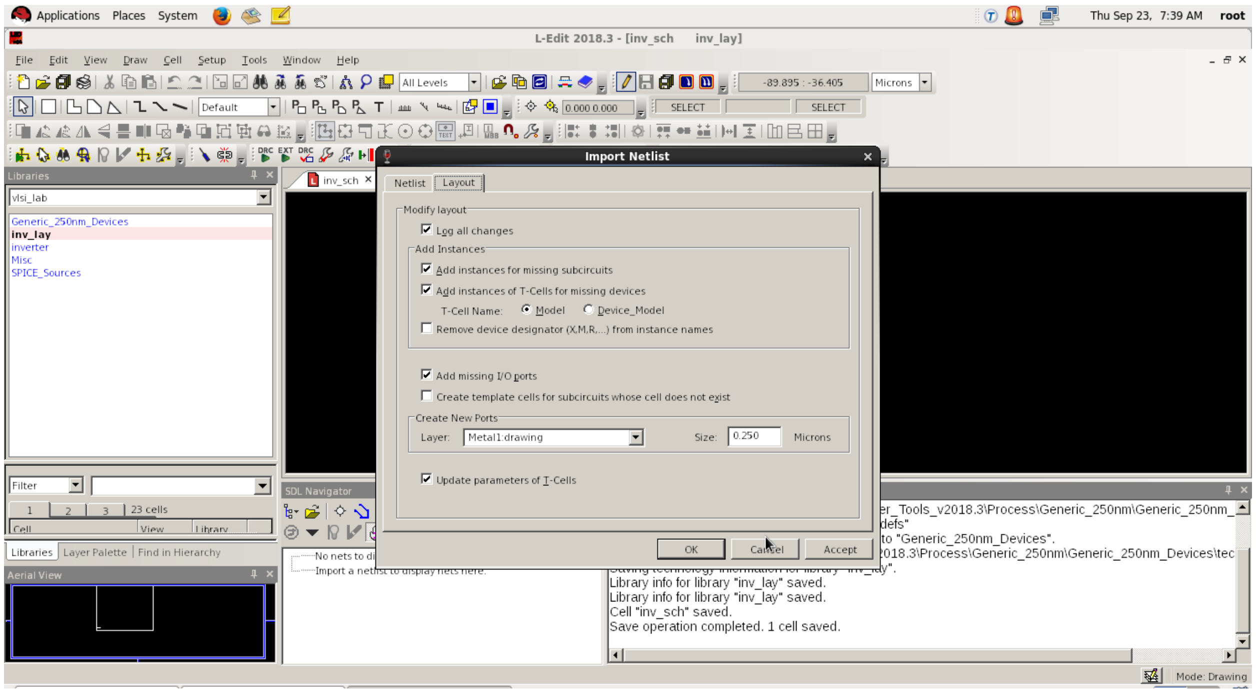
○ Compute and display differences from current netlist

Clear Netlist

OK Cancel Accept

Libraries: Generic_250nm_Devices, inv_lay, inverter, Misc, SPICE_Sources

```
...r_Tools_v2018.3\Process\Generic_250nm\Generic_250nm_
defs"
to "Generic_250nm_Devices".
2018.3\Process\Generic_250nm\Generic_250nm_Devices\tec
...ay".
Library info for library "inv_lay" saved.
Library info for library "inv_lay" saved.
Cell "inv_sch" saved.
Save operation completed. 1 cell saved.
```



Import Netlist

Netlist | Layout

Modify layout

☑ Log all changes

Add Instances

☑ Add instances for missing subcircuits

☑ Add instances of T-Cells for missing devices

T-Cell Name: ● Model ○ Device_Model

☐ Remove device designator (X,M,R,...) from instance names

☑ Add missing I/O ports

☐ Create template cells for subcircuits whose cell does not exist

Create New Ports

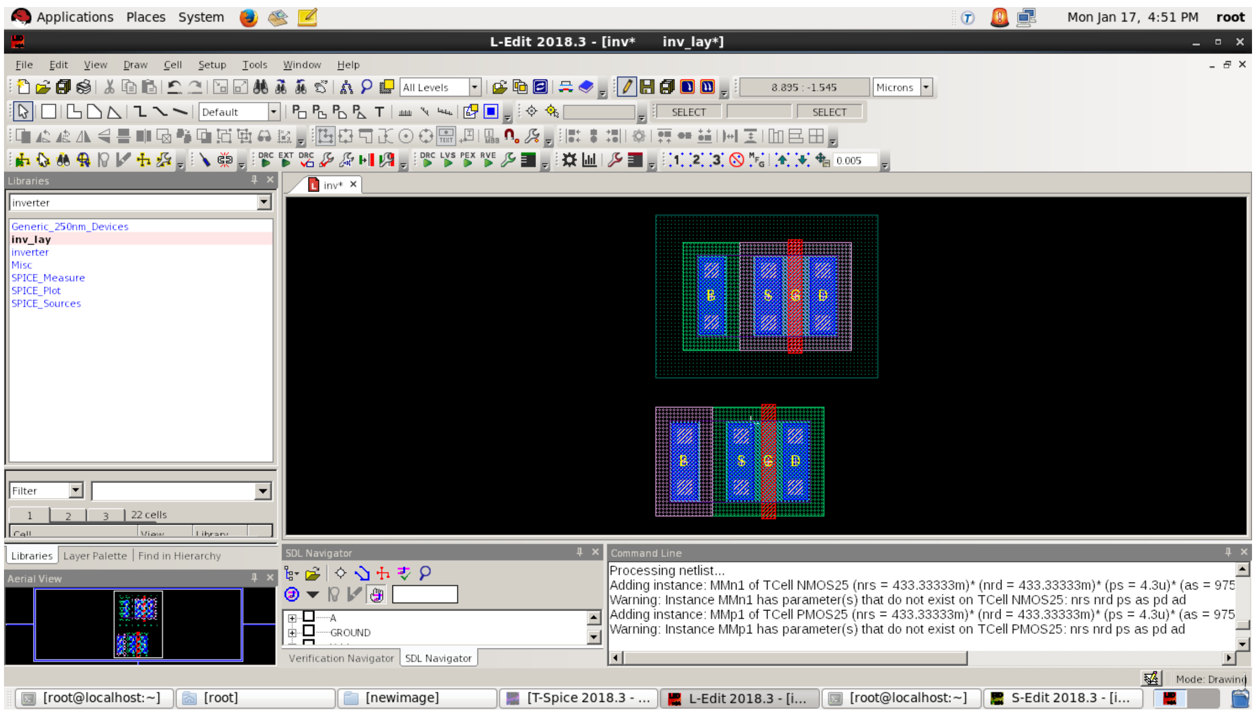
Layer: Metal1:drawing Size: 0.250 Microns

☑ Update parameters of T-Cells

OK Cancel Accept

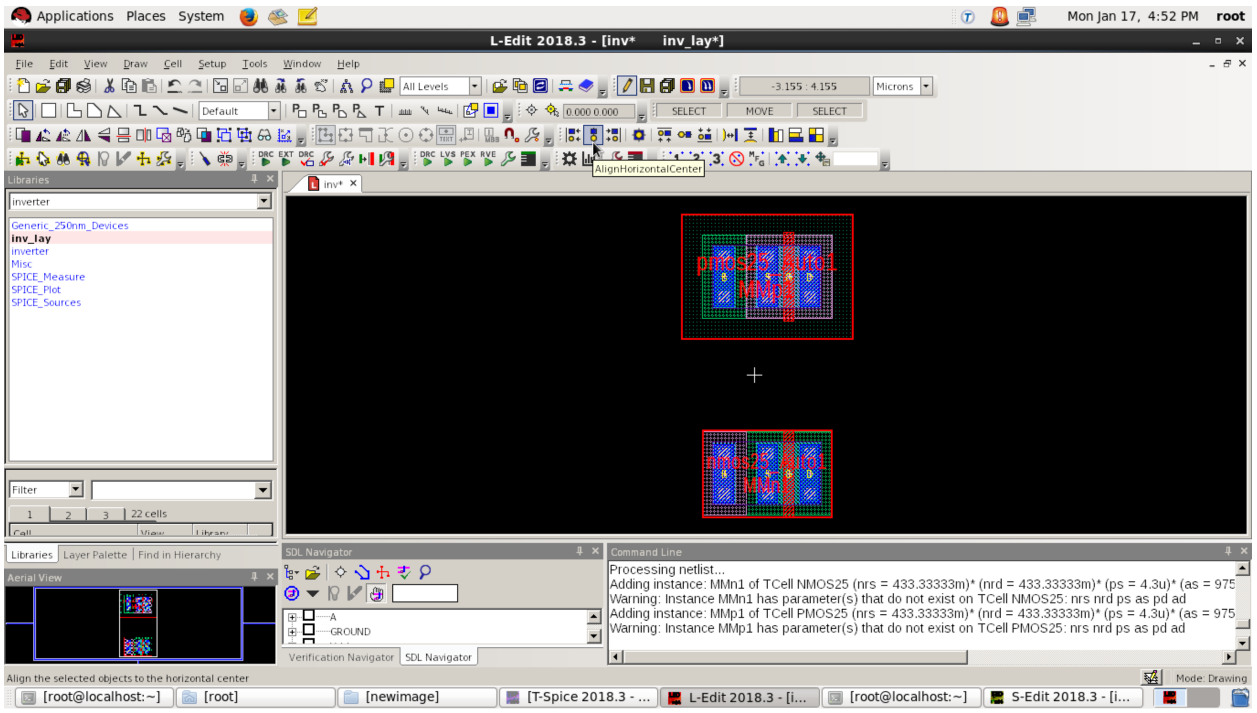
```
Library info for library "inv_lay" saved.
Library info for library "inv_lay" saved.
Cell "inv_sch" saved.
Save operation completed. 1 cell saved.
```

- Layout of pMOS and nMOS transistor are placed on layout editor window



```
Processing netlist...
Adding instance: MMn1 of TCell NMOS25 (nrs = 433.33333m)* (nrd = 433.33333m)* (ps = 4.3u)* (as = 975
Warning: Instance MMn1 has parameter(s) that do not exist on TCell NMOS25: nrs nrd ps as pd ad
Adding instance: MMp1 of TCell PMOS25 (nrs = 433.33333m)* (nrd = 433.33333m)* (ps = 4.3u)* (as = 975
Warning: Instance MMp1 has parameter(s) that do not exist on TCell PMOS25: nrs nrd ps as pd ad
```

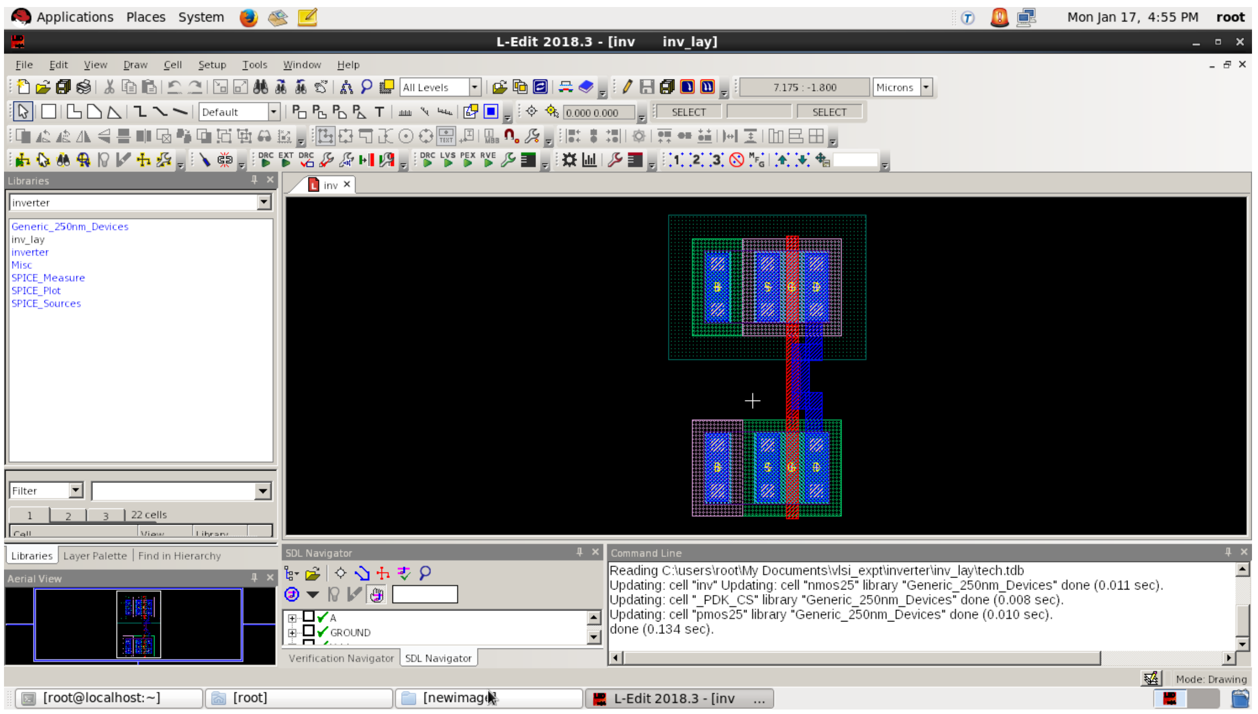
➢ Select both devices and click on **Align Horizontal Centre** icon



```
Processing netlist...
Adding instance: MMn1 of TCell NMOS25 (nrs = 433.33333m)* (nrd = 433.33333m)* (ps = 4.3u)* (as = 975
Warning: Instance MMn1 has parameter(s) that do not exist on TCell NMOS25: nrs nrd ps as pd ad
Adding instance: MMp1 of TCell PMOS25 (nrs = 433.33333m)* (nrd = 433.33333m)* (ps = 4.3u)* (as = 975
Warning: Instance MMp1 has parameter(s) that do not exist on TCell PMOS25: nrs nrd ps as pd ad
```

Align the selected objects to the horizontal center

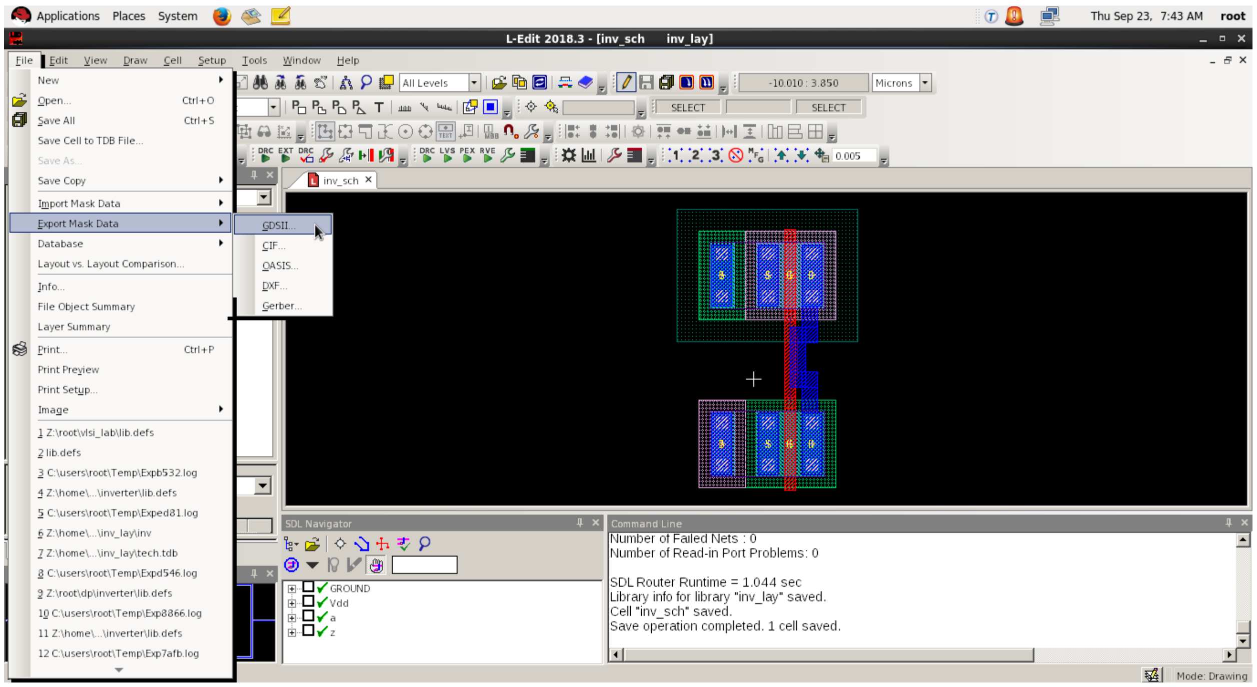
➢ In SDL navigator window, click on **Route All** icon



```
Reading C:\users\root\My Documents\vlsi_expt\inverter\inv_lay\tech.tdb
Updating: cell "inv" Updating: cell "nmos25" library "Generic_250nm_Devices" done (0.011 sec).
Updating: cell "_PDK_CS" library "Generic_250nm_Devices" done (0.008 sec).
Updating: cell "pmos25" library "Generic_250nm_Devices" done (0.010 sec).
done (0.134 sec).
```

❖ Save the layout design

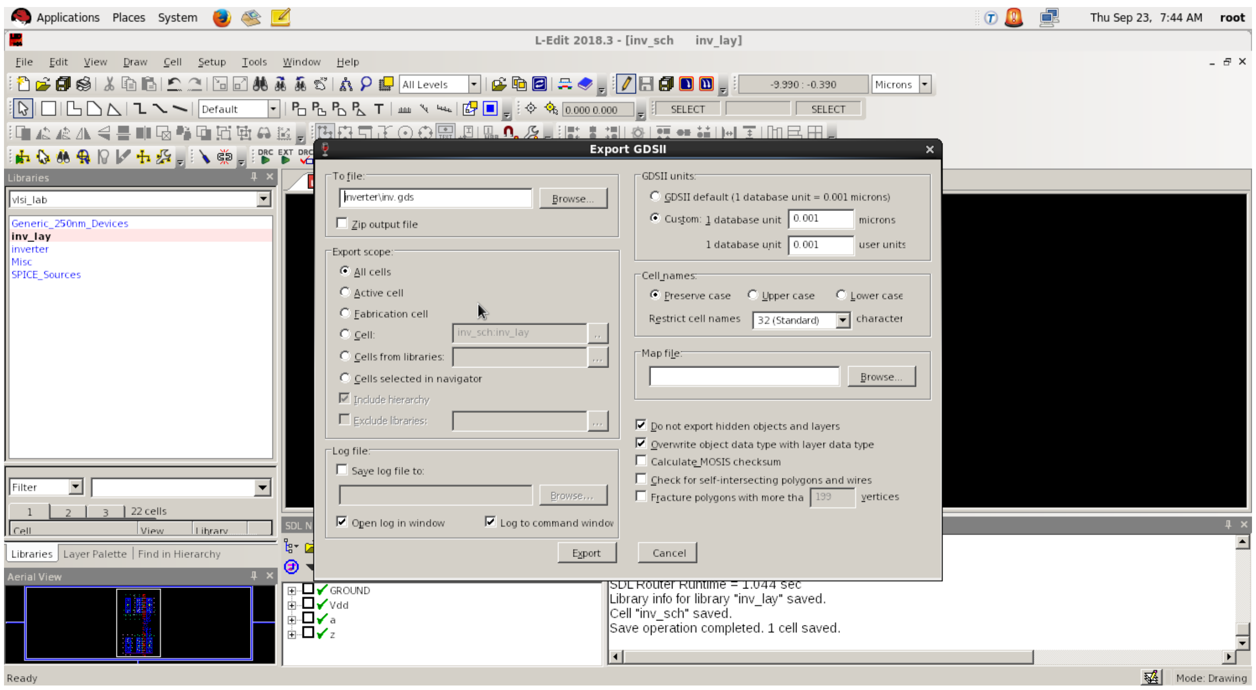
❖ Go to **File → Export Mask Data → GDSII**



```
Number of Failed Nets : 0
Number of Read-in Port Problems: 0

SDL Router Runtime = 1.044 sec
Library info for library "inv_lay" saved.
Cell "inv_sch" saved.
Save operation completed. 1 cell saved.
```

❖ **Save file as inv.gds (for example)** in your folder **(**don't change any other parameters**)**

❖ Export



Export GDSII

To file: inverter\inv.gds Browse...
☐ Zip output file

Export scope:
[✓ All cells]
Active cell
Fabrication cell
Cell: inv_sch.inv_lay
Cells from libraries:
Cells selected in navigator
☑ Include hierarchy
☐ Exclude libraries:

Log file:
☐ Save log file to:
☑ Open log in window ☑ Log to command window

GDSII units:
GDSII default (1 database unit = 0.001 microns)
[✓ Custom: 1 database unit 0.001 microns]
1 database unit 0.001 user units

Cell names: [✓ Preserve case] Upper case Lower case
Restrict cell names 32 (Standard) character

Map file: Browse...

☑ Do not export hidden objects and layers
☑ Overwrite object data type with layer data type
☐ Calculate MOSIS checksum
☐ Check for self-intersecting polygons and wires
☐ Fracture polygons with more tha 199 vertices

Export Cancel

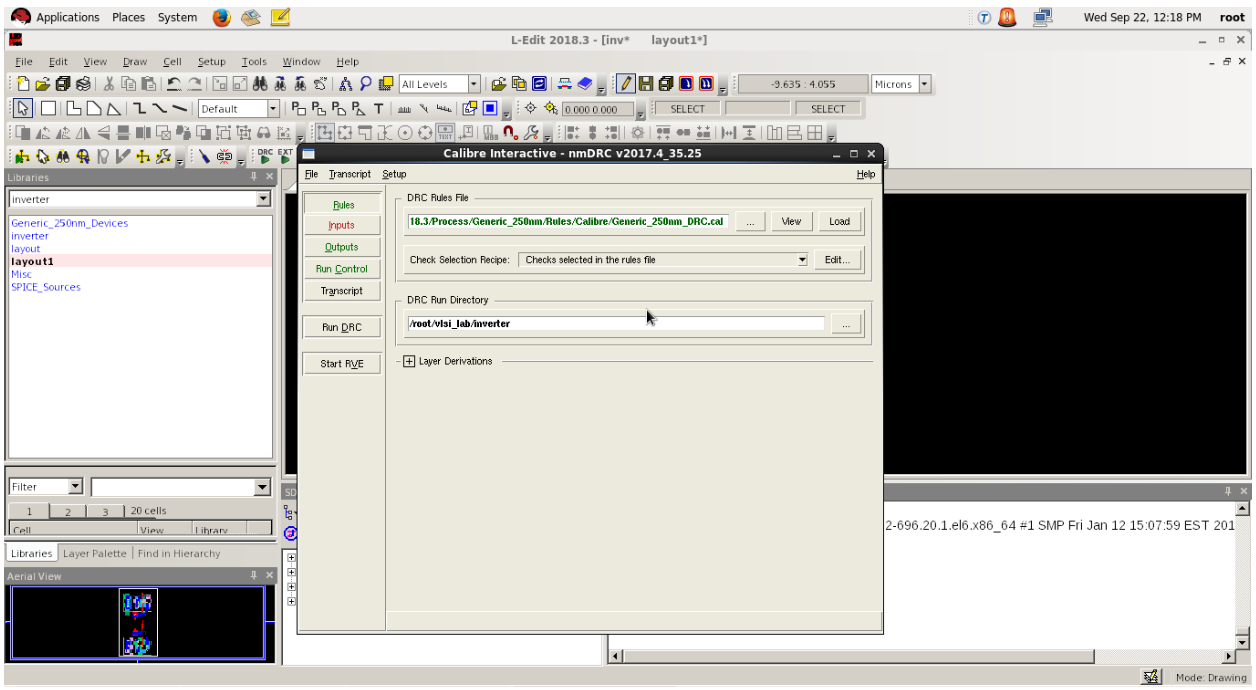
```
SDL Router Runtime = 1.044 sec
Library info for library "inv_lay" saved.
Cell "inv_sch" saved.
Save operation completed. 1 cell saved.
```

DRC (Design Rule Check):

❖ In the layout window, click on **Run Calibre DRC** icon

➢ Click on rules → browse rules file from library

root → Tanner_EDA → Tanner_Tools_v2018.3 → Process → Generic_250nm → Rules → calibre → Generic_250nm_DRC.cal → ok



Calibre Interactive - nmDRC v2017.4_35.25

DRC Rules File: 18.3/Process/Generic_250nm/Rules/Calibre/Generic_250nm_DRC.cal ... View Load

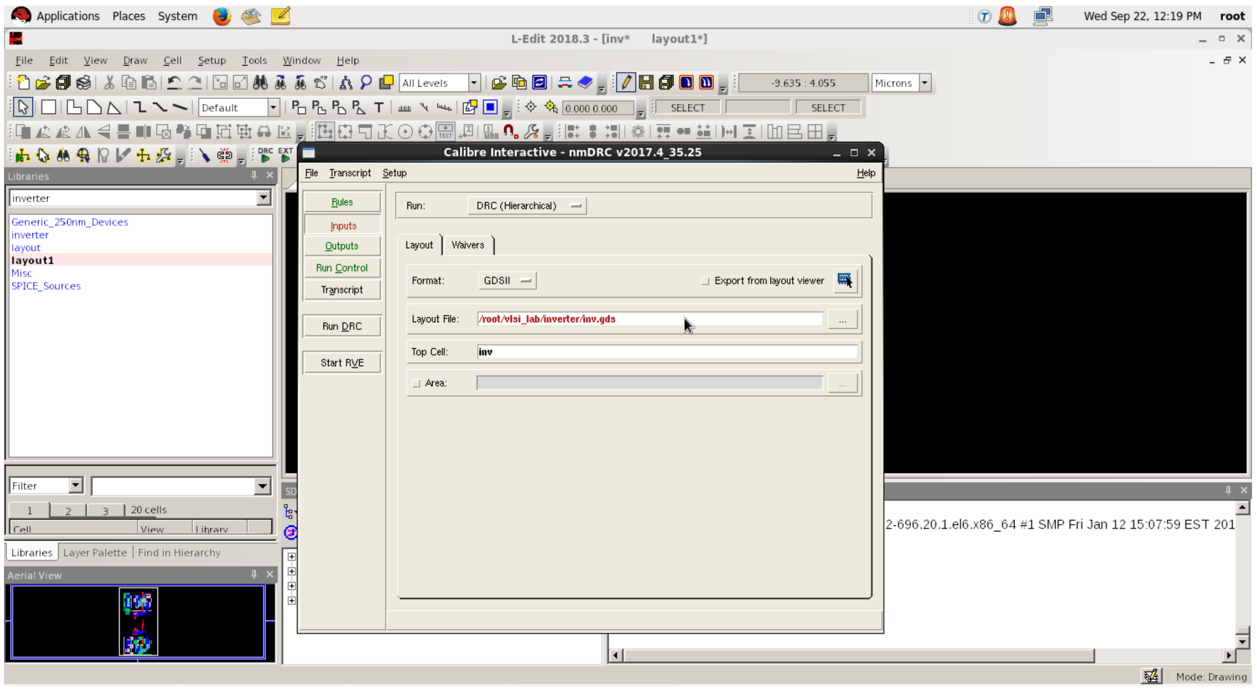
Check Selection Recipe: Checks selected in the rules file Edit...

DRC Run Directory: /root/vlsi_lab/inverter ...

Layer Derivations

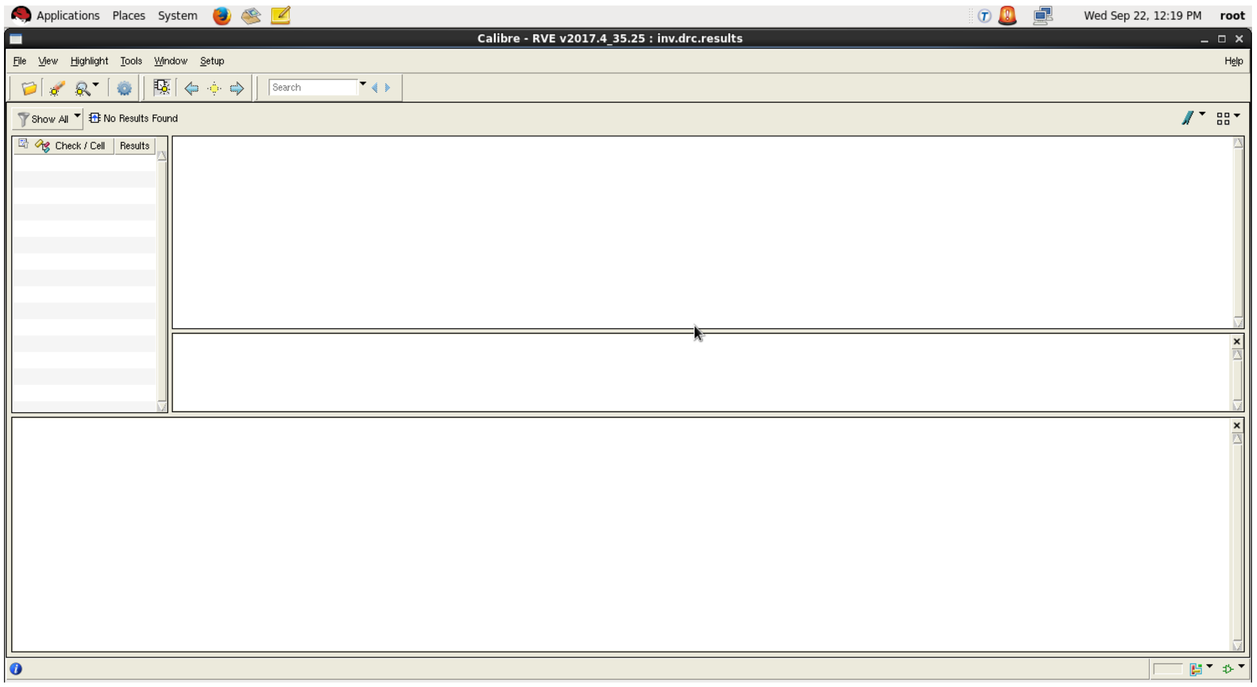
```
2-696.20.1.el6.x86_64 #1 SMP Fri Jan 12 15:07:59 EST 201
```

- **Click on Inputs** → de-select **export from layout viewer** → browse layout file from your folder, (**inv.gds** file) → **ok**
- **Click on Run DRC.**



Run: DRC (Hierarchical)
Format: GDSII
Export from layout viewer
Layout File: /root/vlsi_lab/inverter/inv.gds
Top Cell: inv
Area:

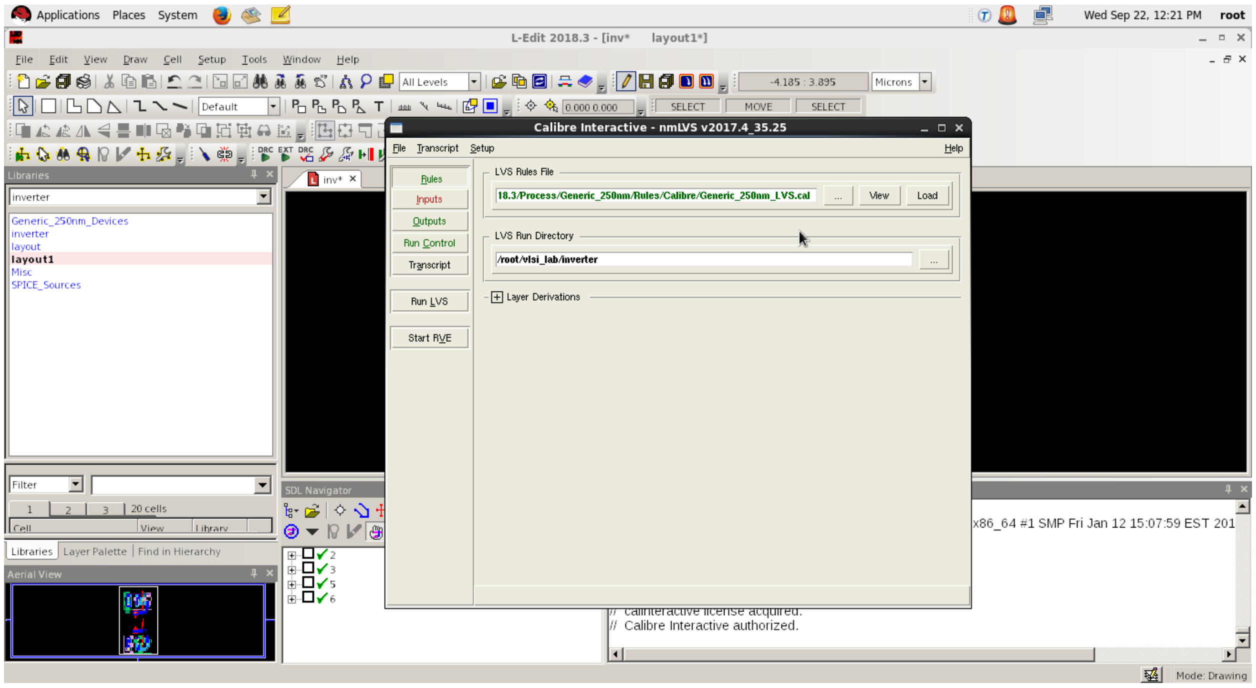
- DRC result window will open → if errors debug it.



LVS (Layout vs Schematic):

- ❖ In the layout window, click on **Run Calibre LVS** icon
    - ➢ Click on rules → browse rules file from library

root → Tanner_EDA → Tanner_Tools_v2018.3 → Process → Generic_250nm → Rules → calibre → Generic_250nm_LVS.cal → ok

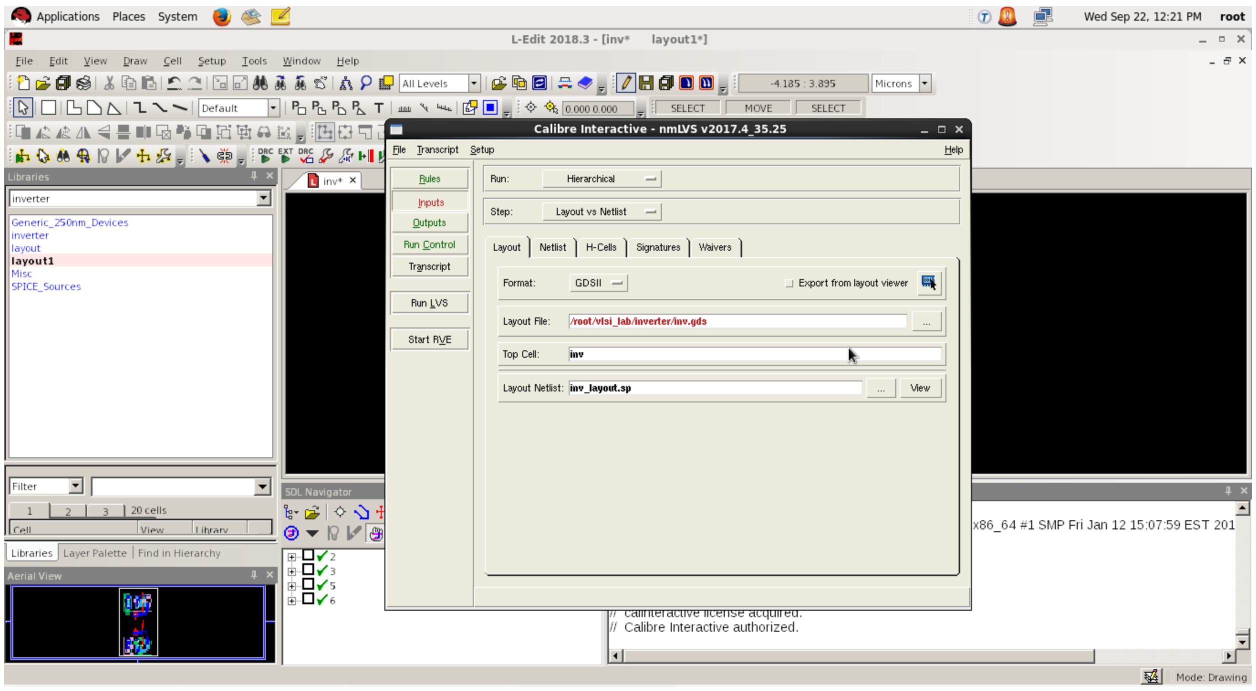


LVS Rules File: 18.3/Process/Generic_250nm/Rules/Calibre/Generic_250nm_LVS.cal

LVS Run Directory: /root/vlsi_lab/inverter

```
x86_64 #1 SMP Fri Jan 12 15:07:59 EST 201
// calinteractive license acquired.
// Calibre Interactive authorized.
```

- ➢ **Click on Inputs**
    - ▪ Go to Layout tab → de-select **export from layout viewer** → browse layout file from your folder, (**inv.gds** file)



Calibre Interactive - nmLVS v2017.4_35.25

Run: Hierarchical

Step: Layout vs Netlist

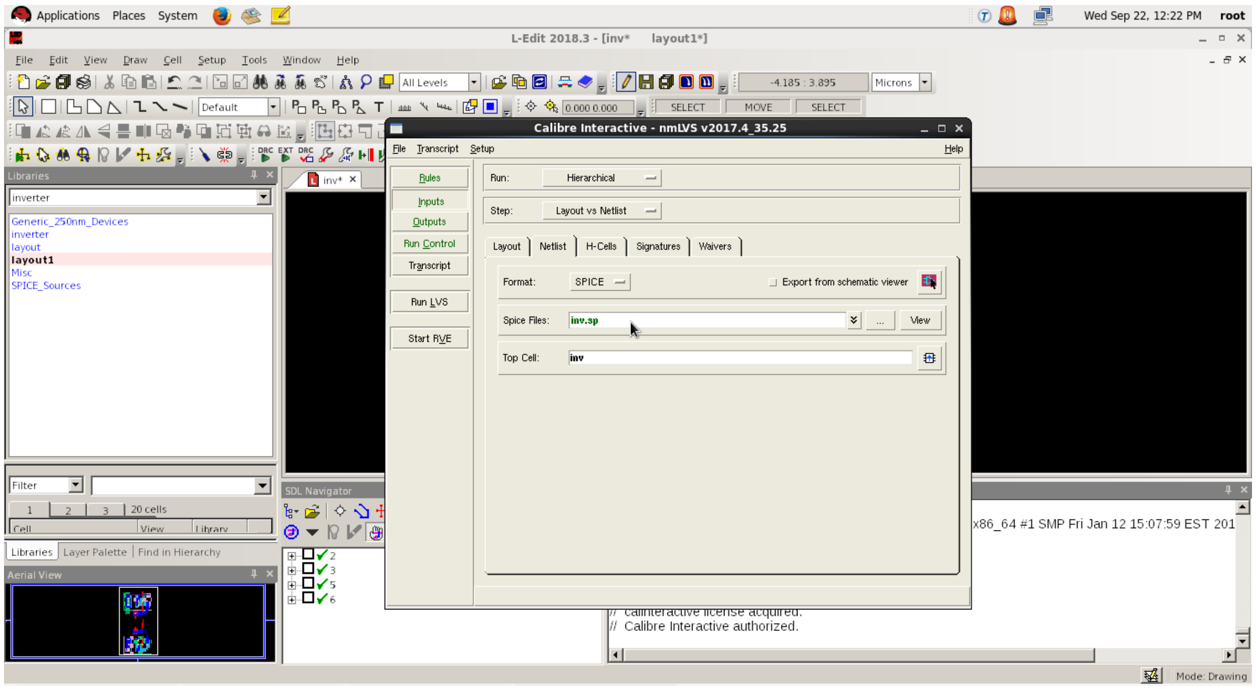
Format: GDSII

Layout File: /root/vlsi_lab/inverter/inv.gds

Top Cell: inv

Layout Netlist: inv_layout.sp

- Go to Netlist tab → Browse netlist file from your folder (.sp file) → **Add** → **OK**



Calibre Interactive - nmLVS v2017.4_35.25

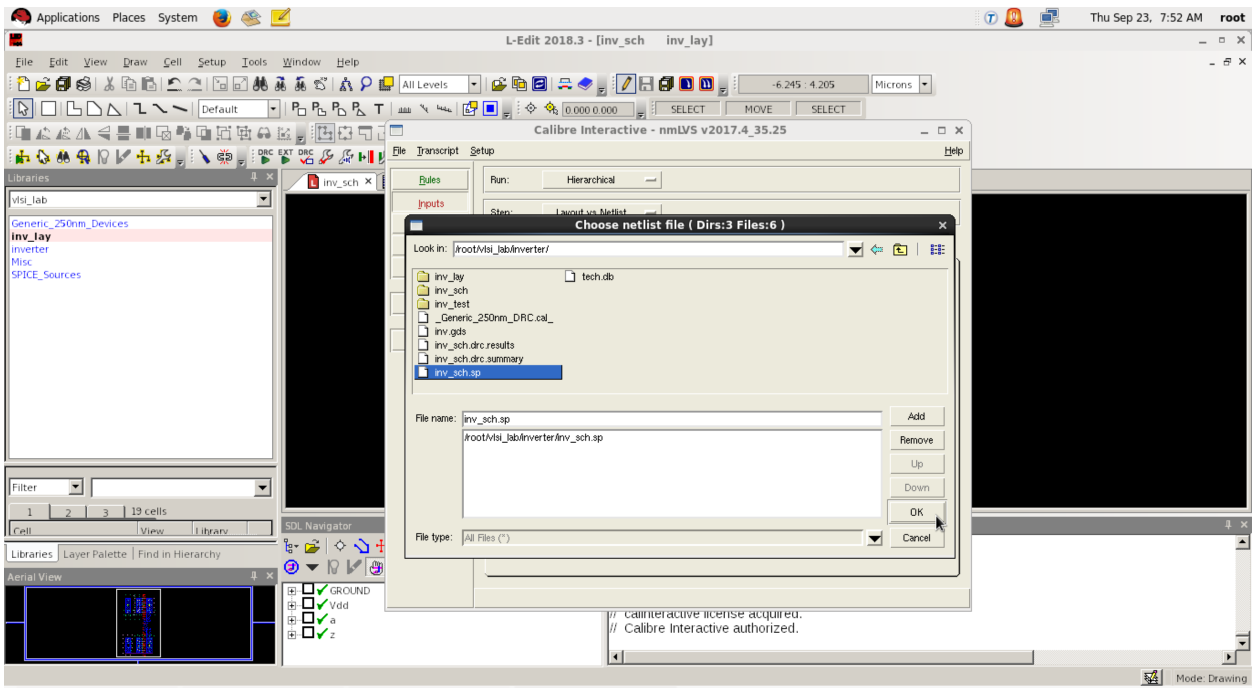
Run: Hierarchical

Step: Layout vs Netlist

Format: SPICE

Spice Files: inv.sp

Top Cell: inv



Choose netlist file (Dirs:3 Files:6)

Look in: /root/vlsi_lab/inverter/

inv_lay
inv_sch
inv_test
_Generic_250nm_DRC.cal_
inv.gds
inv_sch.drc.results
inv_sch.drc.summary
inv_sch.sp
tech.db

File name: inv_sch.sp

/root/vlsi_lab/inverter/inv_sch.sp

File type: All Files (*)

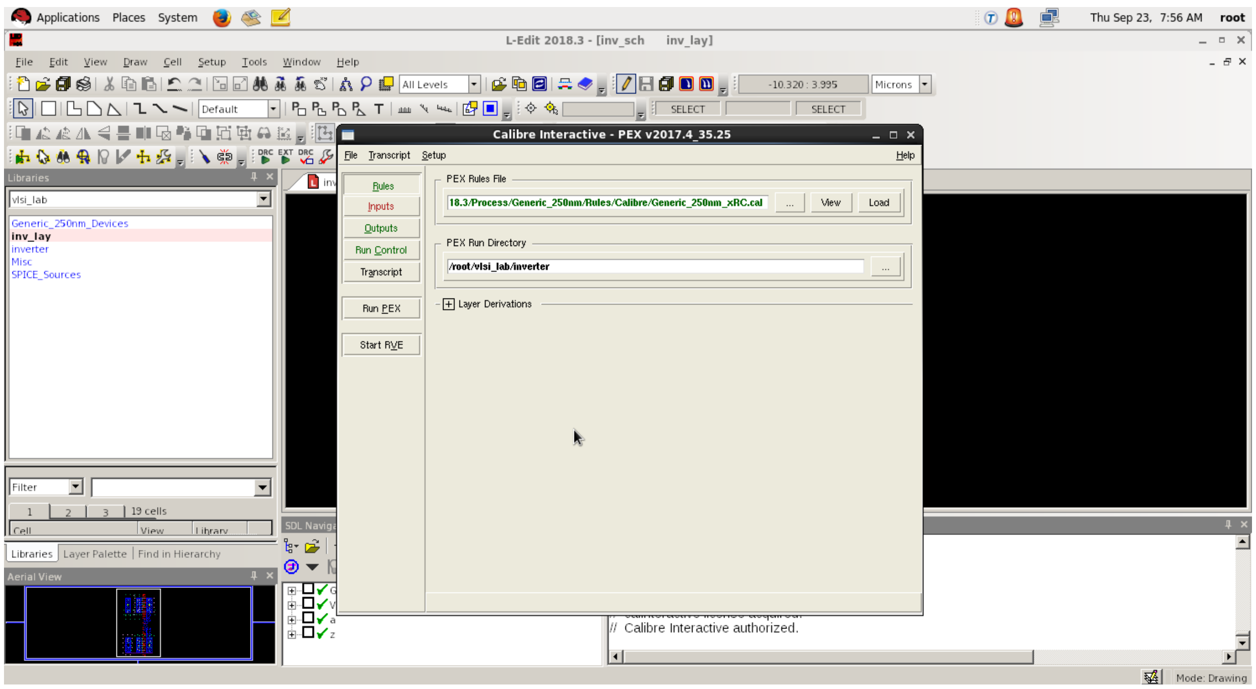
// calibreinteractive license acquired.
// Calibre Interactive authorized.

- **Click on Run LVS.**

PEX (Parasitic Extraction):

- In the layout window, click on Run Calibre PEX icon
    - Click on rules → browse rules file from library

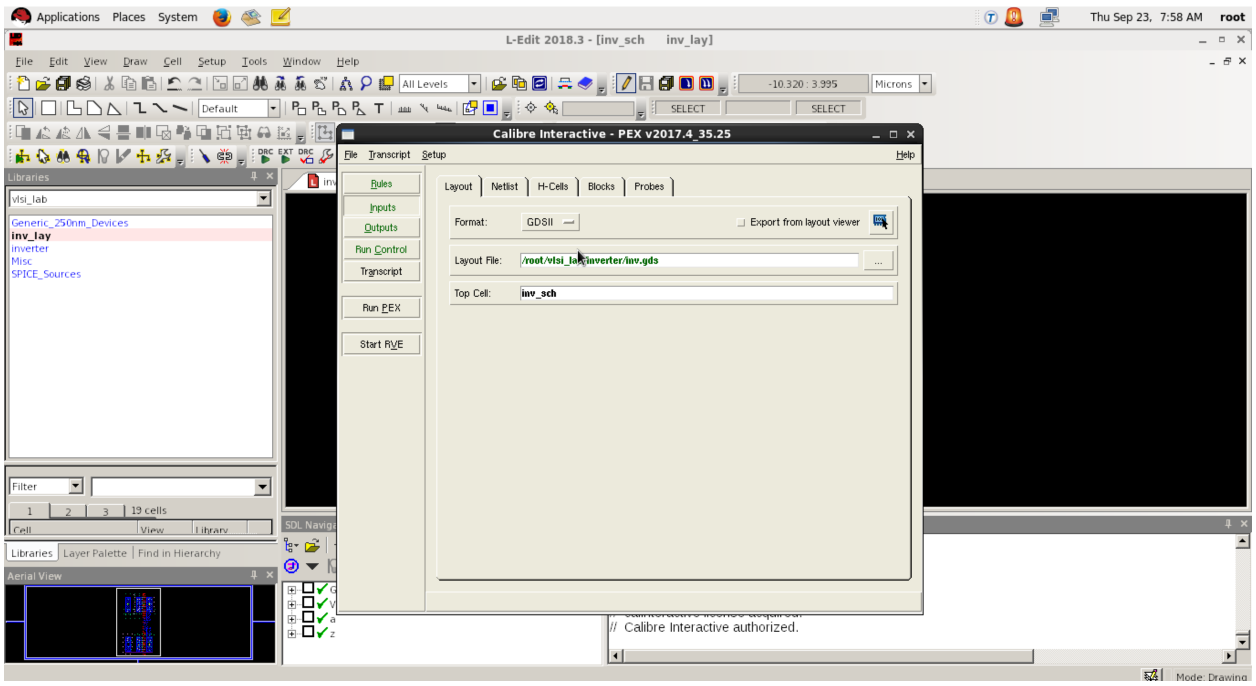
root → Tanner_EDA → Tanner_Tools_v2018.3 → Process → Generic_250nm → Rules → calibre → Generic_250nm_xRC.cal → ok



PEX Rules File: 18.3/Process/Generic_250nm/Rules/Calibre/Generic_250nm_xRC.cal

PEX Run Directory: /root/vlsi_lab/inverter

- **Click on Inputs**
  - Go to layout tab → de-select **export from layout viewer** → browse layout file from your folder, (**inv.gds** file)

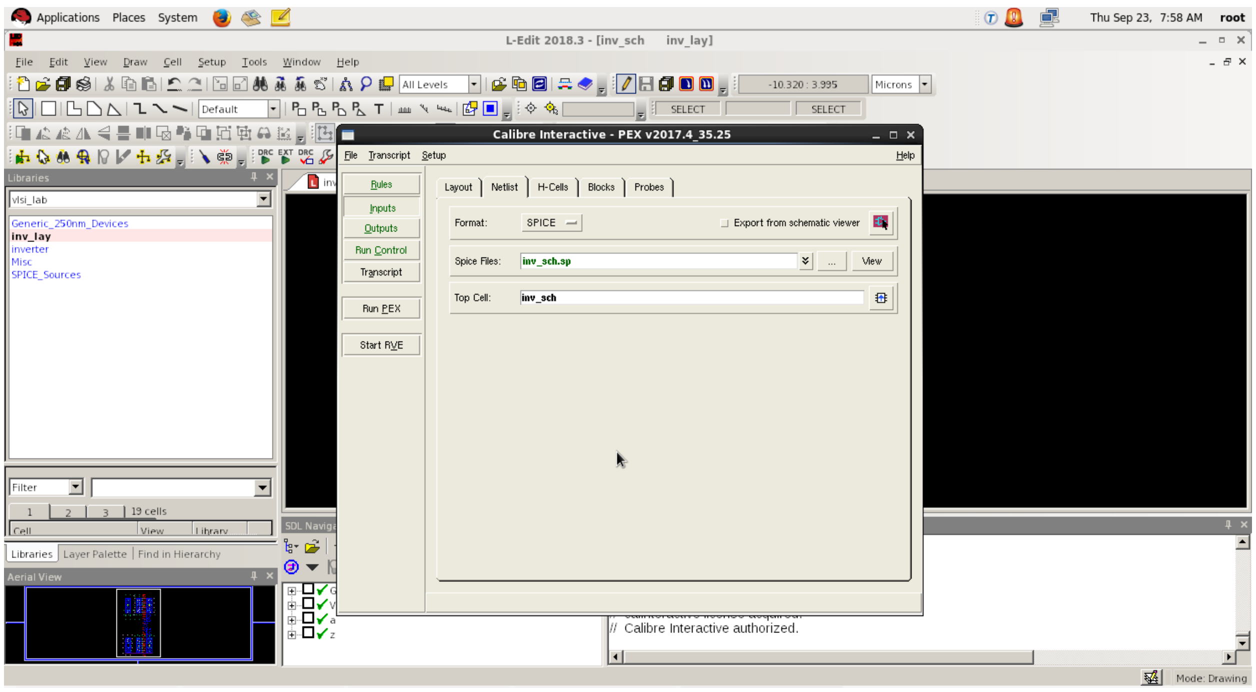


Format: GDSII

Layout File: /root/vlsi_lab/inverter/inv.gds

Top Cell: inv_sch

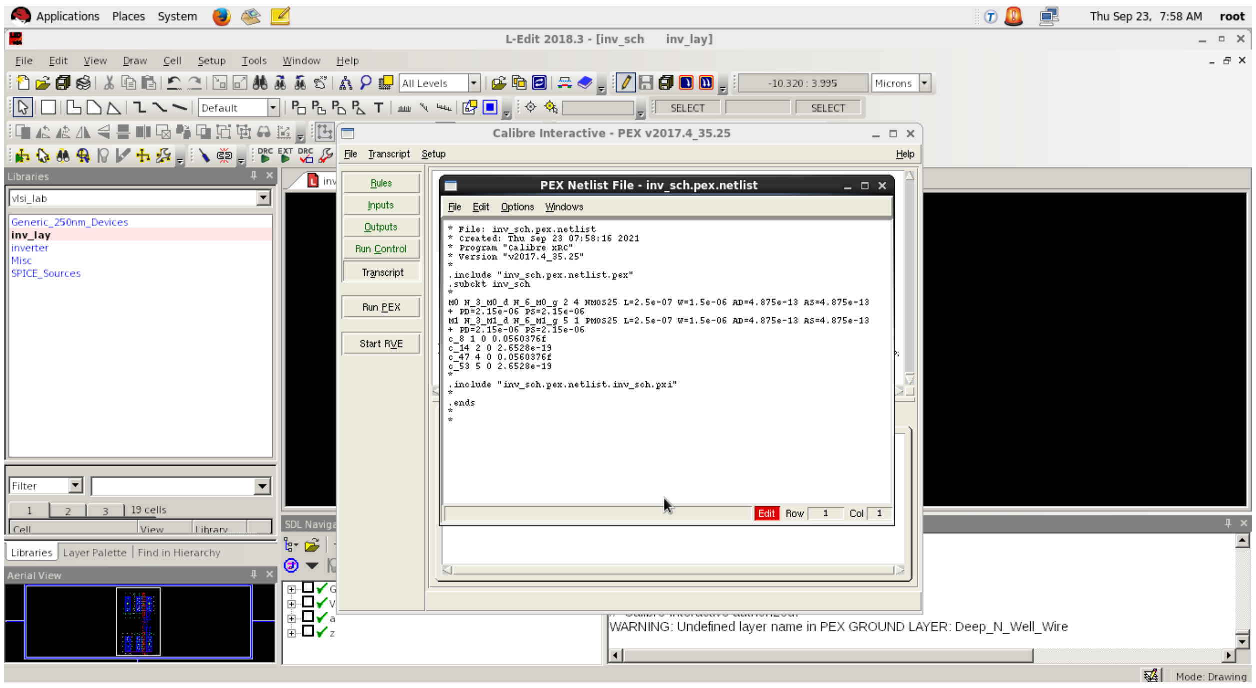
  - Go to netlist tab → Browse netlist file from your folder (.sp file) → **Add** → **OK**



Format: SPICE
Export from schematic viewer
Spice Files: inv_sch.sp
Top Cell: inv_sch

// Calibre Interactive authorized.

- **Click on Run PEX.**


Calibre Interactive - PEX v2017.4_35.25
PEX Netlist File - inv_sch.pex.netlist
File
Edit
Options
Windows

```
* File: inv_sch.pex.netlist
* Created: Thu Sep 23 07:58:16 2021
* Program "Calibre xRC"
* Version "v2017.4_35.25"
*
.include "inv_sch.pex.netlist.pex"
.subckt inv_sch
*
M0 N_3_M0_d N_6_M0_g 2 4 NMOS25 L=2.5e-07 W=1.5e-06 AD=4.875e-13 AS=4.875e-13
+ PD=2.15e-06 PS=2.15e-06
M1 N_3_M1_d N_6_M1_g 5 1 PMOS25 L=2.5e-07 W=1.5e-06 AD=4.875e-13 AS=4.875e-13
+ PD=2.15e-06 PS=2.15e-06
c_8 1 0 0.0560376f
c_14 2 0 2.6528e-19
c_47 4 0 0.0560376f
c_53 5 0 2.6528e-19
*
.include "inv_sch.pex.netlist.inv_sch.pxi"
*
.ends
*
*
```

WARNING: Undefined layer name in PEX GROUND LAYER: Deep_N_Well_Wire

PART – A Analog Design

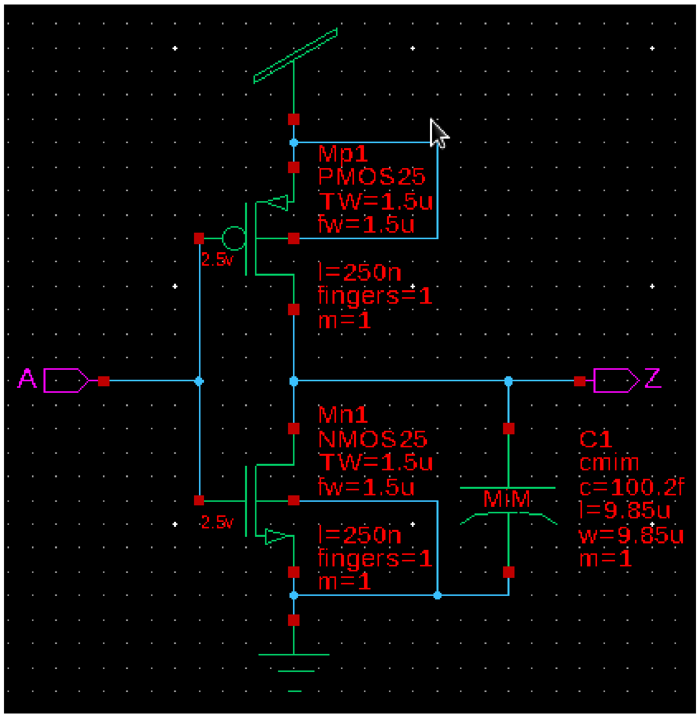
Experiment No: 01

a. **Capture the schematic of CMOS inverter with load capacitance of 0.1pF and set the widths of inverter with $W_n = W_p$, $W_n = 2W_p$, $W_n = W_{p/2}$ and length at selected technology. Carry out the following:**
   i. **Set the input signal to a pulse with rise time, fall time of 1ns and pulse width of 10ns and time period of 20ns and plot the input voltage and output voltage of designed inverter?**
   ii. **From the simulation results compute t_{pHL}, t_{pLH} and t_d for all three geometrical settings of width?**

   Tabulate the results of delay and find the best geometry for minimum delay for CMOS inverter?

b. **Draw layout of inverter with W_p/W_n = 40/20, use optimum layout methods. Verify for DRC and LVS, extract parasitic. Record the observations.**

Schematic diagram:



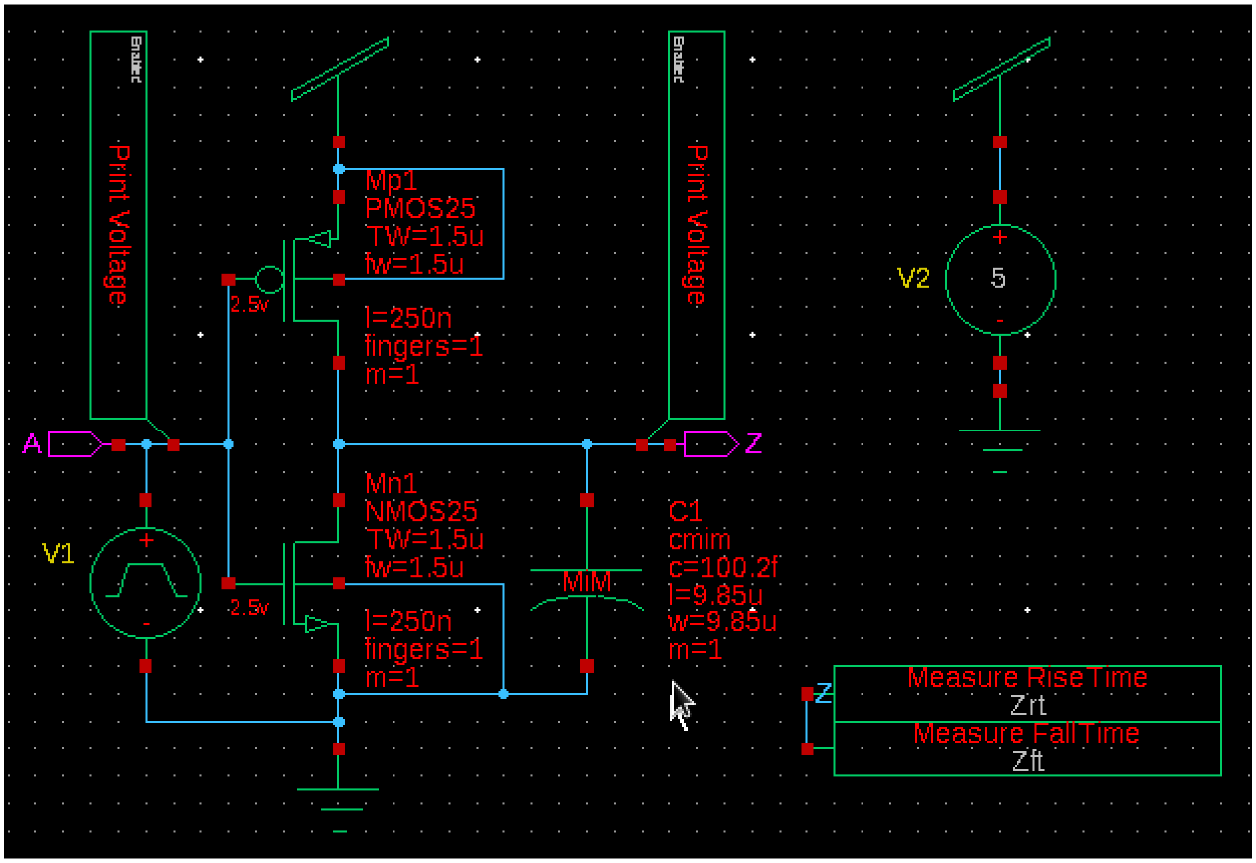
Case 1: $W_n = W_p = 1.5\mu$

Case 2: $W_n = 2W_p$, $W_n = 1.5\mu$, $W_p = 0.75\mu$

Case 3: $W_n = W_{p/2}$, $W_n = 1.5\mu$, $W_p = 3\mu$

Simulation:

Transient analysis:



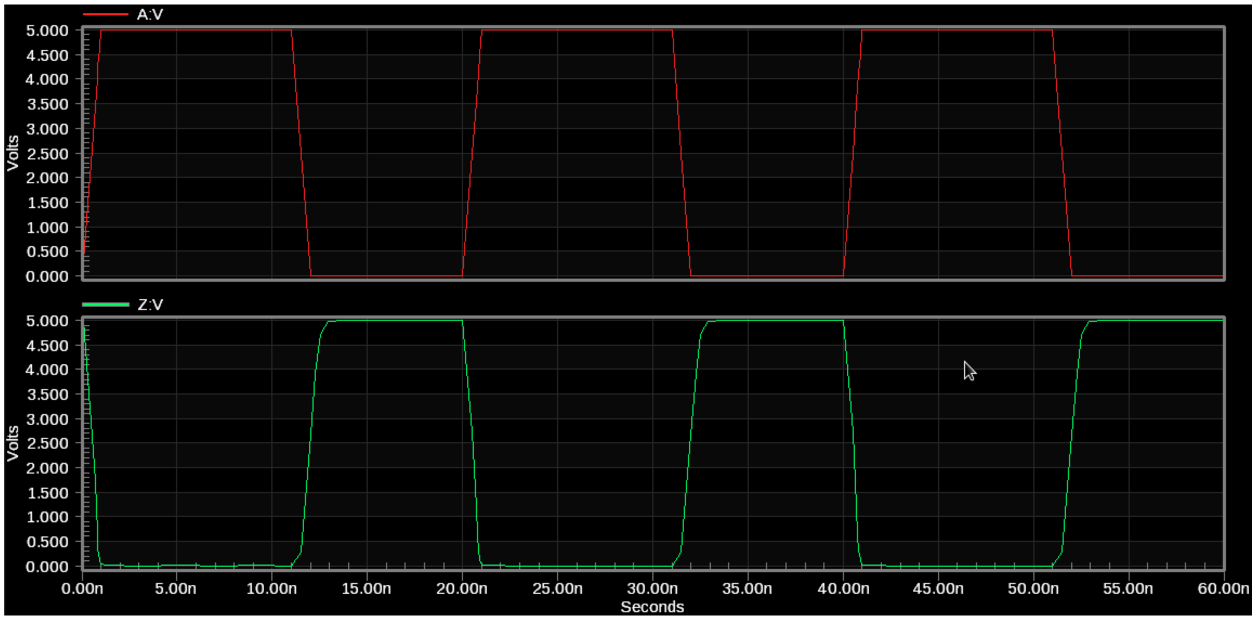
Input A (Vpulse Properties):

Fall Time	1ns
Rise Time	1ns
Pulse Width	10ns
Period	20ns

Transient Analysis parameters:

Stop Time	60ns
Maximum time step	1ns

Transient analysis Waveform:



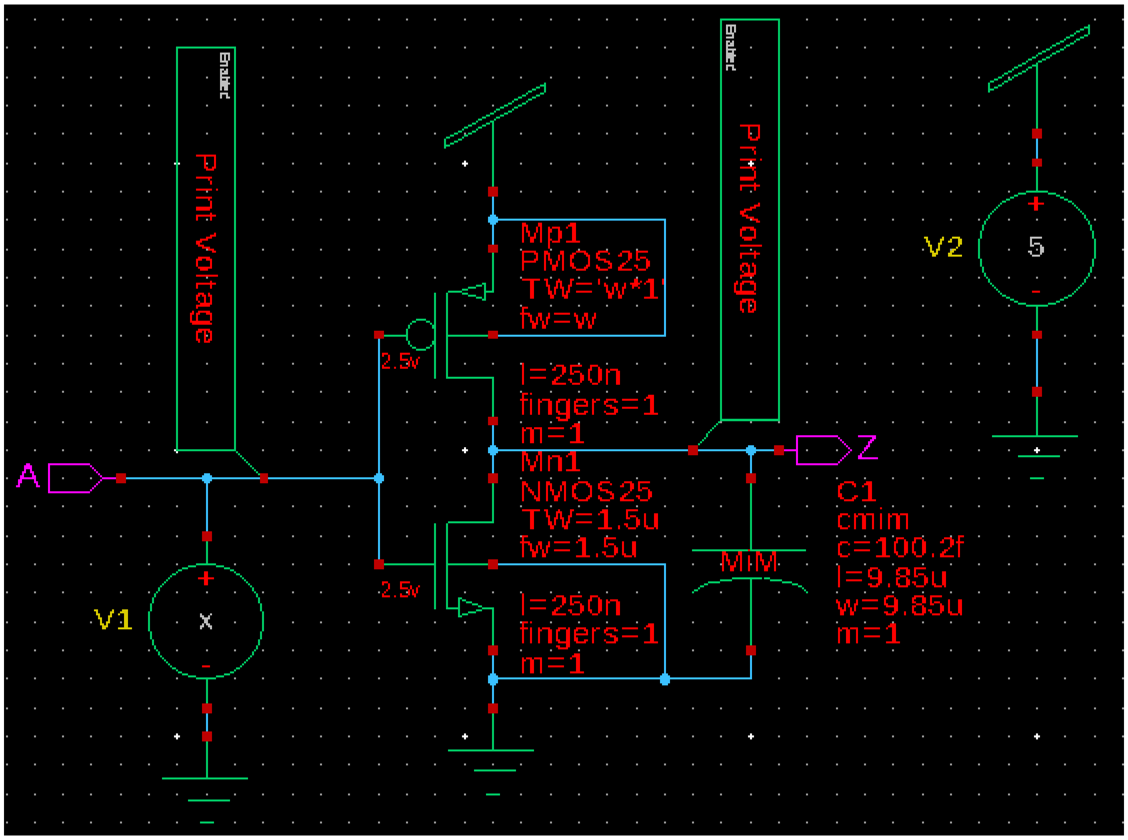
Delay tabulation for different width of pMOS and nMOS transistor.

Transistor Width	MOSFET	Width	t_{pHL}	t_{pLH}	t_d
$W_n = W_p$	pMOS	1.5µm	670.0519ps	907.7724ps	788.9121ps
	nMOS	1.5µm			
$W_n = 2W_p$	pMOS	0.75µm	643.1437ps	1329.5ps	986.3218ps
	nMOS	1.5µm			
$W_n = W_{p/2}$	pMOS	3µm	725.2264ps	625.0963ps	675.1613ps
	nMOS	1.5µm			

Therefore, the best geometry for minimum delay is:

$W_n = W_{p/2}$, $W_n = 1.5\mu$, $W_p = 3\mu$

DC Analysis:



Parameters:

Parameter 1 for voltage source connected to input terminal A

Name → **x**

Value → **5**

Parameter 2 for width of pMOS transistor

Name → **w**

Value → **3u**

Parameter Properties:

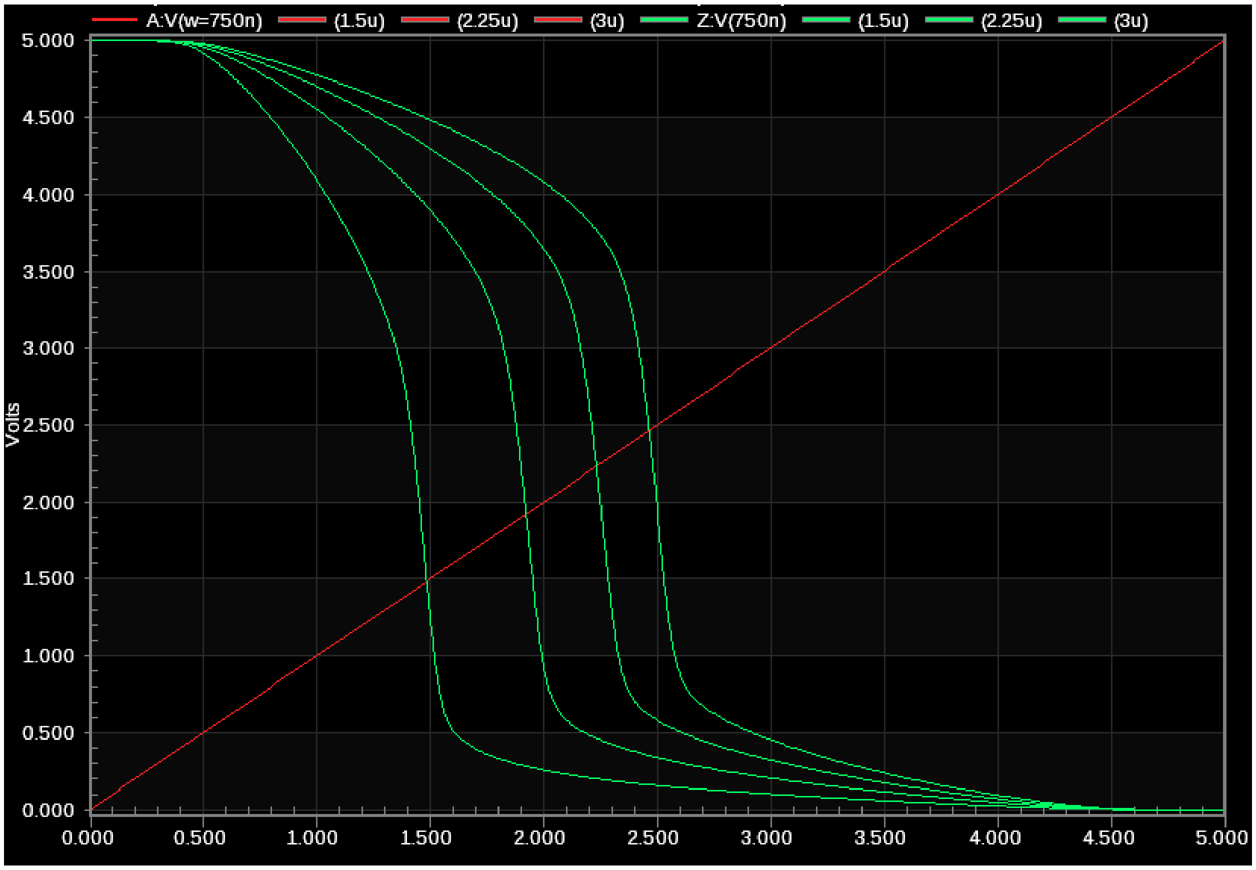
Parameter 1 for voltage source

Source or Parameter Name	**x**
Start value / List of Points	**0**
Stop Value	**5**
Step/ No of Points	**0.01**
Sweep Type	**lin**

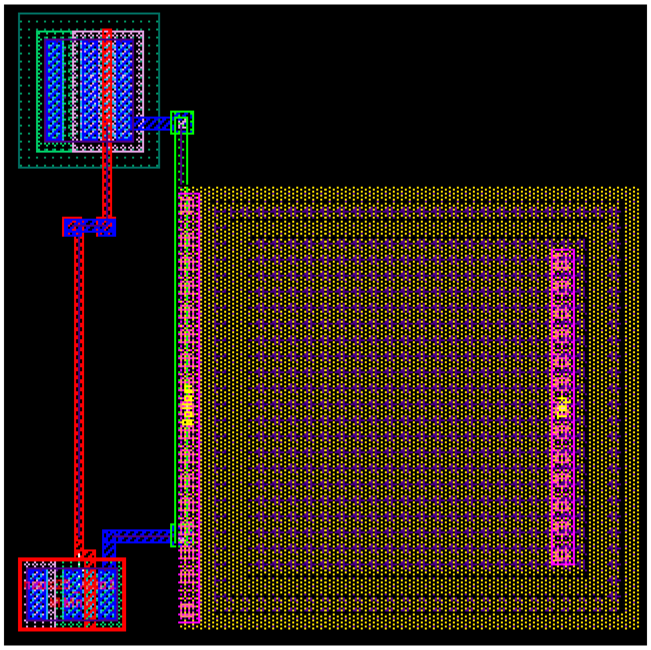
Parameter 2 for width of pMOS

Source or Parameter Name	**w**
Start value / List of Points	**0.75u**
Stop Value	**3u**
Step/ No of Points	**0.75u**
Sweep Type	**lin**

DC Characteristics:



Layout Diagram:

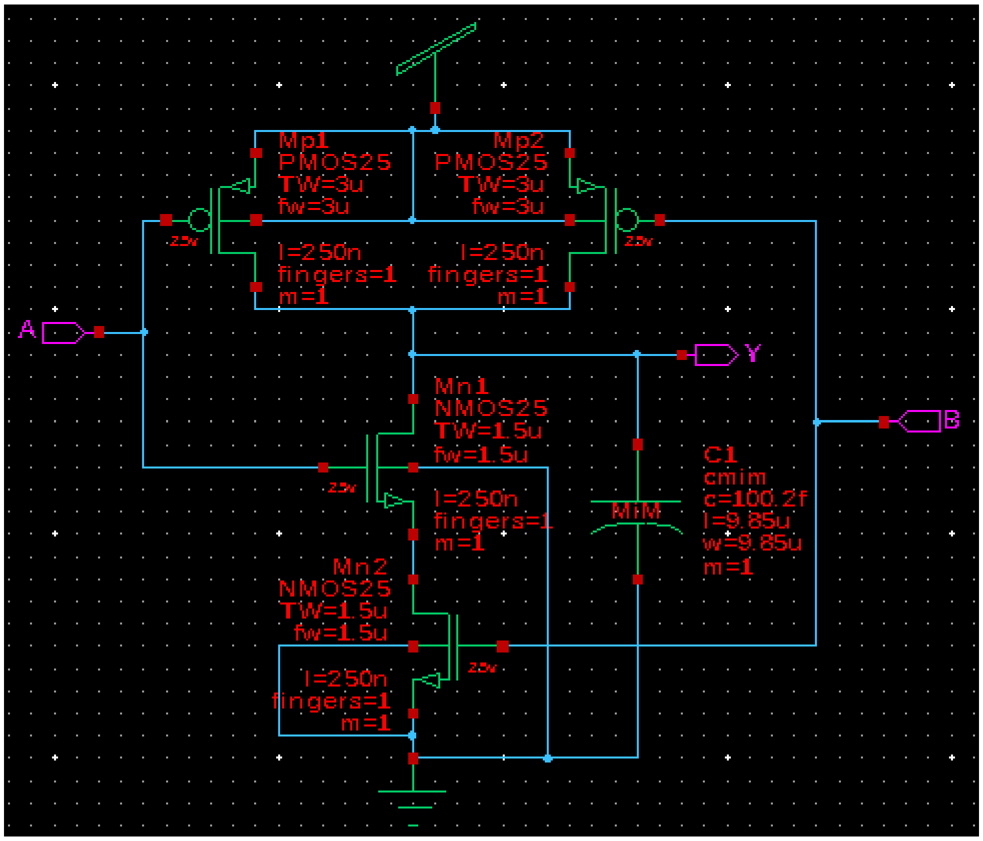


Experiment No: 02

a. **Capture the schematic of 2-input CMOS NAND gate having similar delay as that of CMOS inverter computed in experiment 1. Verify the functionality of NAND gate and also find out the delay td for all four possible combinations of input vectors. Table the results. Increase the drive strength to 2X and 4X and tabulate the results.**

b. **Draw layout of NAND with W_p/W_n = 40/20, use optimum layout methods. Verify for DRC and LVS, extract parasitic. Record the observations.**

Schematic diagram:



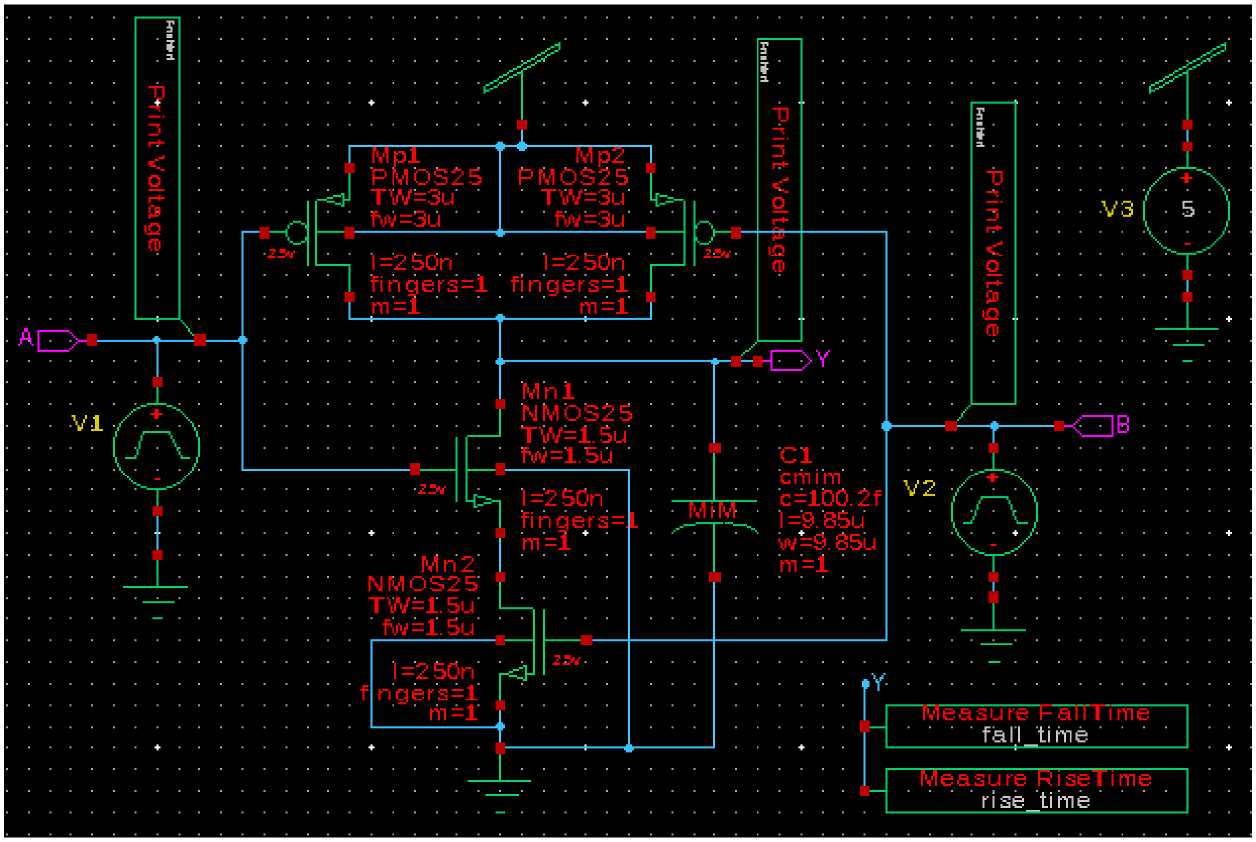
Case 1: $W_p = 3\mu$

Case 2: $W_p = 6\mu$ **(Drive strength = 2X)**

Case 3: $W_p = 12\mu$ **(Drive strength = 4X)**

Simulation:

Transient analysis



Input A (Vpulse Properties):

Fall Time	1ns
Rise Time	1ns
Pulse Width	10ns
Period	20ns

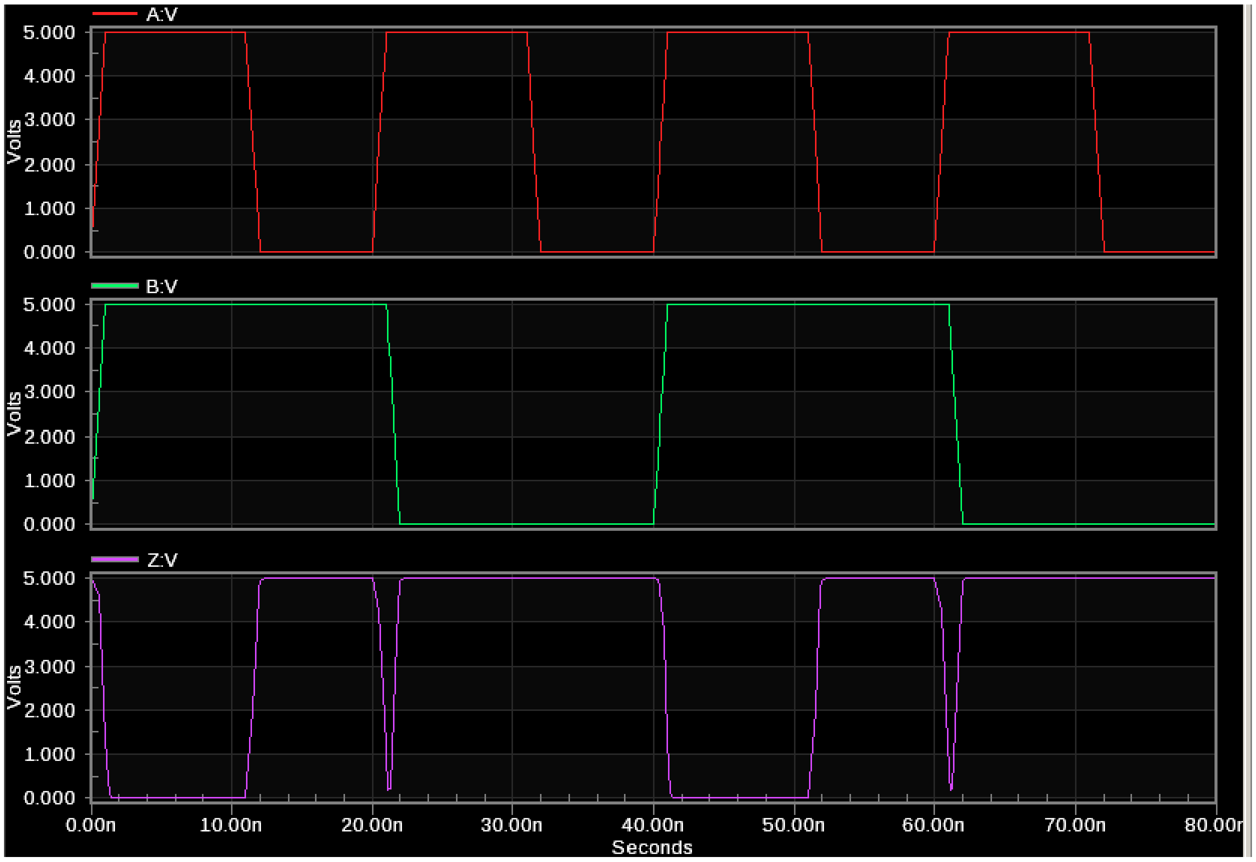
Input B (Vpulse Properties):

Fall Time	1ns
Rise Time	1ns
Pulse Width	20ns
Period	40ns

Transient Analysis parameters:

Stop Time	60ns
Maximum time step	1ns

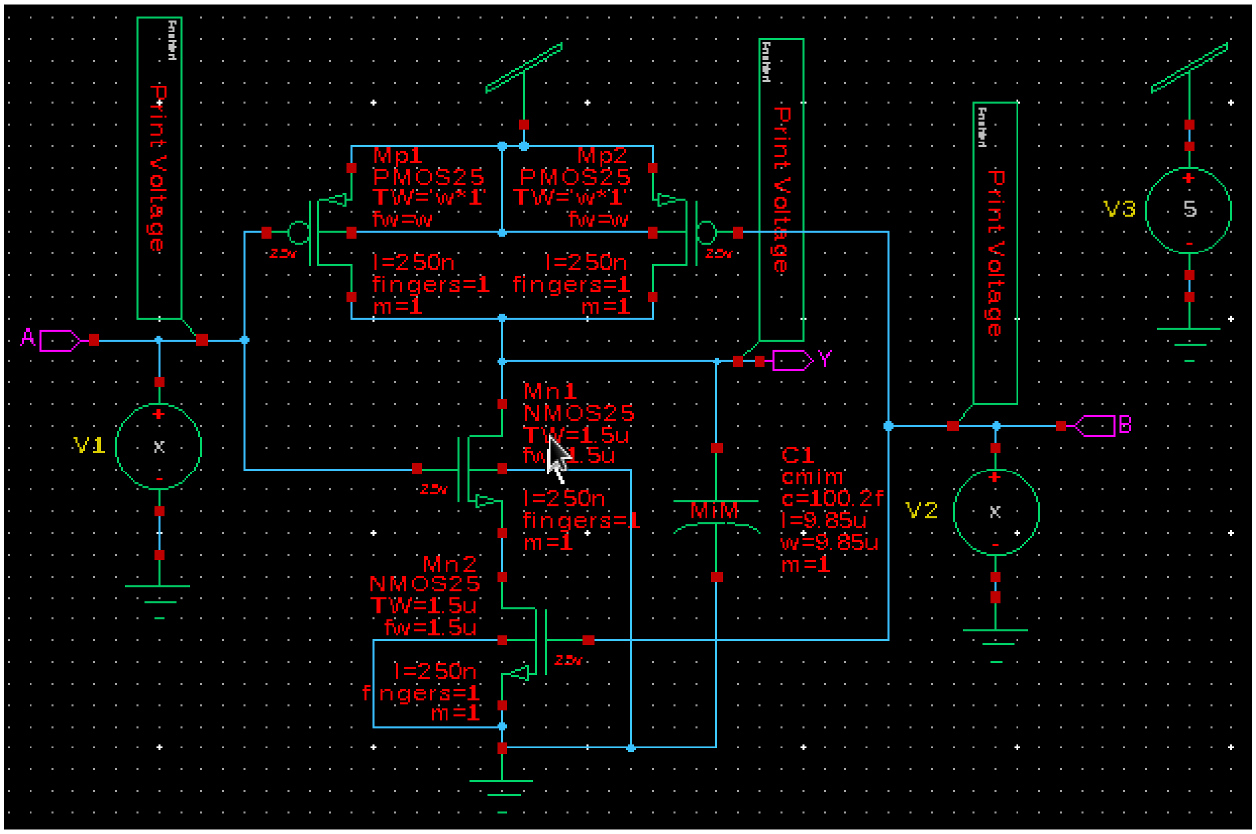
Transient analysis Waveform:



Delay tabulation for different drive strength:

NAND Gate	MOSFET	Width	t_{pHL}	t_{pLH}	t_d
NAND2X1	pMOS	3µm	736.950ps	753.616ps	676.4925ps
	nMOS	1.5µm			
NAND2X2	pMOS	6µm	754.126ps	705.024ps	582.8179ps
	nMOS	1.5µm			
NAND2X4	pMOS	12µm	818.778ps	537.099ps	547.1139ps
	nMOS	1.5µm			

DC Analysis:



Parameters:

Parameter 1 for voltage source connected to input terminals A and B

Name ⟶ **x**

Value ⟶ **5**

Parameter 2 for width of pMOS transistors

Name ⟶ **w**

Value ⟶ **12u**

Parameter Properties:

Parameter 1 for voltage source

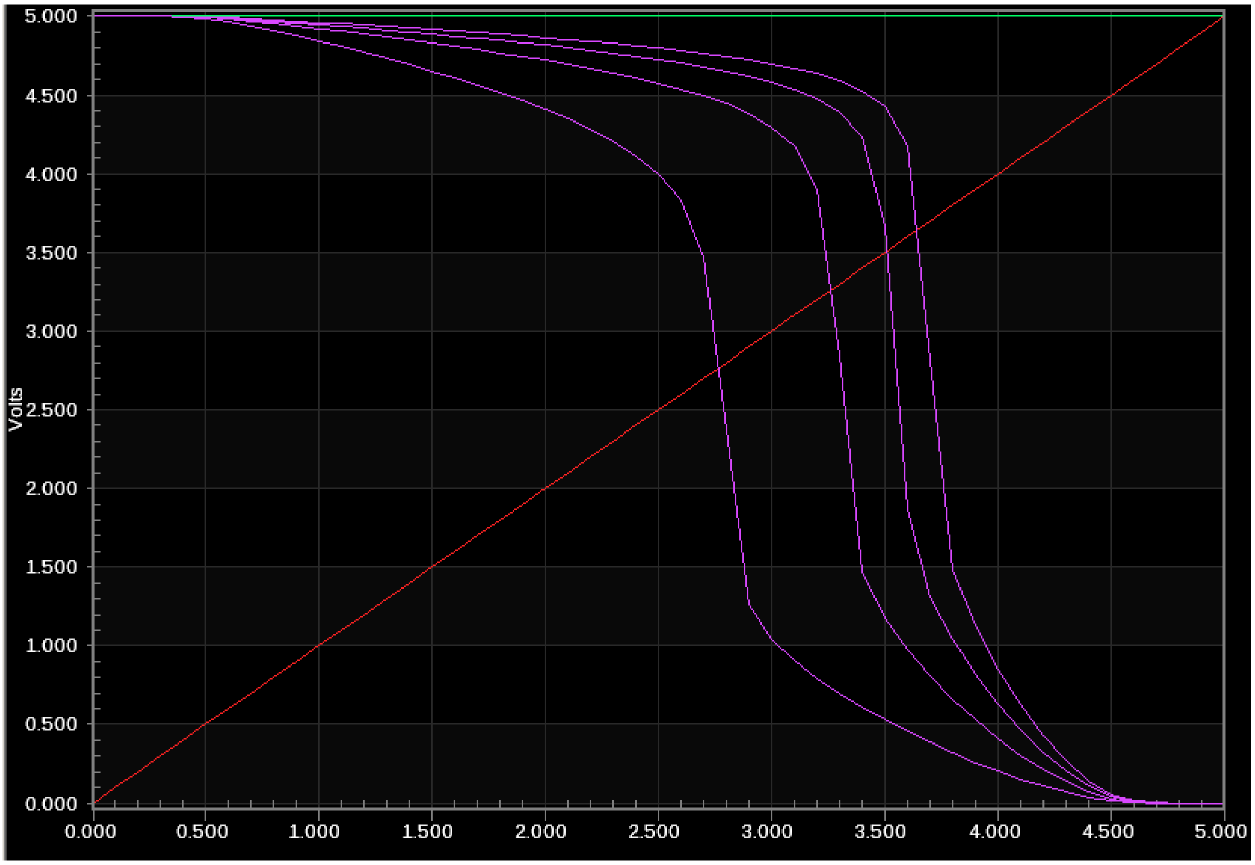
Source or Parameter Name	**x**
Start value / List of Points	**0**
Stop Value	**5**
Step/ No of Points	**0.01**
Sweep Type	**lin**

Parameter 2 for width of pMOS

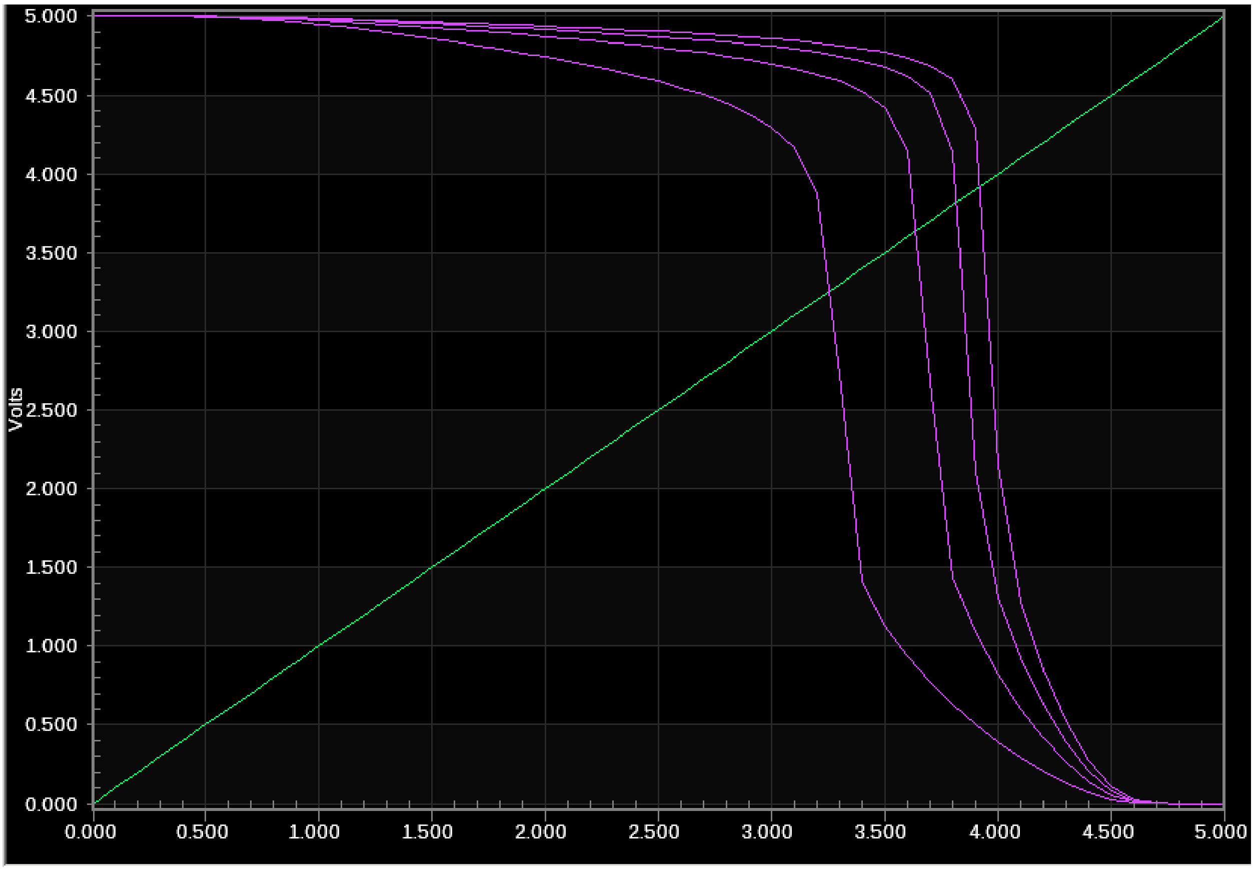
Source or Parameter Name	**w**
Start value / List of Points	**3u**
Stop Value	**12u**
Step/ No of Points	**3u**
Sweep Type	**lin**

DC Characteristics:

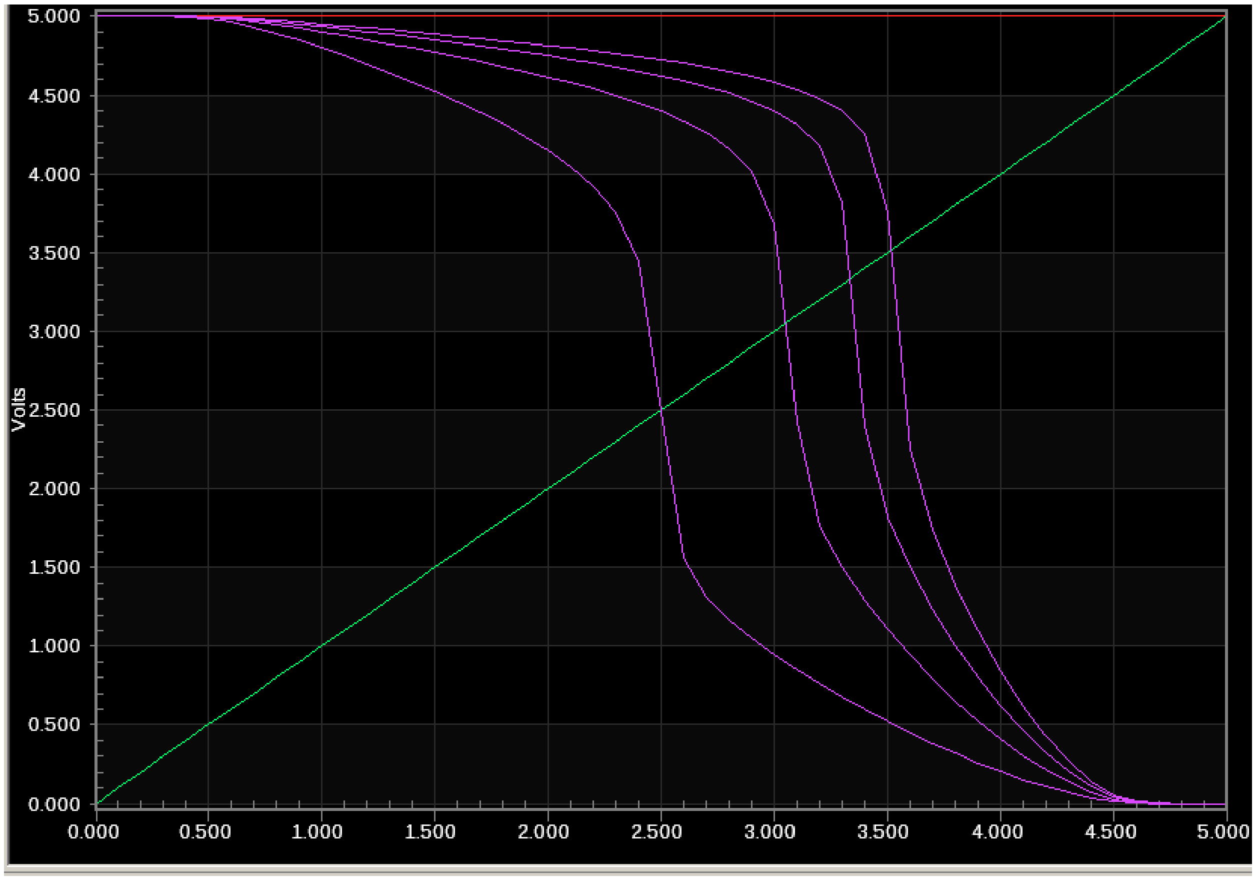
Case 1: A changes (0V to 5V), B = 5V



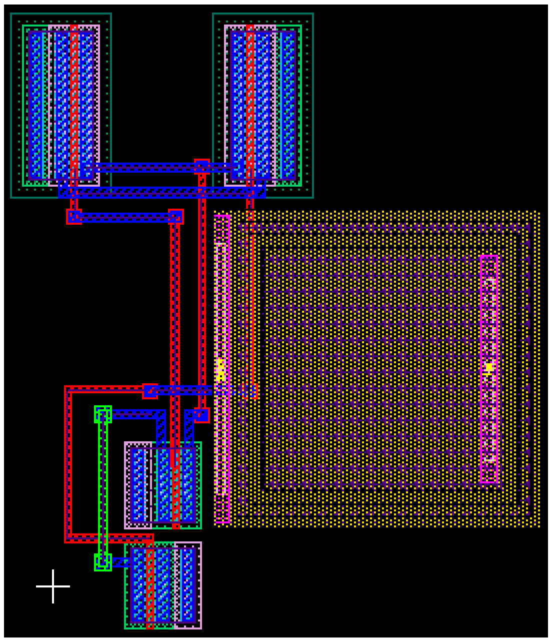
Case 2: A&B changes (0V to 5V)



Case 3: A = 5V, B changes (0V to 5V)



Layout Diagram:



PART – B Digital Design

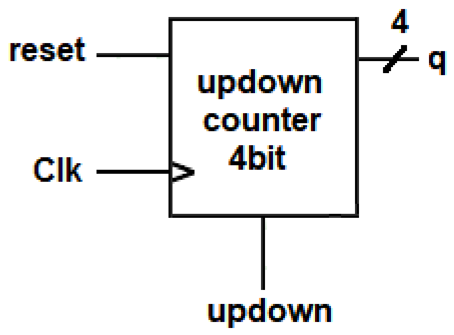
Experiment No: 01

Write Verilog code for 4-bit up/down asynchronous reset counter and verify the functionality using test bench

Perform the above for 32-bit up/down counter

4-bit up/down asynchronous reset counter:

Block Diagram:



Verilog Code:

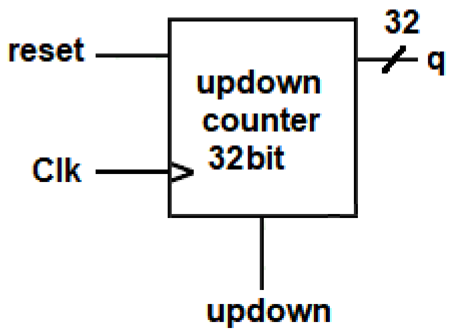
```
        module udcounter (clk, reset, up_down, q);
        input clk, reset, up_down;
        output [3:0] q;
        reg [3:0] temp;

        always @ (posedge clk or posedge reset)
                begin
                if (reset)
                        temp = 4'd0;
                else if (up_down == 1)
                        temp = temp + 1'b1;
                else
                        temp = temp - 1'b1;
                end
        assign q = temp;
        endmodule
```

Stimulus Block / Test bench:

```
    module udcounter_tb;
        reg clk, reset, up_down;
        wire [3:0] q;
        udcounter uut (clk, reset, up_down, q);
    initial
        begin
            clk = 0;
            forever #5 clk = ~clk;
        end
    initial
        begin
            reset = 1;
            #20 reset = 0; up_down = 0;
            #100 reset = 0; up_down = 1;
        end
    endmodule
```

32-bit up/down asynchronous reset counter:

Block Diagram:



Verilog Code:

```
module udcounter (clk, reset, up_down, q);
input clk, reset, up_down;
output [31:0] q;
reg [31:0] temp;

always @ (posedge clk or posedge reset)
        begin
        if (reset)
                temp = 32'd0;
        else if (up_down == 1)
                temp = temp + 1'b1;
        else
                temp = temp - 1'b1;
        end
assign q = temp;
endmodule
```

Stimulus Block / Test bench:

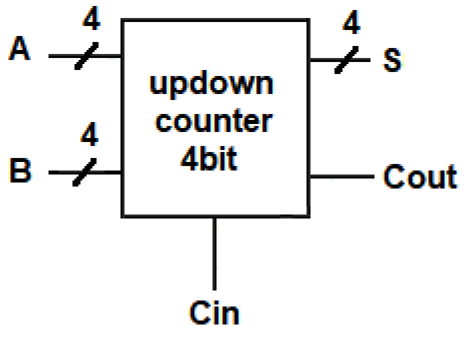
```
module udcounter_tb;
        reg clk, reset, up_down;
        wire [31:0] q;
        udcounter uut (clk, reset, up_down, q);
initial
        begin
                clk = 0;
                forever #5 clk = ~clk;
        end
initial
        begin
                reset = 1;
```

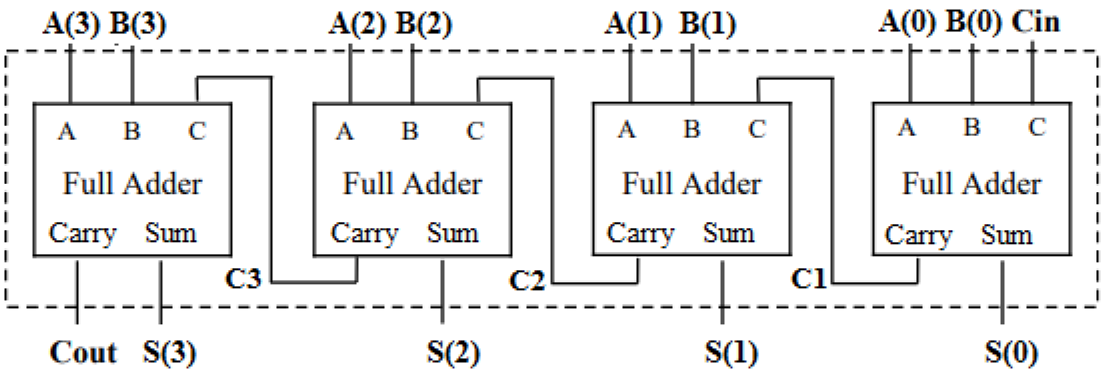
```
            #20 reset = 0; up_down = 0;
            #100 reset = 0; up_down = 1;
        end
    endmodule
```

Experiment No: 02

Write Verilog code for 4-bit adder and verify its functionality using test bench.

Block Diagram:





Verilog code:

```
module adder_4bit (a, b, cin, s, cout);
input [3:0] a, b;
input cin;
output [3:0] s;
output cout;
wire c1, c2, c3;
full_adder fa0 (a[0], b[0], cin, s[0], c1);
full_adder fa1 (a[1], b[1], c1, s[1], c2);
full_adder fa2 (a[2], b[2], c2, s[2], c3);
full_adder fa3 (a[3], b[3], c3, s[3], cout);
endmodule
```

```
module full_adder(a, b, c, sum, carry);
input a, b, c;
output sum, carry;
        assign sum = a ^ b ^ c;
        assign carry = (a & b) | (b & c) | (a& c);
endmodule
```

Stimulus Block / Test bench:

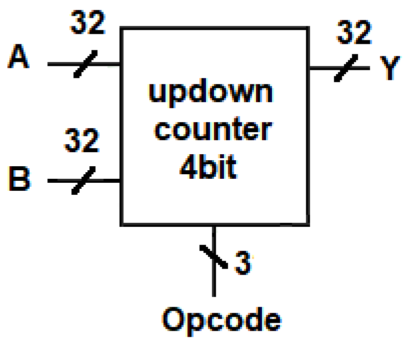
```
module adder_4bit_tb;
reg [3:0] a, b;
reg cin;
wire [3:0] s;
wire cout;
adder_4bit uut (a, b, cin, s, cout);
initial
        begin
                a = 0; b = 0; cin = 0;
                #100 a = 2; b = 3; cin = 0;
                #100 a = 12; b = 6; cin = 1;
        end
endmodule
```

Experiment No: 03

Write Verilog code for 32-bit ALU supporting four logical and four arithmetic operations, use case statement and if statement for ALU behavioral modeling. Perform functional verification using test bench

32-bit ALU using CASE statement:

Block Diagram:



Verilog Code:

```
        module alu_32bit_case (a, b, opcode, y);
        input [31:0] a;
        input [31:0] b;
        input [2:0] opcode;
        output reg [31:0] y;
        always @ (a, b, opcode)
                begin
                case(opcode)
                        3'b000: y = a & b;
                        3'b001: y = a | b;
                        3'b010: y = ~ (a & b);
                        3'b011: y = ~ (a | b);
                        3'b010: y = a + b;
                        3'b011: y = a - b;
                        3'b100: y = a * b;
                        default: y = 32'bx;
                endcase
                end
        endmodule
```

Test bench:

```
module alu_32bit_case_tb;
reg [31:0] a;
reg [31:0] b;
reg [2:0] opcode;
wire [31:0] y;
alu_32bit_case uut (a, b, opcode, y);
initial
        begin
                a = 32'h00000000;
                b = 32'hFFFFFFFF;
                #10 opcode = 3'b000;
                #10 opcode = 3'b001;
                #10 opcode = 3'b010;
                #10 opcode = 3'b100;
        end
endmodule
```

32-bit ALU using CASE statement:

Verilog Code:

```
module alu_32bit_case(a,b,opcode,y);
input [31:0] a;
input [31:0] b;
input [2:0] opcode;
output reg [31:0] y;
always @ (a, b, opcode)
```

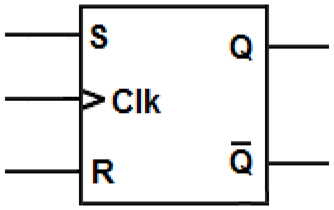
```
        begin
        if (opcode == 3'b000)
                y = a & b;              // AND Operation
        else if (opcode == 3'b001)
                y = a | b;              // OR Operation
        else if (opcode == 3'b010)
                y = a + b;              // Addition
        else if (opcode == 3'b011)
                y = a - b;              // Subtraction
        else if (opcode == 3'b100)
                y = a * b;              // Multiply
        else
                y = 32'bx;
        end
endmodule
```

Experiment No: 04

Write Verilog code for Flip-flops and verify its functionality using test bench.

a. **SR Flip-flop**

Block Diagram:



Truth Table:

INPUTS			OUTPUTS	
clk	**s**	**r**	**q**	**qb**
0, 1, ↓	X	X	X	**X**
↑	0	0	q	qb
↑	0	1	0	1
↑	1	0	1	0
↑	1	1	invalid	invalid

Verilog Code:

```
module sr_ff (clk, s, r, q, qb);
input clk, s, r;
output q, qb;
reg q, qb;
always @ (posedge clk)
        begin
                case ({s, r})
                        2'b00: q = q;
                        2'b01: q = 0;
                        2'b10: q = 1;
```

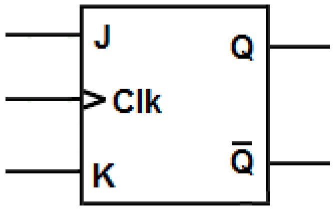
```
				2'b11: q = 1'bz;
			endcase
			qb = ~ q;
		end
	endmodule
```

Stimulus Block / Test bench:

```
	module sr_ff_tb ( );
	reg clk, s, r;
	wire q, qb;
	sr_ff s1 (clk, s, r, q, qb);
	initial
		begin
			clk = 1'b0;
			forever #10 clk = ~ clk;
		end
	initial
		begin
			s = 1'b0; r = 1'b0;
			#20 s = 1'b0; r = 1'b1;
			#20 s = 1'b1; r = 1'b0;
			#20 s = 1'b1; r = 1'b1;
		end
	endmodule
```

b. JK flip-flop

Block Diagram:



Truth Table:

INPUTS			OUTPUTS	
clk	**j**	**k**	**q**	**qb**
0, 1, ↓	X	X	X	**X**
↑	0	0	q	qb
↑	0	1	0	1
↑	1	0	1	0
↑	1	1	qb	q

Verilog Code:

```
module jk_ff (clk, j, k, q, qb);
input clk, j, k;
output q, qb;
reg q, qb;
always @ (posedge clk)
        begin
                case ({j, k})
                        2'b00: q = q;
                        2'b01: q = 0;
                        2'b10: q = 1;
                        2'b11: q = ~ q;
                endcase
                qb = ~q;
        end
endmodule
```

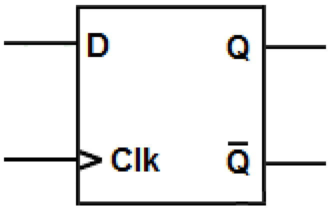
Stimulus Block / Test bench:

```
module jk_ff_tb();
```

```
reg clk, j, k;
wire q, qb;
jk_ff j1 (clk, j, k, q, qb);
initial
        begin
                clk = 1'b0;
                forever #5 clk = ~ clk;
        end
initial
        begin
                j = 1'b0; k = 1'b0;
                #20 j = 1'b0; k = 1'b1;
                #20 j = 1'b1; k = 1'b0;
                #20 j = 1'b1; k = 1'b1;
        end
endmodule
```

c. D flip-flop

Block Diagram:



Truth Table:

INPUTS		**OUTPUTS**	
clk	**d**	**q**	**qb**
0, 1, ↓	X	X	X
↑	1	1	0
↑	0	0	1

Verilog Code:

```
module d_ff (clk, d, q, qb );
input clk, d;
output q, qb;
reg q, qb;
always @ (posedge clk)
        begin
                q = d;
                qb = ~q;
        end
endmodule
```

Stimulus Block / Test bench:

```
module d_ff_tb();
reg clk, d;
wire q, qb;
d_ff d1 (clk, d, q, qb);
initial
        begin
                clk = 1'b0;
                forever #5 clk = ~ clk;
        end
initial
        begin
                d = 1'b0;
                #20 d = 1'b1;
        end
endmodule
```