

**EEE 476 VLSI Sessional**

**Experiment No.: 01**

**Draw the schematic of a 2-input NAND Gate, Create a Symbol and  
Perform Simulation**

---

**Learning Objectives:**

- To login in to the Cadence Server shell and start the Cadence virtuoso software
- To create a working library
- To draw the schematic of a 2-input NAND gate in Cadence Virtuoso Schematic Editor
- To create a symbol view of the NAND gate from the schematic
- To simulate the NAND gate using MMSIM Spectre

**1-1. Getting Started**

1. Run **Xming** from the desktop. Then run **putty.exe** from your desktop. Under the **Saved Sessions** dialog box click on **VLSI** and then click on **Load**. (**confirm that Server IP address is 172.16.93.205, port is 22 and X forwarding is clicked**) Next click on Open tab. Log in to Cadence workstation using the username and password provided. Your user name will be *groupx* and your password will be *vlsi*, where *x* is any number from 1 to 20.
2. After successful login, type `$source .bashrc` in a command window and type `cd cds_work` and then press enter.
3. Type **virtuoso &** and virtuosos **Command Interpreter Window (CIW)** appears at the bottom of the screen.

**1-2. Creation of a Design Project and adding Library to the design**

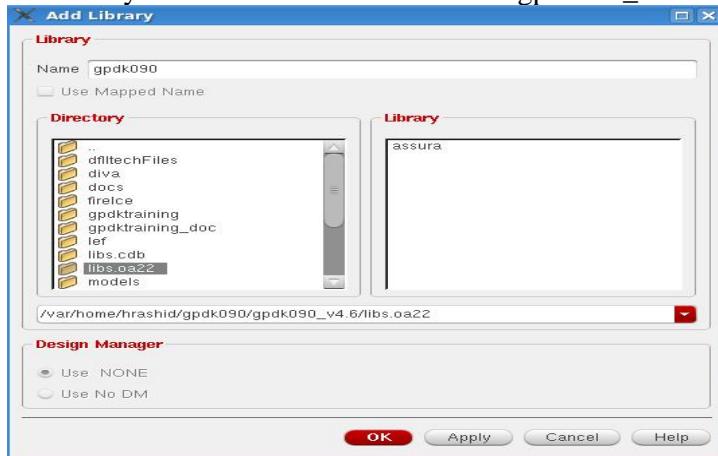
A unique directory should be created for each circuit design project. For you, the directory **cds\_work** is created by the administrator. So all your project works must be performed under this directory.

First of all we will add the generalized 90 nm process design kit from cadence known as **gdk090** to our library path as follows.

## Adding the gpdk090 library to your library path

The Cadence® Library Path Editor helps you to define the libraries being used in your design. You can set your cds.lib file or lib.defs file, or both, to point to the reference and design libraries you want to use in your design. (*However, in your case these directory may already has been added by the administrator. You can verify this by checking the library paths*)

1. In the CIW, execute **Tools → Library Path Editor → Edit → Add Library**. The add Library form appears.
2. Add the “gpdk090” library from “/usr/local/cadence/local/gpdk090\_v4.6/libs.0a22” and press ok



3. In the library path editor execute **File → Save as**. Make sure that both cds.lib and lib.defs are selected.

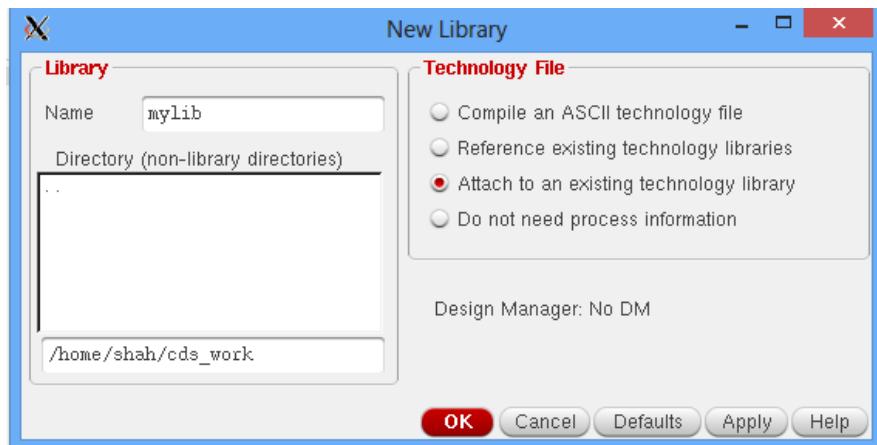
Click OK the new paths will be saved and you will be able to use the above library.



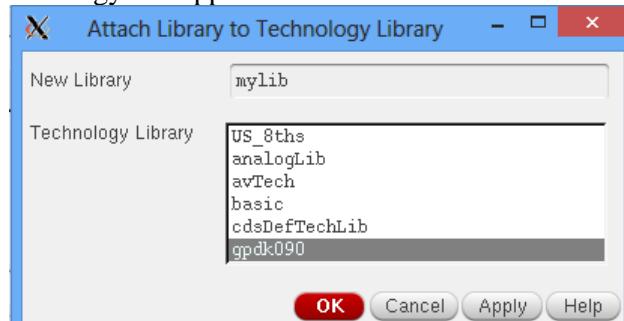
4. Following the above procedure add the **analogLib** library from the following location  
`/usr/local/cadence/ic612/tools.lnx86/dfII/etc/cdslib/artist`
5. Similarly add the library **basic** from the following location  
`/usr/local/cadence/ic612/tools.lnx86/dfII/etc/cdslib`

Now we will create a working library to store our design and attach it to desired technology library.

6. In the CIW, execute **File**→**New**→**Library**
7. The new Library form appears. In the name field of the New Library type mylib.



8. In the field under the Directory Section, verify that the path to the library is set to /home/groupx/cds\_work
9. Select **Attach to an existing technology library** and click ok.
10. Attach Library to Technology Lib appears.



11. Select the gpdk90 technology library and click OK.
12. In the Library Manager window (**Tools** → **Library Manager**) verify that the mylib library is listed.

### **1-3. Creating the schematic of a 2 input NAND Gate**

Open a new schematic window in the *mylib* library and build the *nand2* design. You will simulate this design later on.

1. In the Command Interpreter Window (CIW) or Library manger execute **File**→**New**→**Cellview**.
2. Set up the create new file form as follows –
  - Library Name – mylib
  - Cell Name – nand2
  - View name – schematic

Open with – Schematic L

3. Click **Ok** when done.

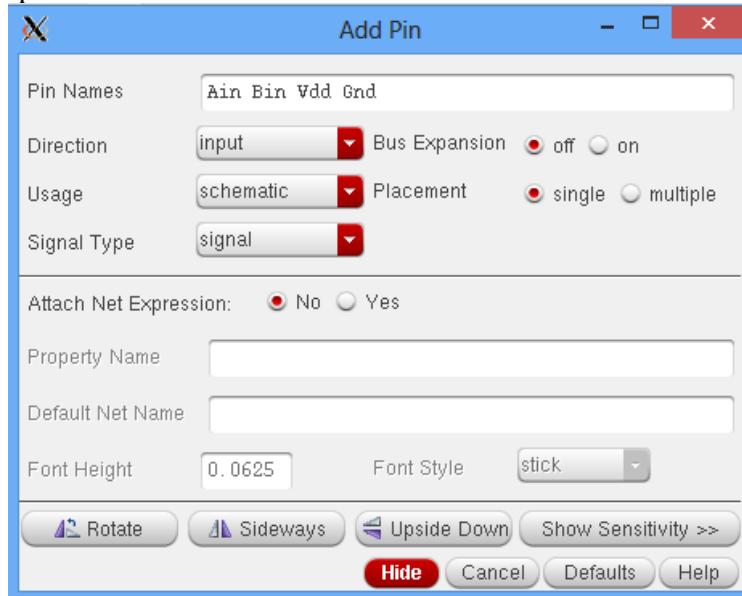
A blank schematic window for the *nand2* design appears. In the schematic window execute **Create→ Instance**. Make sure that the View name field in the form is set to symbol. You will update the Library Name, Cell Name, and the property values given in the table below as you place each component. After you complete the Create instance form, move your cursor to the schematic window and click left to place the component. The *nand2* design contains the following cells from the following library.

Library Name	Cell Name	Properties
gdk090	nmov1v	Total Width = 240n
gdk090	pmov1v	Total Width = 240n

If you place a component with wrong parameter values, use the **Edit→ Properties→ Objects** command to change the parameters. Use the **Edit→ Move** command if you placed components in the wrong location.

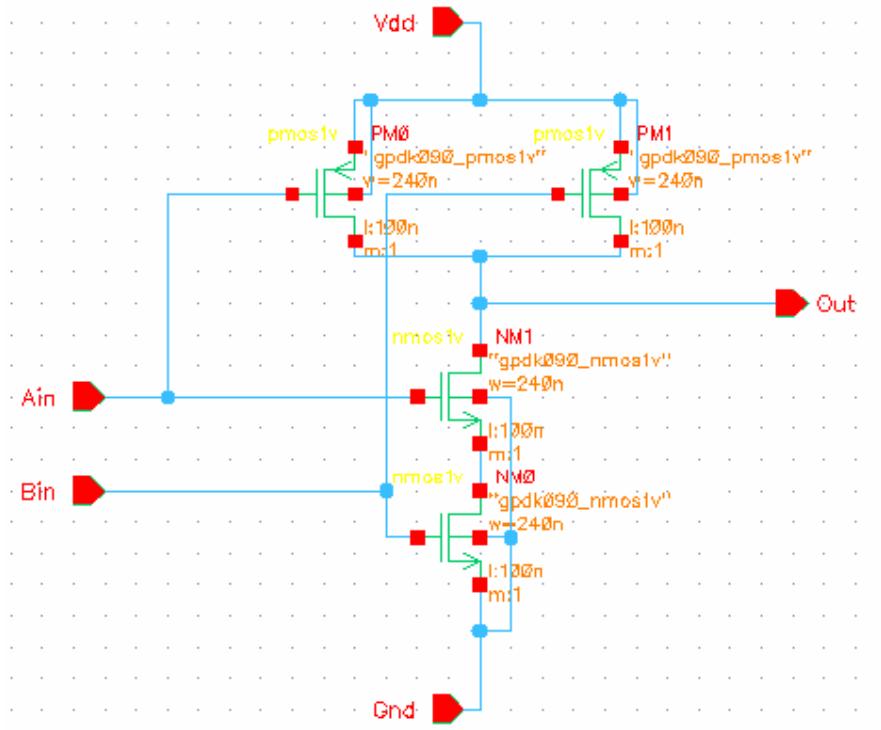
4. After entering the components, click Cancel in the Create Instance form or press Esc with your cursor in the schematic window.

5. Execute **Create→ Pin**. Enter the **Pin Names** Ain Bin Vdd Gnd, Select **Direction** to be input. Place the pins and then follow the same process to create and place a pin name of out with direction to be output.



6. Use **Create →Wire** to create the wire connections between pins and instances.

7. **File →Check and save** when your schematic looks like the following:



#### 1-4. Creating the Symbol View of a 2 input NAND Gate

1. In the nand2 schematic window execute **Create → Cellview → From Cellview**.
2. The **Cellview From Cellview** form appears. With the edit options function active, you can control the appearance of the symbol to generate.

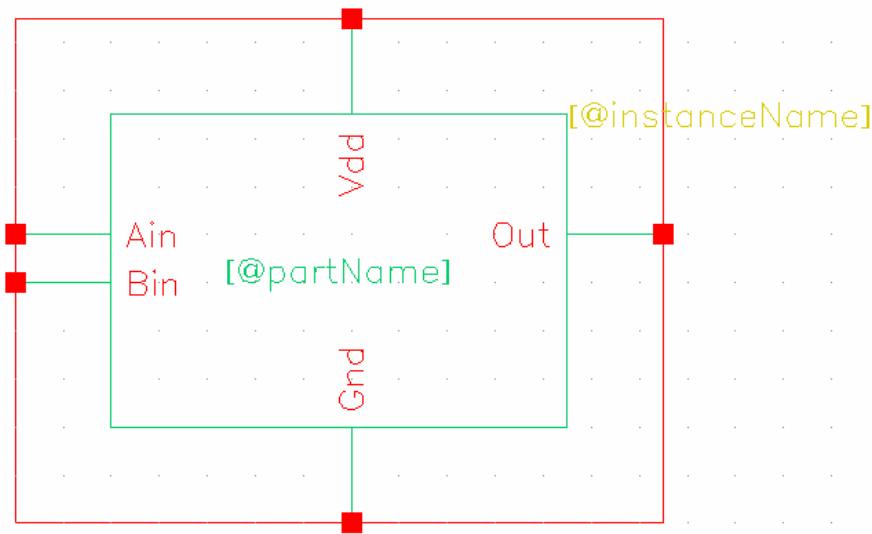


2. Verify that the From View Name field is set to **schematic**, and the To view Name field is set to **symbol**, with the Tool/Data Type set as **schematicSymbol**.

3. Click OK in the Cellview From Cellview form. The Symbol Generation Form appears. Change the values such that Ain & Bin is at the left, Out is at the right, Vdd is at the top and Gnd at the bottom:

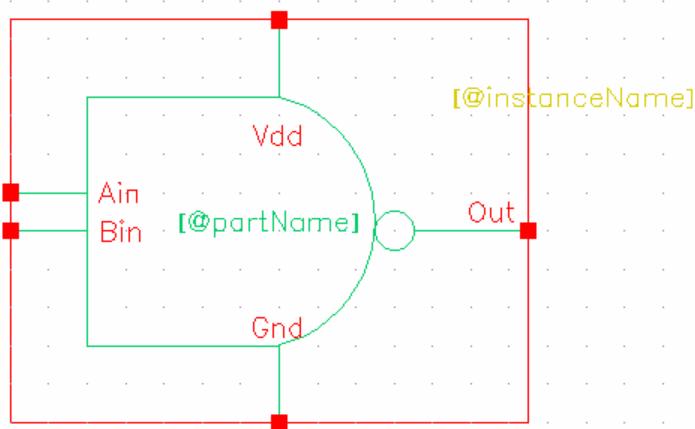


4. Click OK and the default box-shaped symbol view is created as shown below:



5. Move your cursor over the symbol, until entire green rectangle is highlighted (selected). Click **left** to select it.  
 6. Click the **delete** icon in the symbol window.  
 7. Execute **Create → Shape → Line/Arc/Circle** to draw the symbol shown in the final picture.  
 8. Move the labels to the desired location.  
 9. To save your edited symbol view to disk, click the Save icon in the symbol editor window. When you save the final version, make sure it is bug-free. As in the schematic entry, check the CDS.log window. It should display following message for the correctly designed symbol view:

“mylib nand2 symbol” saved.



10. Now we can instantiate this symbol to build other circuits.

### 1-5. Simulation of a the 2 input NAND Gate

Open a new schematic window in the *mylib* library and build the *nand2\_sim* design.

In the Command Interpreter Window (CIW) or Library manger execute **File → New → Cellview**.

1. Set up the create new file form as follows –

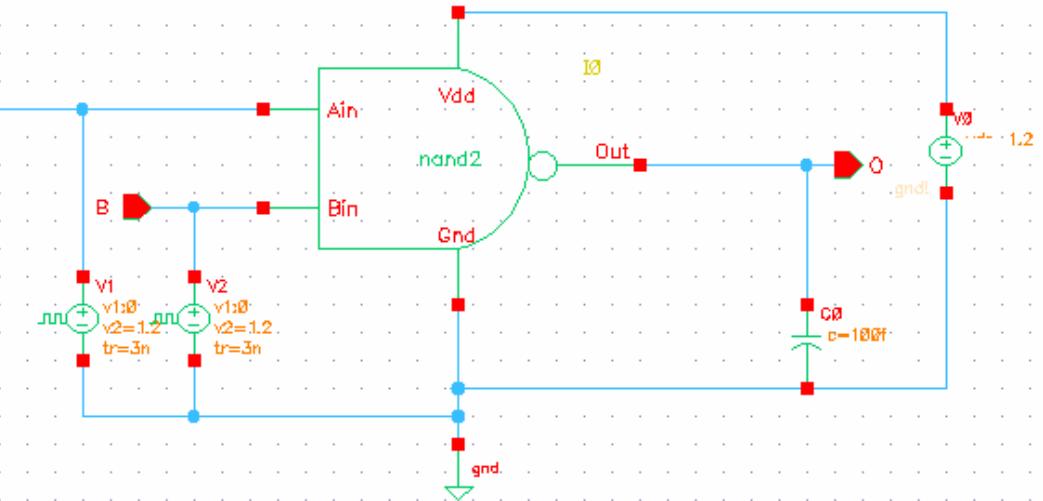
Library Name – mylib  
 Cell Name – nand2\_sim  
 View name – schematic  
 Open with – Schematic L

2. Click **Ok** when done.

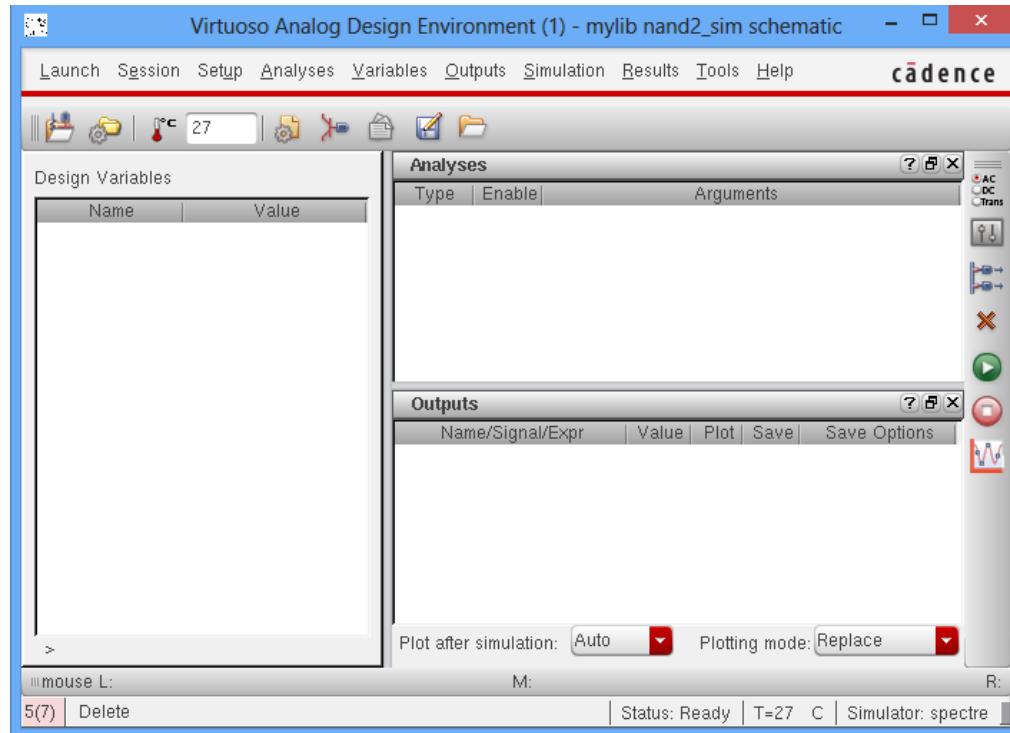
A blank schematic window for the *nand2\_sim* design appears. In the schematic window execute **Create → Instance**. Make sure that the View name field in the form is set to symbol. You will update the Library Name, Cell Name, and the property values given in the table below as you place each component. After you complete the Create instance form, move your cursor to the schematic window and click left to place the component. The *nand\_2* design contains the following cells from the following library.

Library Name	Cell Name	Properties
mylib	nand2	
analogLib	vdc	DC voltage = 1.2V
analogLib	gnd	
analogLib	vpulse	Voltage1 = 0V, Voltage 2 = 1.2V, Delay time = 3ns, Rise time = 3ns, Fall time = 3ns, *Period = 40ns/50ns, *Pulse width = 20ns/25ns
analogLib	cap	Capacitance = 0.1pF

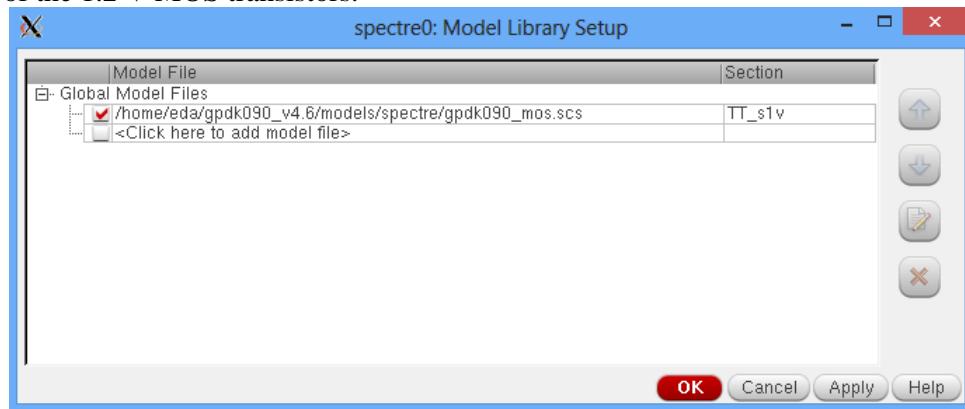
4. After entering the components, click Cancel in the Create Instance form or press Esc with your cursor in the schematic window.
5. Execute **Create → Pin**. Enter the **Pin Names A & B**, Select **Direction** to be input. Place the pins and then follow the same process to create and place a pin name of O with direction to be output.
6. Use **Create → Wire** to create the wire connections between pins and instances.
7. **File → Check and save** when your schematic looks like the following:



8. In the Schematic window execute **Launch → ADE L.** Analog Design Environment (ADE) window will open.

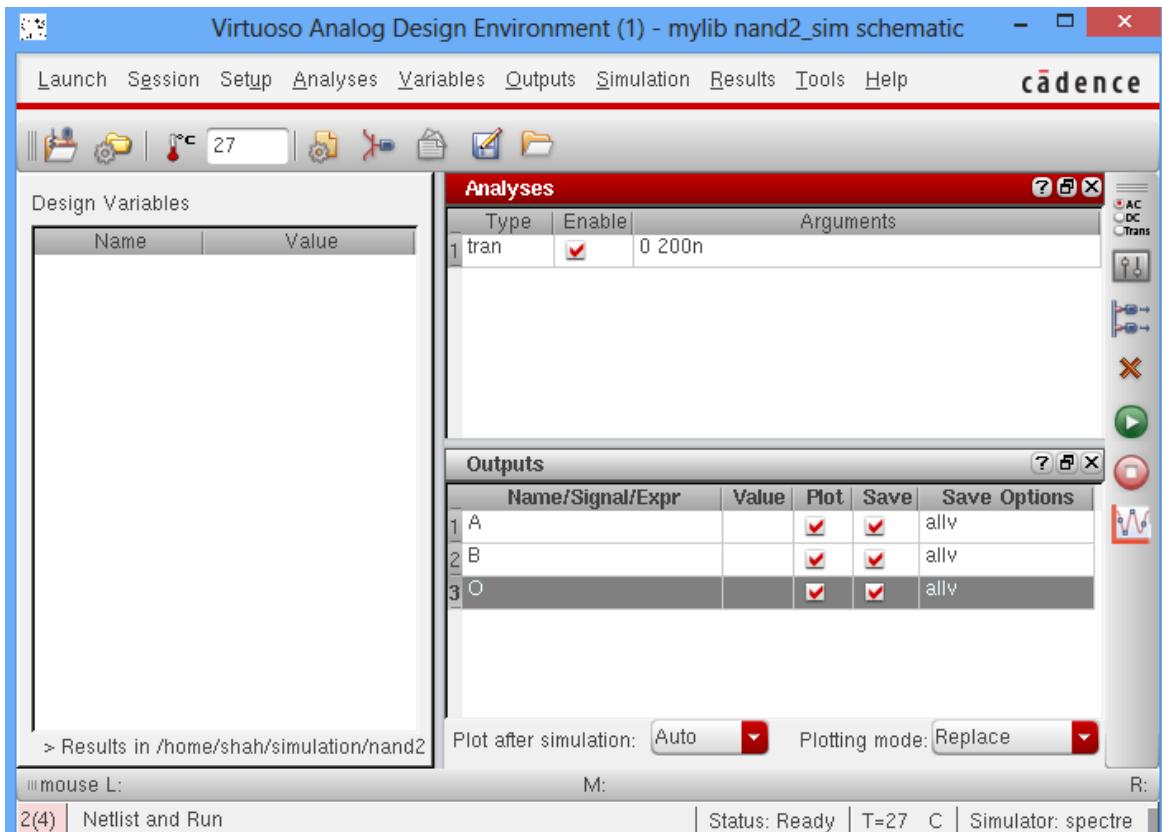


9. Set up the model libraries by executing *Setup → Model Libraries*. Select the following model library: gpdk090\_mos.scs. In this model library there are models to simulate various corners like fast-fast (FF), fast-slow (FS), typical-typical(TT) etc. We will choose the section typical from the section scroll bar and select the section 'TT\_s1v'. These will enable us to use the TT models of the 1.2 V MOS transistors.

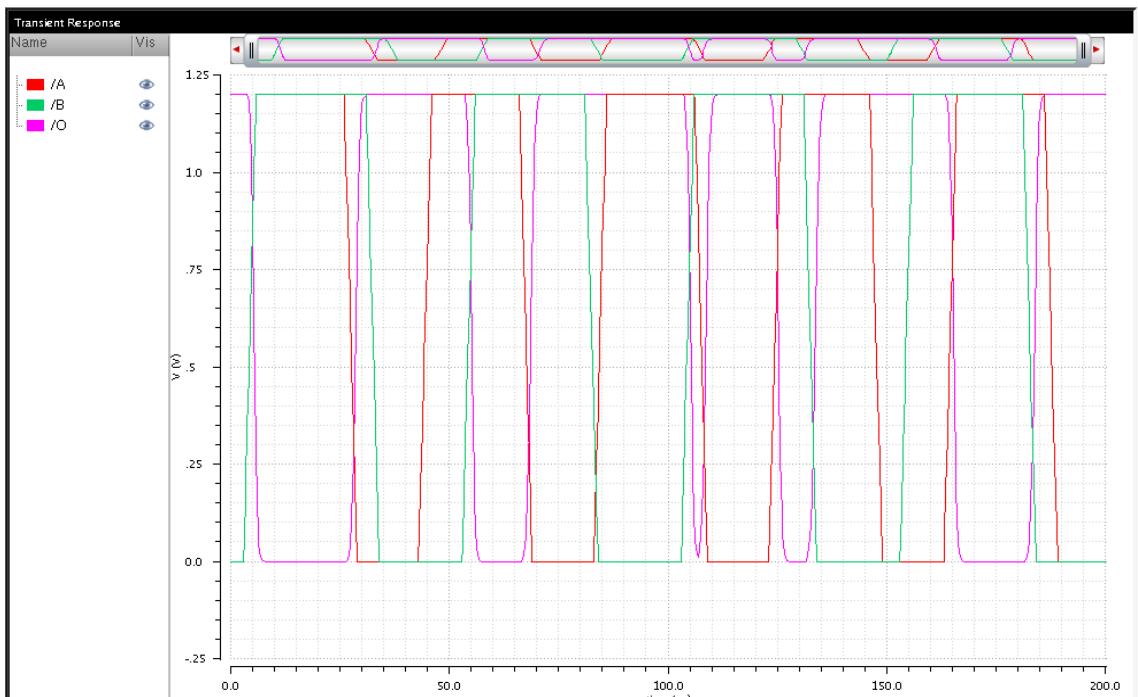


10. Now choose the analysis to be done by *Analysis → Choose*. Select transient analysis to be done. Provide stop time as 200ns.

11. Select the output to be plotted by executing *Outputs → To be plotted → Select on schematic* in the ADE window. Select A, B and O from the schematic to plot and save. The ADE window should appear to be something like the following:



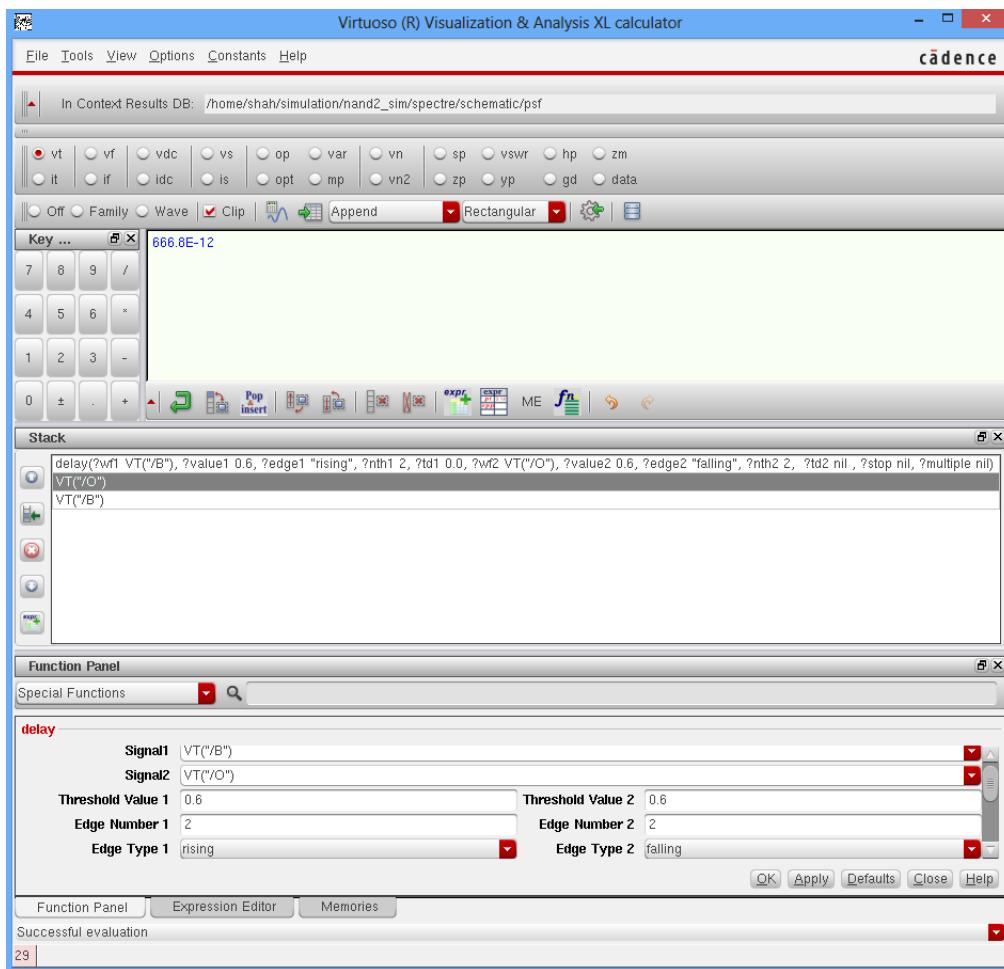
12. We will create the netlist and run the simulation by executing the following in the ADE environment ***Simulation → Netlist and Run***. A netlist file will be created. The netlist is saved in your *simulation* directory with name *input.scs*. The simulation will run and the output will appear as shown below. You can toggle the visibility of the plots by clicking on the eye icons under Vis column beside the Name column.



13. Next we measure the falling propagation delay for input port B. To start it we go to **Tools** → **Calculator**. The calculator window appears. Click on **vt**. Next select the pins B and O from the schematic. Next click on the **Enter** symbol. You will see the expression related to the transient voltages of the selected entities being stacked in the calculator.

14. Next select **delay** under the Special Functions tab in the Function Panel of the Calculator. The parameters corresponding to the delay function appears on the screen.

15. Clear the Signal 1 and Signal 2 lines if there is any expression already present on those lines. After that, Drag and drop **VT("B")** from stack or simply type it on the Signal 1 line. Next do the same for **VT("O")**. Enter the threshold value for both 1 and 2 to be 0.6. Choose Edge Number 1 and 2 to be 2. Choose rising for Edge Number 1 and falling for Edge Number 2. Click on **Apply** and then go to **Tools** → **Plot**. This will calculate the falling propagation delay for input port B.



**Department of Electrical & Electronic Engineering  
Chittagong University of Engineering & Technology**  
**EEE 476 VLSI Sessional**  
**Experiment No.: 02**  
**Layout design of a 2 input NAND Gate**

**Learning Objective:**

- To create a layout view of the basic two input NAND circuit from scratch

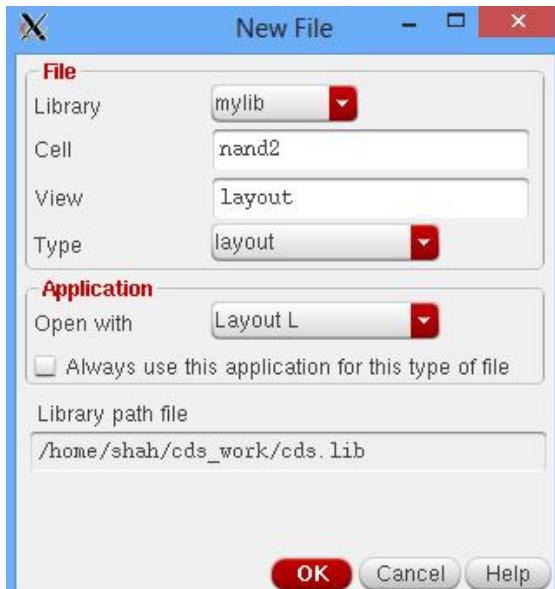
## **2-1. Setting up the Virtuoso Layout Suite L**

The VLSI design process goes through the following steps:

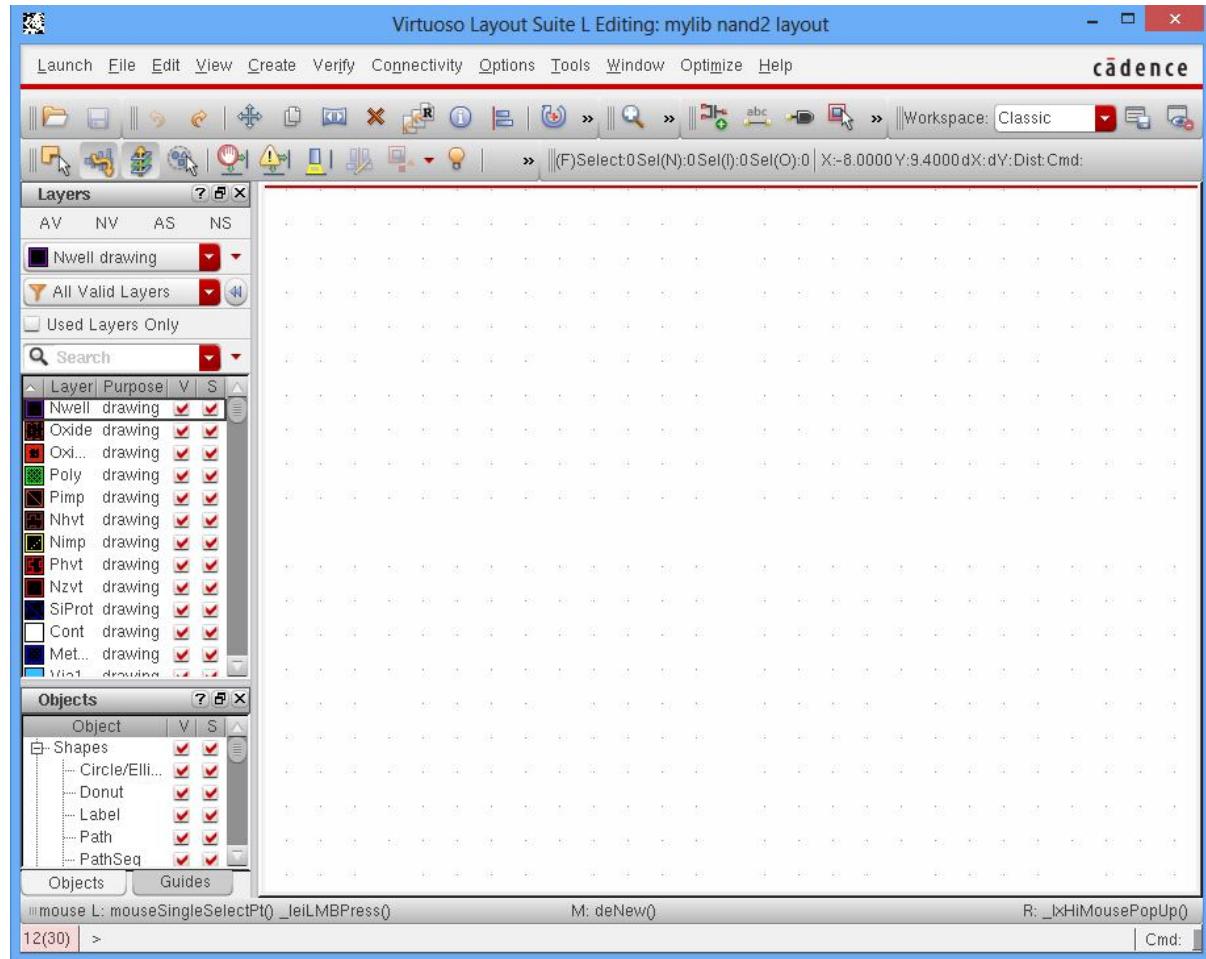
**Schematic Design → Schematic level Simulation → Layout creation → Design Rule Check → active device and parasitic Extraction → Layout level Simulation**

The tool for layout creation is called Virtuoso Layout Suite L. Before working with Layout editor you need to set the design layers. These information are stored in the **display.drf** file.

After the preparation is done invoke the Layout Suite L Editor from the CIW by executing **File → New → Cell View**. The new file form appears and fill it as shown in the figure below.



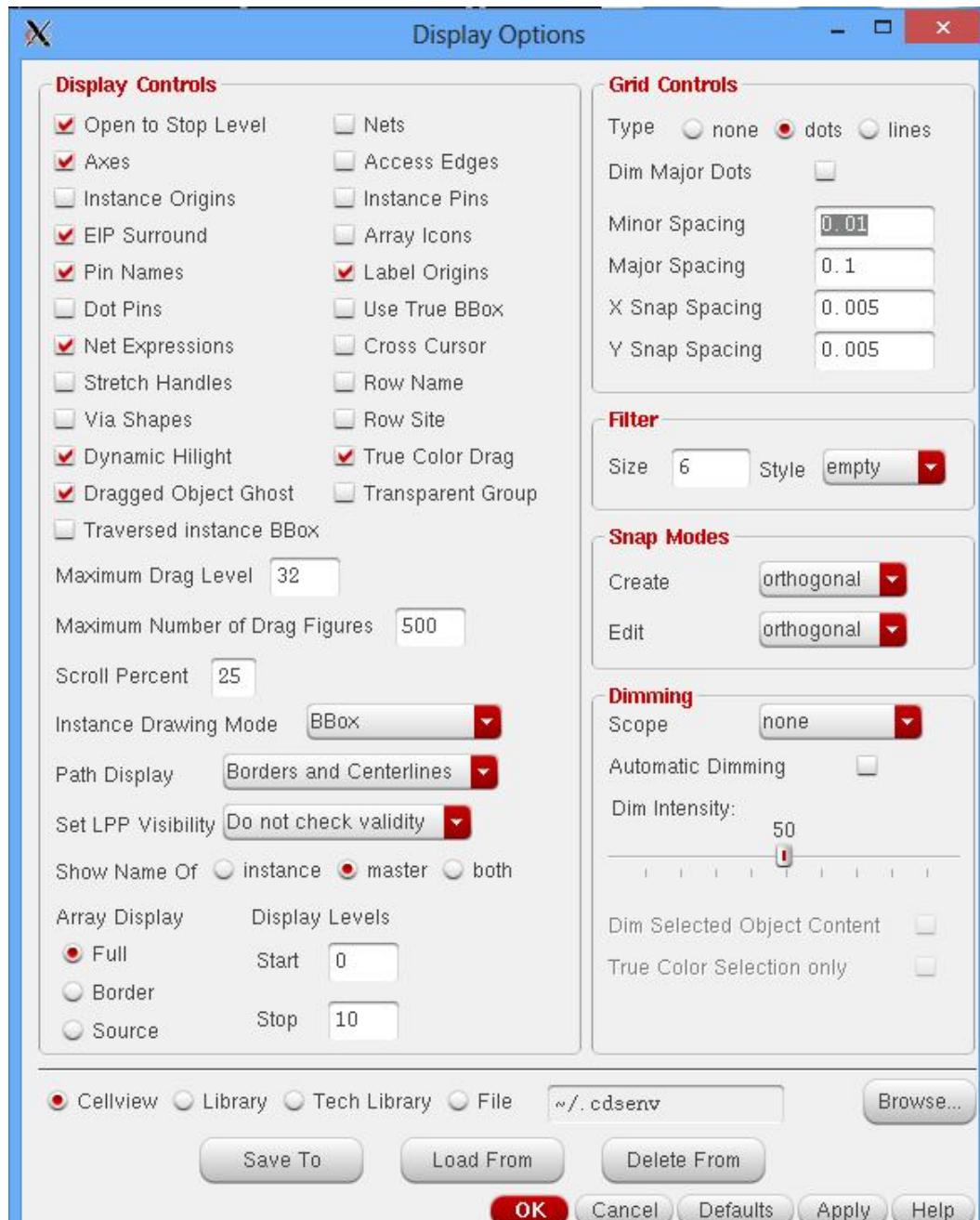
The Layers window is the one you will use to choose the different layers of the IC design.



The LSW window is divided in three main categories which are: layer color, layer name and layer purpose. The detailed is described in the table below:

Color	Matches the color in the Editing window. Each layer has its own color and pattern. Each layer has two colors associated with it; a fill color and an outline color.
Name	What is the type of layer (Nwell, Oxide, Poly, Metal1, etc)
Purpose	In gpdk090 the only purpose classification is drawing & slot Drawing is used in layout, slot is used to create a whole for metal stress relief

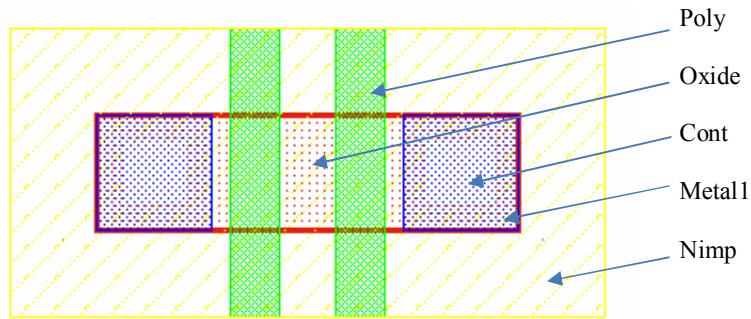
Before start layout, you need to set the layout configuration. Execute the following in the Virtuoso Layout Editor: **Options → Display**. Configure the form as shown in the figure below:



## 2-1. Building the layout of the CMOS inverter

Now we are going to build the layout of the NAND gate. A NAND gate has 2 PMOS transistors in parallel and 2 NMOS transistors in series. We will create NMOS and PMOS regions sequentially.

As seen from the layer diagram the NMOS consists of Oxide, Nimp, Cont and Poly layers. Study the rules of these layers.



**Fig: Series NMOS layout**

The rules related to the NMOS region can be summarised as follows:

Contact size: 0.12 umX 0.12 um (Fixed)

Poly width Minimum: 0.1 um (Fixed MOS gate length)

Contact to Poly spacing (Minimum): 0.1 um

Contact to Oxide spacing (Minimum): 0.06 um

Poly extending to Oxide (Minimum): 0.18 um

Nimp (Pimp as well)overlappingOxide (Minimum): 0.18 um

Minimum Metal 1 width: 0.12 um

Maximum Metal 1 width: 12.0 um

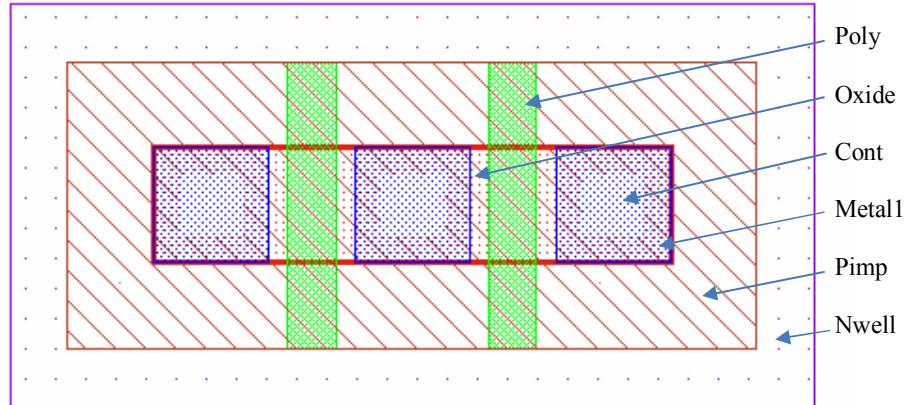
Minimum Metal 1 to Contact enclosure: 0.06 um

Poly to Poly spacing (Minimum): 0.12 um

Now we start building the NMOS region layout. Look at the Layers window and find the current drawing layer and follow the procedure described below:

1. To create the active area of the NMOS, left click the Cont layer and make it the current drawing layer. In the Layout editor window execute **Create → Shape → Rectangle**.
2. Draw the contact (.12u x .12u) and the surrounding Metal1 (.24u x .24u) layer. We will use copies of this one. So keep it untouched. And whenever required make a copy of it.
3. Select the Poly layer and draw the poly gate rectangle (0.1u x 0.6u). Make a copy of it at a distance of 0.12u. Maintain a distance between contact and poly gate of 0.12u.
4. Select the Oxide layer and draw oxide surrounding the contact (0.88u x 0.24u).
5. Select Nimp layer and draw a rectangle around the oxide (1.24u x 0.6u).

The PMOS transistor consists of Oxide, Poly, Pimp, Cont and Nwell layer. Study the rules of these layers.



**Fig: Parallel PMOS layout**

The rules related to PMOS are same as NMOS except the there is an additional layer the Nwell, whose rules are as follows:

Minimum Nwell width: 0.6 um

Minimum Nwell spacing to Nwell (same potential): 0.6 um

Minimum Nwell spacing to Nwell (different potential): 1.2 um

Minimum Nwell spacing to N+ active area: 0.3 um

Minimum Nwell spacing to P+ active area: 0.3 um

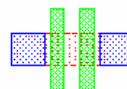
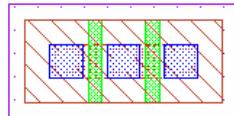
Minimum Nwell enclosure to P+ active area: 0.12 um

Minimum Nwellenclosure to N+ active area: 0.12 um

Minimum N+ Active Area to P+ Active Area Spacing: 0.15 um

Now we start building the PMOS region layout. Look at the Layers window and find the current drawing layer.

6. We build the PMOS region layout the same way as NMOS region just replace Nimp by Pimp layer. Note: Oxide dimension is now 1.02u x 0.24u & Pimp dimension is now 1.44u x 0.6u.
7. Additionally, select Nwell and draw nwell region around the Pimp region.
8. Next we restrict our space to a constant height so that other circuit layouts could be added without modifying ours. We can do this by drawing two Metal2 layers at 5um apart. These will work as Vdd and Gnd bus. This is shown in the following figure:



9. Next we construct 6 12um x 12 um Metal1 box to emulate IO Pads.



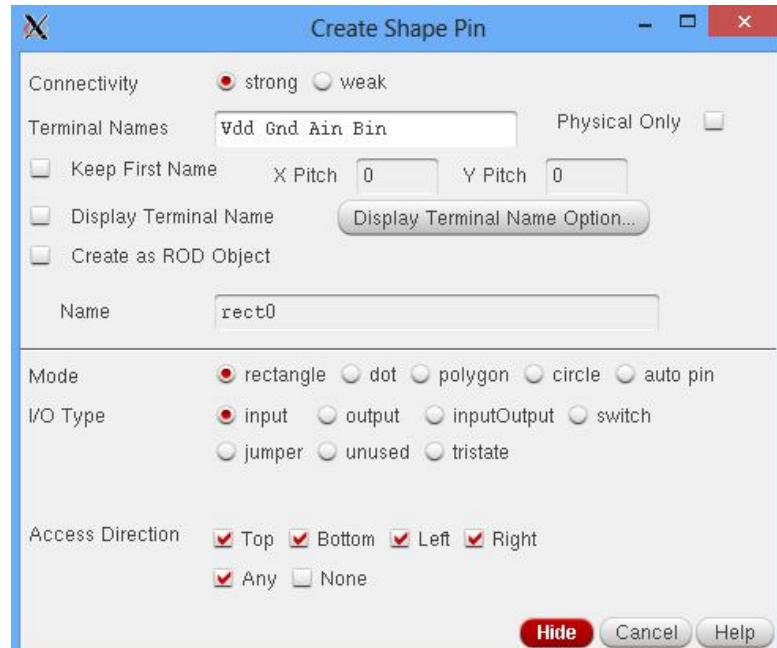
=



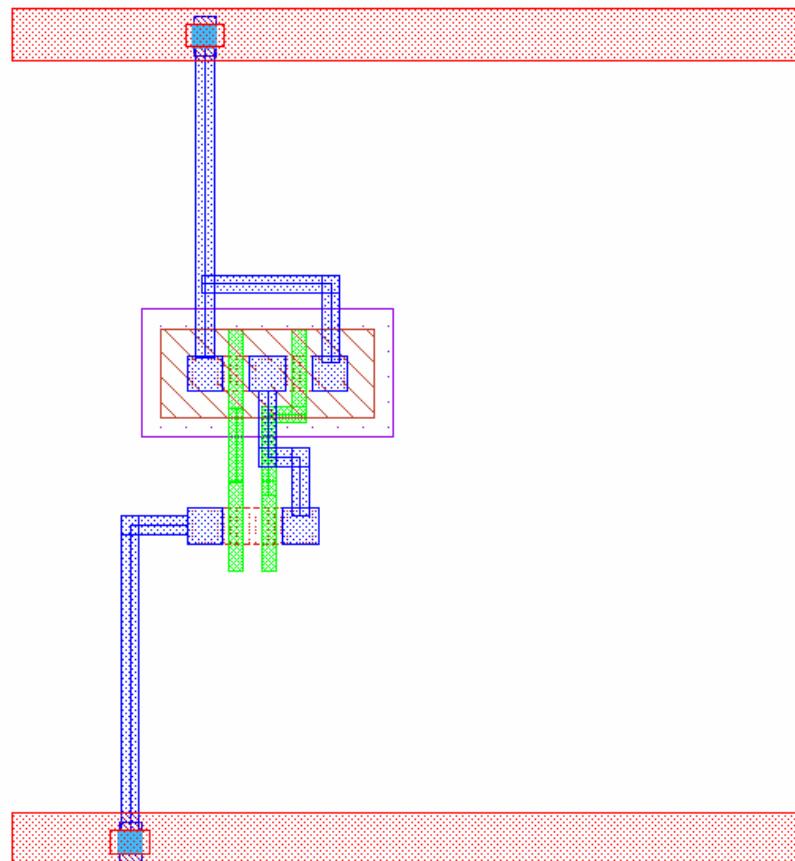
10. Finally, define Metal1 pins for Input and Output pins. In the layout editor window execute the following :**Create → Pin**. Make sure that you are selecting the **Metal1** in LSW.

11. To create the Vdd Gnd Ain and Bin input pins fill in **Create shape pin** Form as shown below and place a small rectangle on the desired pads in the same sequence.

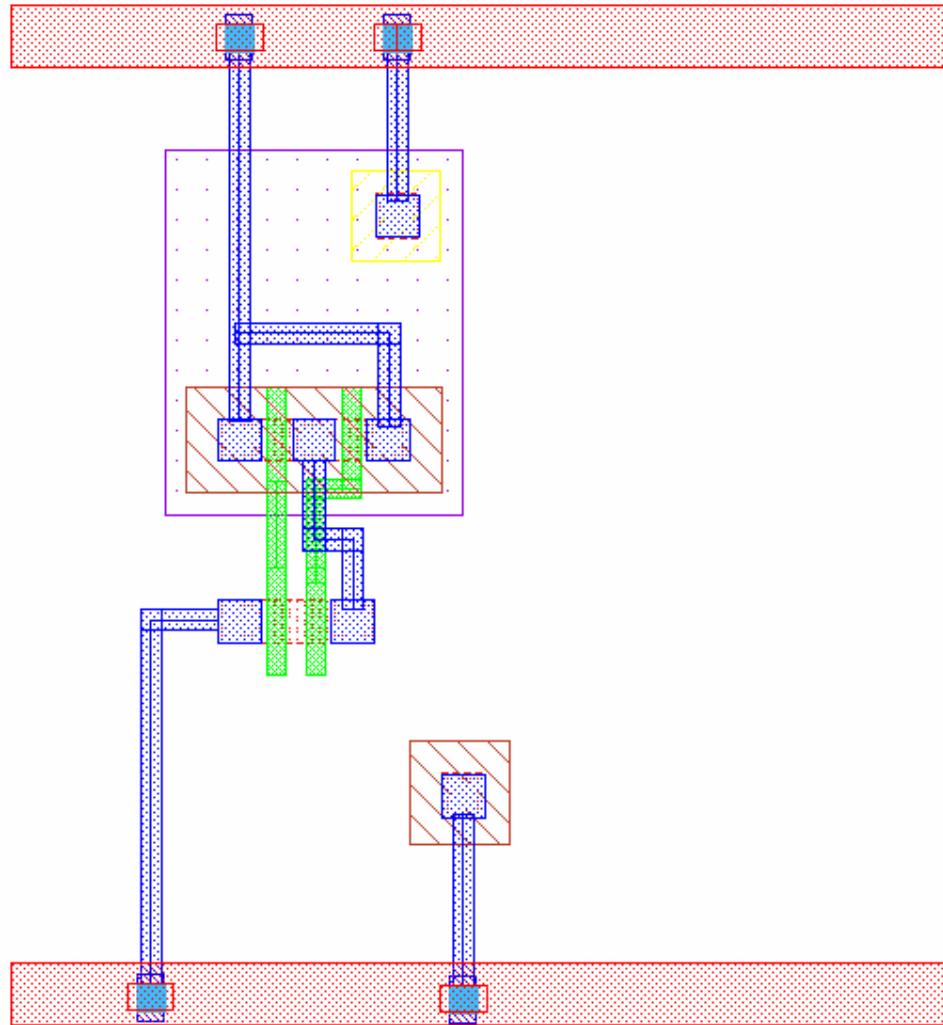
12. In a similar way create Out pin (select I/O type output).



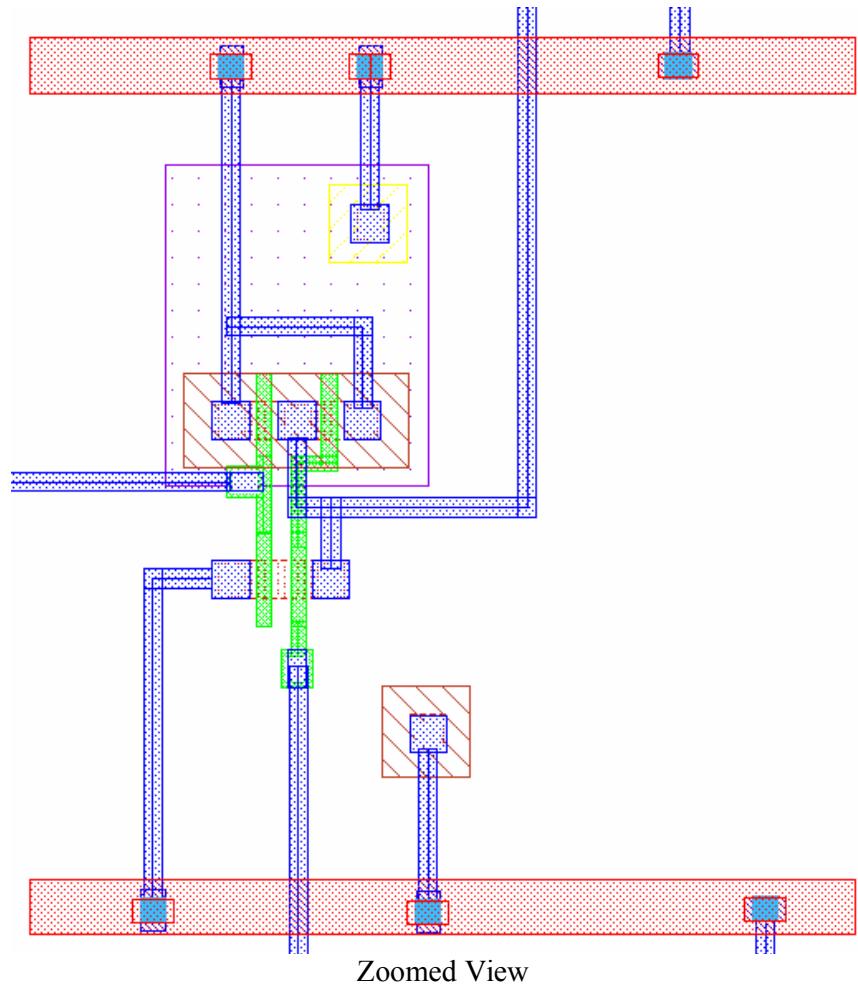
13. Now place PMOS region and NMOS region as shown earlier and right click the mouse then click **Create ->Wires** to create interconnects between poly gates, contacts, Vdd and Gnd as the image shown below. You need to click **Create ->Via** to create connections between Metal1 and Metal2 at Vdd and Gnd.



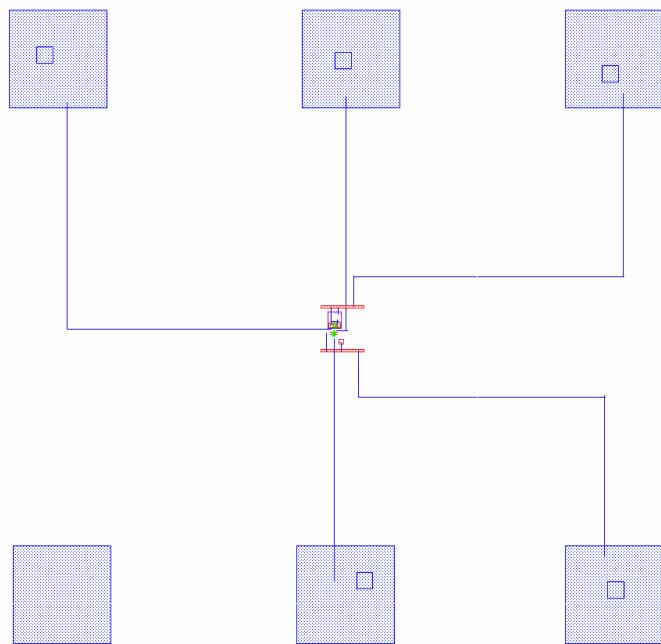
14. Next we need to bias the p-substrate and n-well by creating Pimp and Nimp layer respectively over those regions and providing oxide and contact. Finally connecting them with Gnd and Vdd respectively. After this step your layout should look like the following figure. Note that minimum oxide area must be 0.06 $\mu$  and hence a dimension of 0.25 $\mu$  x 0.24 $\mu$  would be suitable for providing the biasing.



15. Finally we need to create Metal2 to Metal1 Via and connect Vdd and Gnd bus with IO Pads. Furthermore we need to create Contacts for connecting the poly gates to the different IO Pads by Metal1 layer. Additionally, we need to connect the output node Metal1 layer to another IO Pad. After all the connections the layout should look like the following figure. At the end of these steps click **File ->Save**.



Zoomed View



Overall View

## **Report**

Follow standard template of EEE 458 lab report and include the following also:

1. Show the print out of the layout. Measure its size. Could you achieve minimum sized layout?
2. What types of error did you received? What are the meanings of the error?
3. Describe some good practices for 2 input NAND gate layout.

**EEE 476 VLSI Sessional  
Experiment No.: 03  
DRC, LVS and Post Layout Simulation**

**Objectives:**

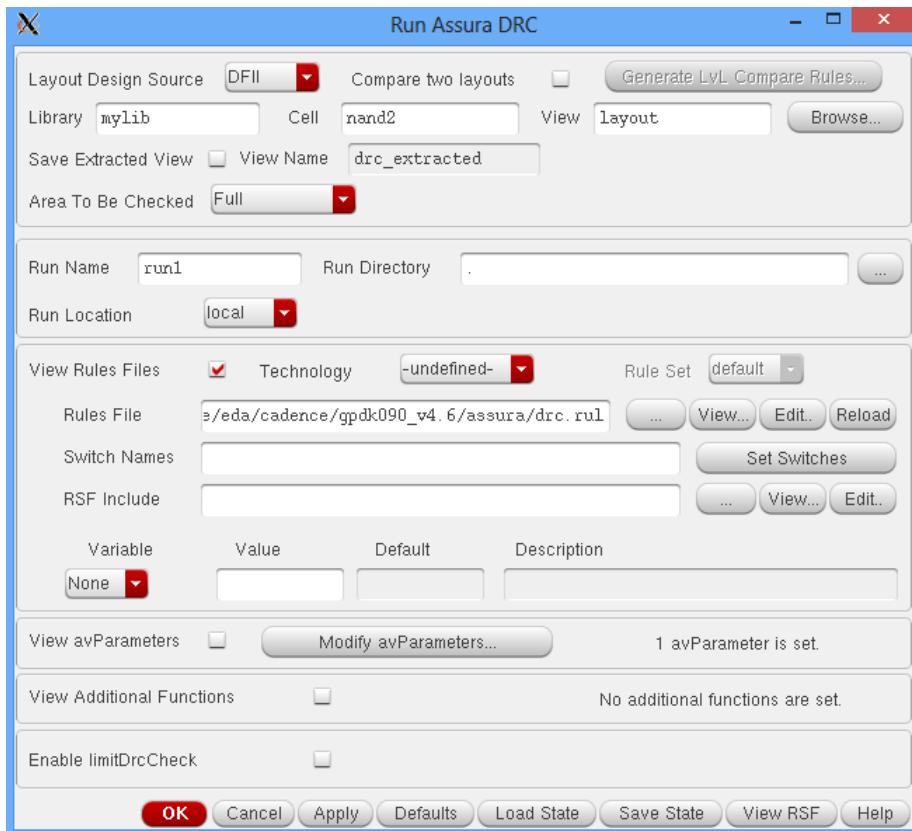
- To perform the design rule check of the NAND gate layout
- To perform the layout vs. schematic check for the NAND gate layout
- To perform parasitic extraction on the NAND gate layout
- To simulate the extracted view
- To perform tape out

**3-1. DRC Rules Check by Cadence's ASUURA**

1. In Virtuoso Layout Suite L window click Assura → Run DRC. Alternately In the same shell window (if you typed & after virtuoso) execute the command `avview`. An Assura Window appears as below. Run the DRC by clicking DRC → Run.



2. A DRC window appears as shown below. Fill the form as indicated in the picture.

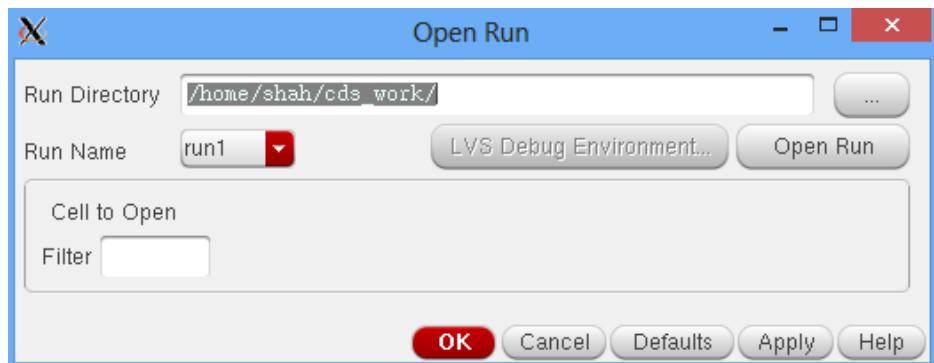


Select the Rules File from the path: "/home/eda/cadence/gpdk090\_v4.6/assura/drc.rul"

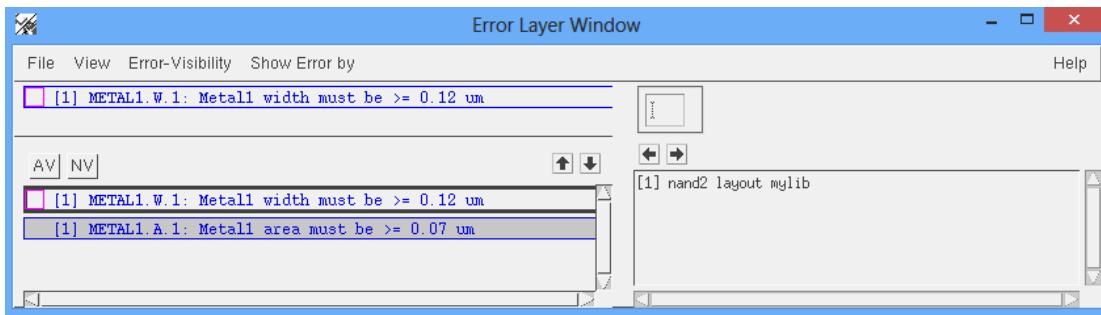
3. Click on the OK button. A DRC completed window appears as shown below:



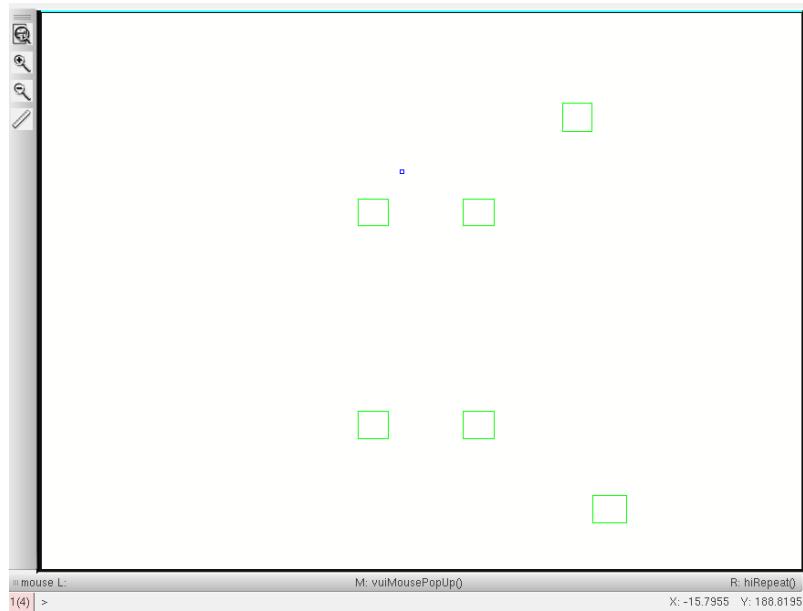
4. Click Yes. An Open Run window appears. Click Open Run.

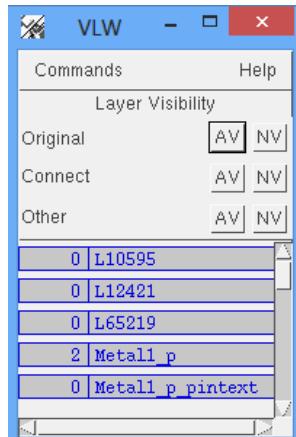


5. Cell name appears. Select the cell and press OK.
6. Error layer appear as shown below with a nand2 layout window which shows the error



7. Click to the errors and correct them. The Error Layer Window controls pointing out the error locations whereas VLW windows controls displaying different layers. Thus we can easily identify the errors and correct them from layout editor. The GPDK reference manual can be consulted for details on the DRC rules and errors.





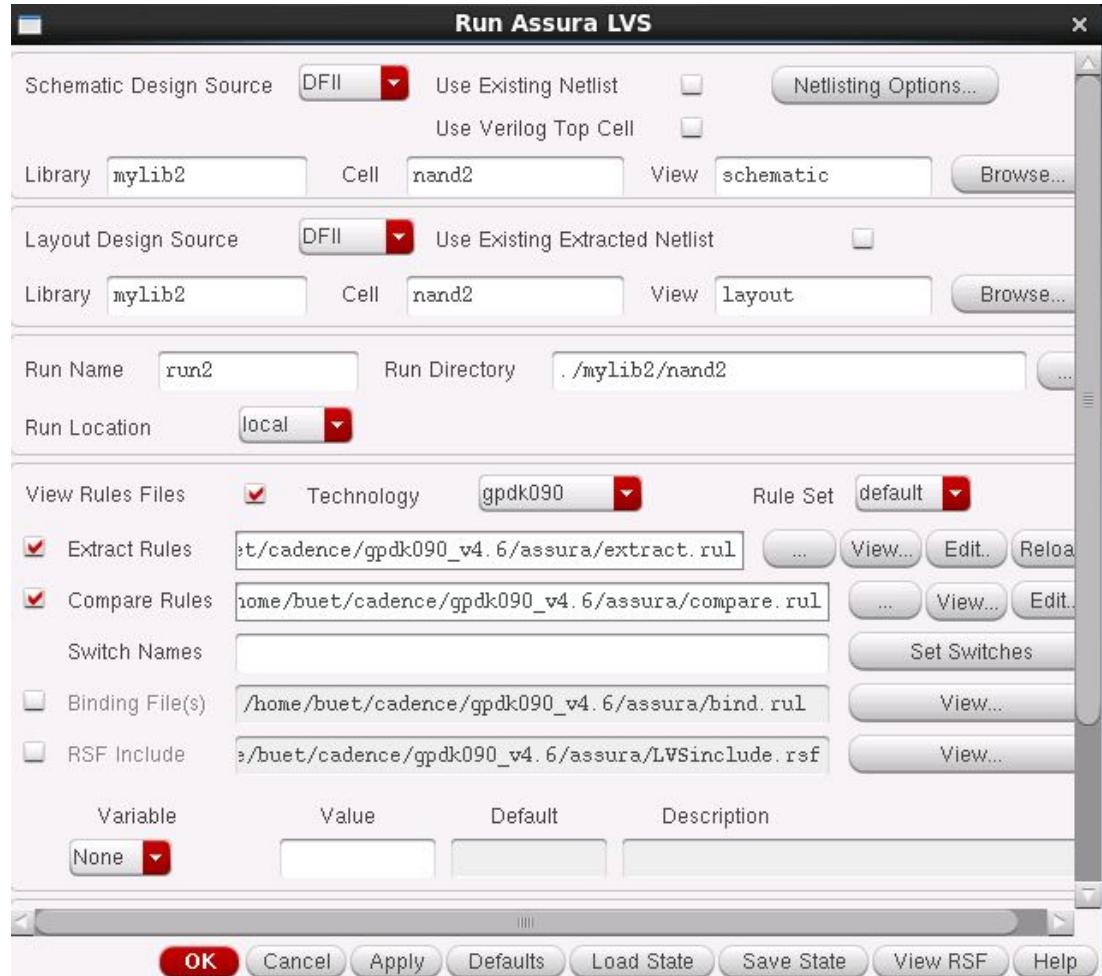
8. After correcting errors we have follow the same procedure until all errors are corrected.
9. When there will be no errors the following appears:



### **3-2. Layout versus Schematics (LVS) Check using Assura**

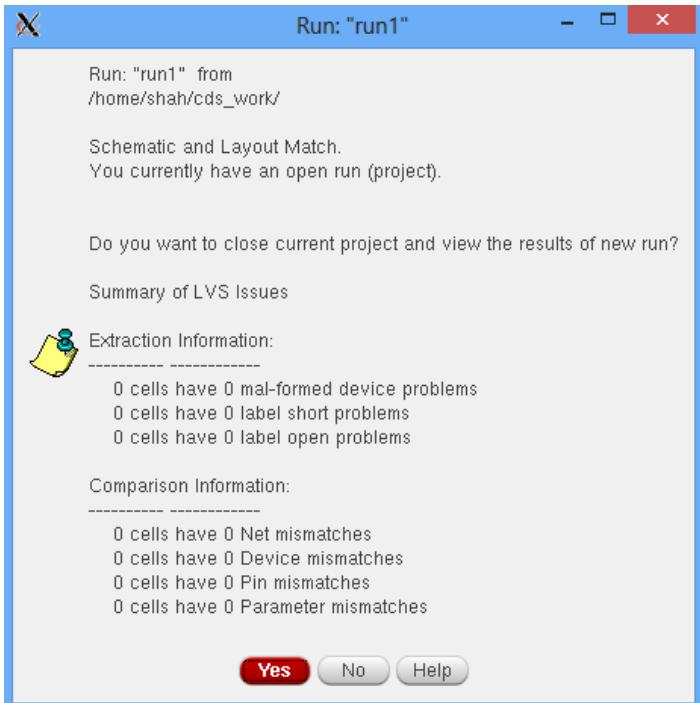
We still need to verify that the layout we designed in Lab2 matches the schematic created in Lab 1. This verification is accomplished by checking Layout Versus Schematic (LVS) rules in Assura.

1. In avview assura window, Run the LVS by clicking LVS → Run or in layout window using Assura → Run LVS. A pop-up window will appear, fill it up as follows:



Select the Extract Rules File from the path: “/home/eda/Cadence/gpdk090\_v4.6/assura/extract.rul” and the Compare Rules File from the path: “/home/eda/Cadence/gpdk090\_v4.6/assura/compare.rul”

2. If there is no error, after the LVS run is finished, you should see the following pop-up window:



(Otherwise correct your layout/schematic accordingly)

3. Click No.

### **3-3. Parasitic Extraction using Assura**

---

1. In the avview Assura window click on **Commands ->Show Setup Functions -> Technology**. The following window will appear.

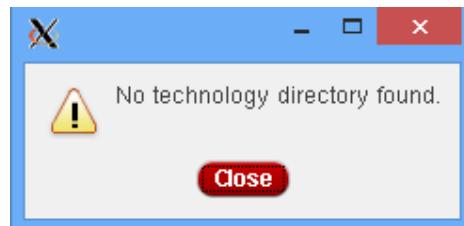


Select the Assura Technology File from the path:

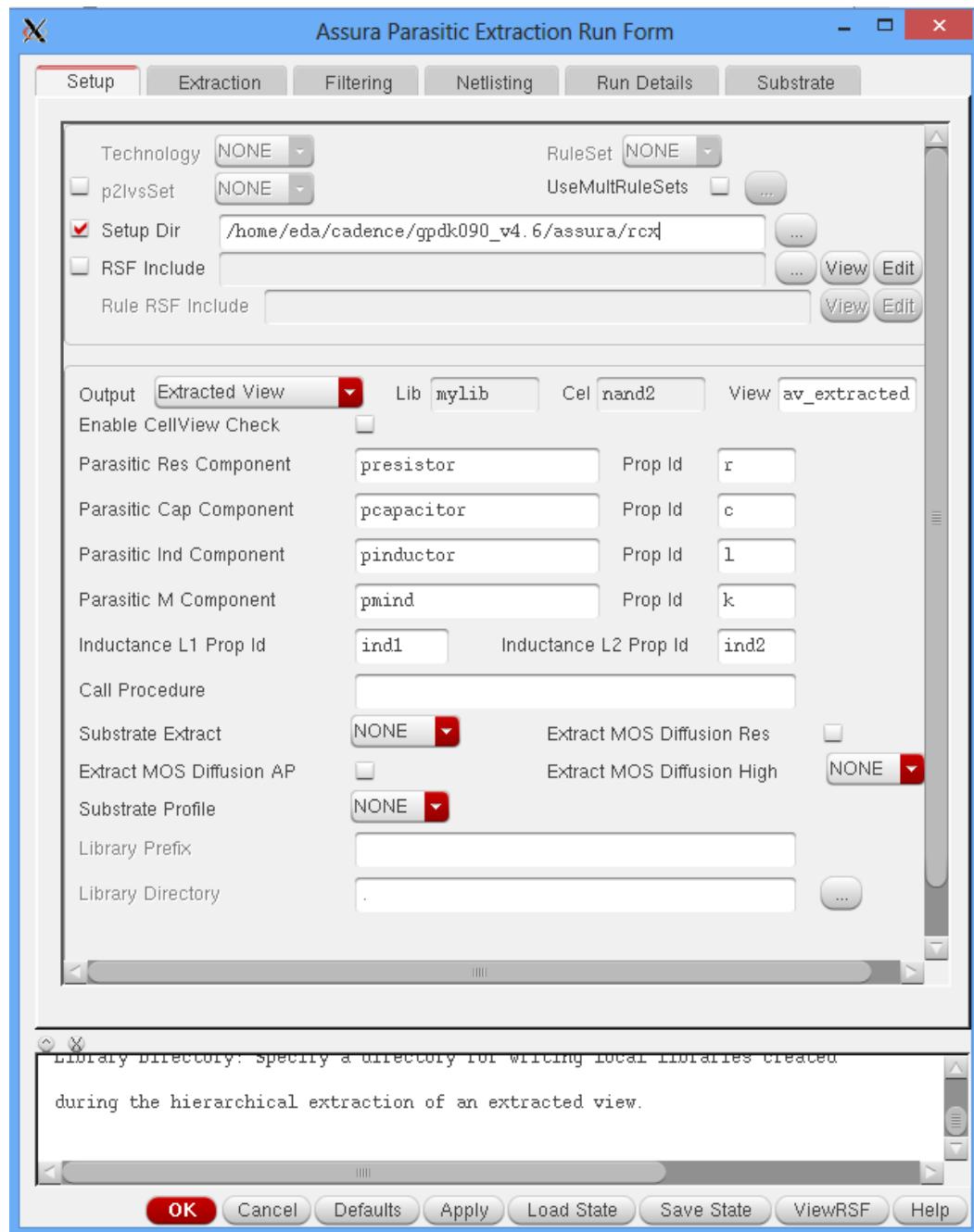
"/home/eda/cadence/gpdk090\_v4.6/assura\_tech.lib"

Else in layout window execute Assura → Technology and confirm that assura technology file is defined in ./assura\_tech.lib i.e. in the assura\_tech.lib in the current directory. Now click view and confirm the content of the assura\_tech.lib file define gpdk090 as /home/eda/Cadence/gpdk00\_v4.6/assura

2. Click **Open Run... ->Open Run**. After this you will see the **RCX -> Run** menu in Assura window become executable.
3. Now click **Run RCX** the following appears:

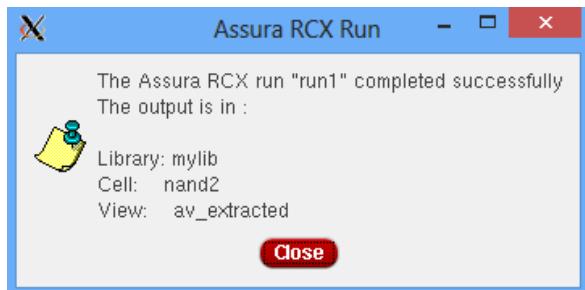


4. Click on **Close** and Assura Parasitic Extraction Run Form will appear. This will show Technology field grey with NONE. On the Setup Tab, choose Extracted View for Output. Make sure Setup Dir is: “/home/eda/cadence/gpdk090\_v4.6/assura/rcx”.



5. On the Extraction Tab, choose RC as the Extraction Type and Schematic Names as the Name Space. Also type gnd! as the reference node. Click OK.

- When the Assura run is complete, the following window pops-up. Click Close.



### **3-4. Post Layout Simulation**

---

- In the CIW, execute **File → Open**
- Set up the OpenFile form as follows –
  - Library Name – mylib
  - Cell Name – nand2
  - View name – av\_extracted
  - Open with – Layout L

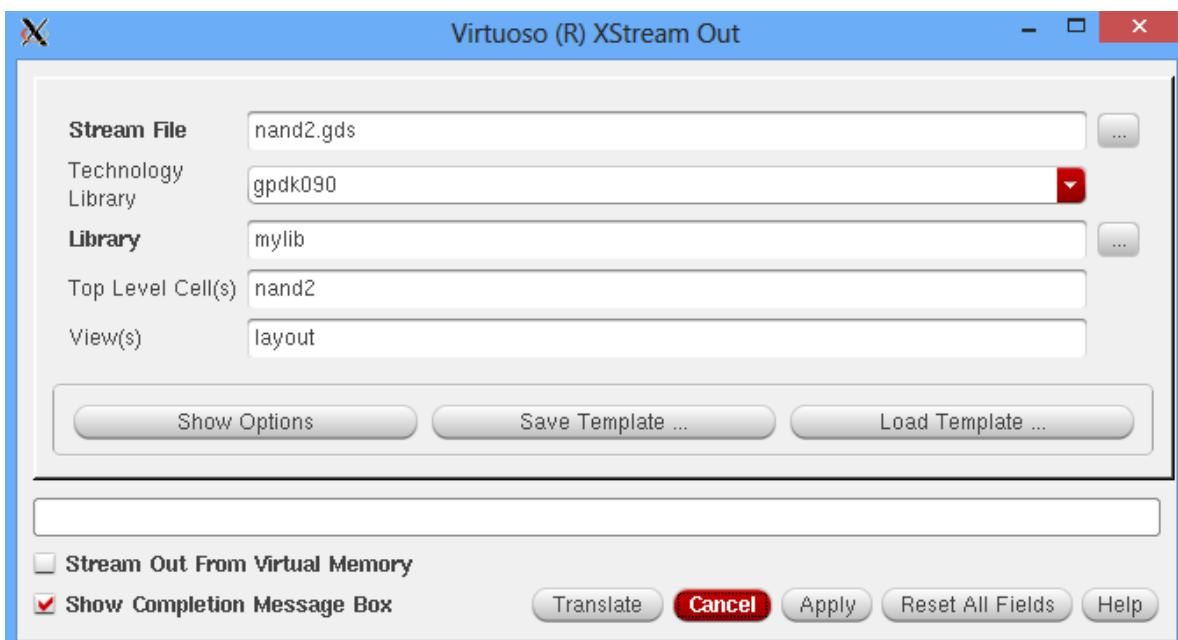
After the run you can verify that beside the 2 NMOS and 2 PMOS there area a lot of parasitic capacitor and resistor extracted. You may zoom in the layout to verify their location.

- In the Schematic window execute **Launch → ADE L**. Analog Design Environment (ADE) window will open.
- Setup the model libraries by executing **Setup → Model Libraries**. Select the following model library: gpdk090\_mos.scs, gpdk090\_capacitor.scs and gpdk090\_resistor.scs. Choose ‘TT\_s1v’ in the Section column.
- Setup the Stimuli by executing **Setup → Stimuli**.
- Click on the input **Vdd**. Click on **Enabled**. Choose Function **dc**. Enter DC Voltage as 1.2V. Click **Apply**.
- Click on the input **Gnd**. Click on **Enabled**. Choose Function **dc**. Enter DC Voltage as 0 V. Click **Apply**.
- Click on the input **Ain**. Click on **Enabled**. Choose Function **pulse**. Enter Voltage1, Voltage2, Period, Delay time, Rise time, Falltime and Pulse width as 0V, 1.2V, 40ns, 3ns, 3ns, 3ns and 20ns respectively. Click **Apply**.
- Click on the input **Bin**. Click on **Enabled**. Choose Function **pulse**. Enter Voltage1, Voltage2, Period, Delay time, Rise time, Fall time and Pulse width as 0V, 1.2V, 50ns, 3ns, 3ns, 3ns and 25ns respectively. Click **Apply**. Click **OK**.
- Now choose the analysis to be done by **Analysis → Choose**. Select transient analysis to be done. Provide stop time as 200ns.

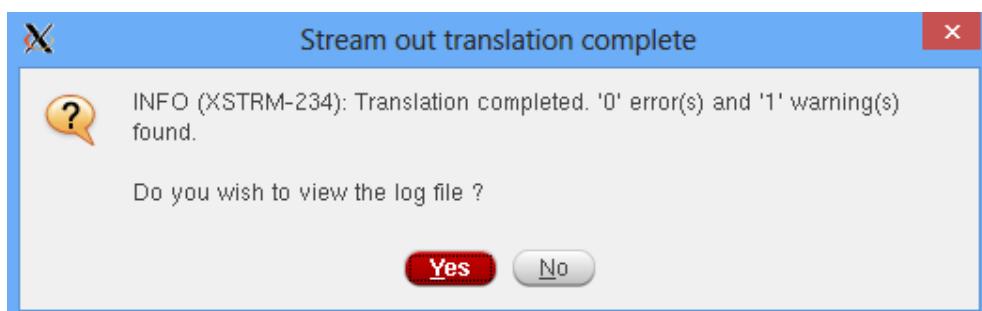
11. Select the output to be plotted by executing **Outputs** → **To be plotted** → **Select on schematic** in the ADE window. Select Ain, Bin and Out pins from layout to plot and save.
12. We will create the netlist and run the simulation by executing the following in the ADE environment **Simulation** → **Netlist and Run**.
13. The simulation will run and the output will appear. From the log file verify the number of MOS transistors, resistor and capacitors and number of nodes.
14. We can measure the relevant parameters by using Calculator tool and check whether they meet our requirements.

### **3-5. Tape Out**

1. Now that post layout simulation has been successfully, we proceed to tape out.
2. In the CIW click **File** → **Export** → **Stream**.
3. Fill out the XStream Out window as shown below:



4. Click on **Translate**.
5. A notification displaying Translation Completed appears.



**EEE 476 VLSI Sessional  
Experiment No.: 04**

**Design a Two Stage CMOS Operational Amplifier and Study of its DC and AC Characteristics Using Cadence Virtuoso**

---

**Objectives:**

- To design an un-buffered two stage CMOS Operational Amplifier with an n-channel input pair
- To study the large signal dc characteristics of the Op-Amp
- To study the small signal characteristics of the Op-Amp

**Lab 8-1. To write the specification of a two stage CMOS Op-Amp and to fix the transistor sizes and bias currents from the level 2 model parameters of the NMOS and PMOS transistors.**

---

The design procedures of the Two-stage CMOS Op-Amp are described in great details in the book : *CMOS Analog Circuit Design, Second Edition by Philips E. Allen and Douglas R. Holberg, OXFORD University Press*. You are required to study section 6.3 *Design procedure for the Two-stage CMOS Op-Amps (Page 269-280)* carefully and design the op-amp using typical NMOS and typical PMOS transistors of the Analog library provided by Cadence.

**Specification of the Op-Amp :**

The specification of the Op-Amp is given below

Av> 5000 V/V	VDD : 2.5 V	VSS : -2.5 V
GB : 5 MHz	C <sub>L</sub> : 10 pF	SR > 10 V/us
V <sub>out</sub> range : ± 2V	ICMR : -1 to 2 V	P <sub>diss</sub> ≤ 2 mW

The model files of the typical NMOS and PMOS transistors are in the following location:  
`/home/eda/Cadence/IC615/tools.lnx86/dfII/samples/artist/models/spectre/cornerMos_mod.scs`  
We will use the typical NMOS and typical PMOS section of the model (Section TNTP)

```

simulator lang=spice
* VTI-derived Level=2 nominal model
.model nmos4 nmos level=2 vto = 0.775 gamma=0.4 tox = 400e-10 nsub = 8e+15 xj = 0.15U ld =
0.20U u0 = 650 ucrit = 0.62e+5 uexp = 0.125 vmax = 5.1e+4 neff = 4.0 delta = 1.4 rsh = 36 cgso =
1.95e-10 cgdo = 1.95e-10 cj = 195U cjsw = 500P mj = 0.76 mjsw = 0.30 pb = 0.8

.model pmos4 pmos level=2 vto = -0.75 gamma=0.57 tox = 400e-10 nsub = 6e+15 xj = 0.05U ld =
0.20U u0 = 255 ucrit = 0.86e+5 uexp = 0.29 vmax = 3.0e+4 neff = 2.65 delta = 1.0 rsh = 101 cgso =
1.90e-10 cgdo = 1.90e-10 cj = 250U cjsw = 350P mj = 0.535 mjsw = 0.34 pb = 0.8

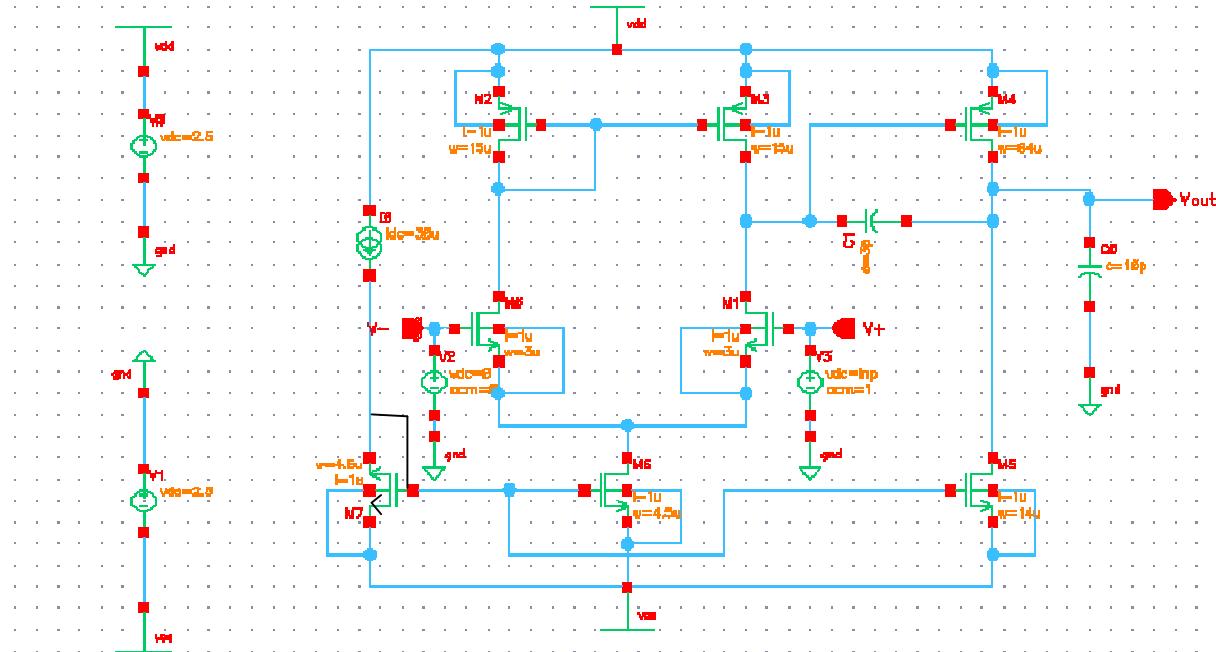
```

**PRE\_LAB : From the model parameters and with reference to *Figure 6.3-2 of the Book CMOS Analog Circuit Design by Philips E Allen, and Douglas R. Holberg* design the various components of the Op-Amp and fill up the table below. Please fill up the form before coming to the Lab for this experiment. You will not be allowed to proceed further without completing the PRE-LAB.**

Parameters	Equation/Criteria used to fix value	Design value
Compensation capacitor Cc		
Steering current I <sub>5</sub>		
Current Mirror Transistor size (W/L) <sub>3</sub> and (W/L) <sub>4</sub>		
g <sub>m1</sub>		
Input Pair Transistor size (W/L) <sub>1</sub> and (W/L) <sub>2</sub>		
V <sub>DS5</sub>		
Current source transistor size (W/L) <sub>5</sub> , (W/L) <sub>8</sub>		
g <sub>m6</sub>		
Output stage load transistor size (W/L) <sub>6</sub>		
Output stage drive transistor size (W/L) <sub>7</sub>		

## Lab 8-2. Design of the Two Stage CMOS Operational Amplifier

The following figure show the Schematic diagram of the two stage Op amp. Take nmos4, pmos4 etc. from analog lib. Please note that the transistor dimension of your circuit may be different from that shown in the figure below.



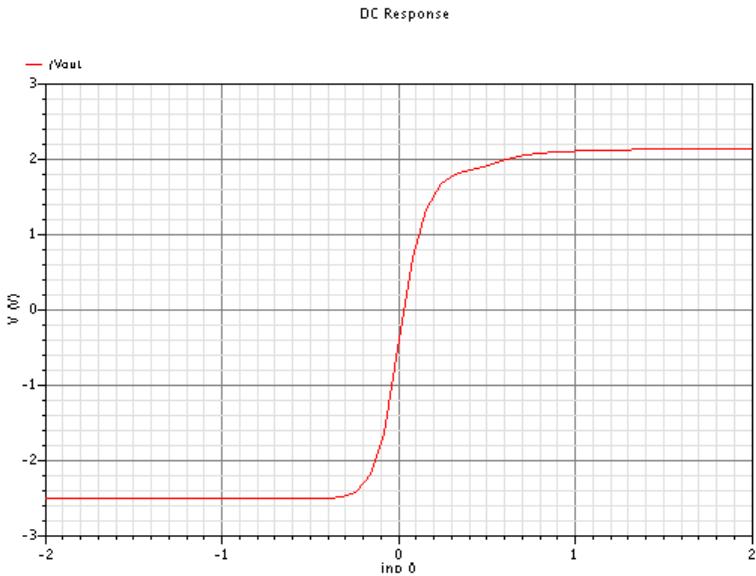
**Fig. Schematic of the two stage Op-Amp (Change the figure)**

### 1. Simulate the transfer characteristics of the Op-Amp.

Find the transfer characteristics of the Op-amp and determine the output offset voltage of the amplifier. For this purpose set the voltage of the  $V+$  terminal as a variable and then make a coarse sweep of this variable form -2 Volts to +2 Volts and run a dc analysis to find the values of  $V_{in}$  where the output makes the transition from  $VDD$  to  $VSS$ . Once the transition range is found,  $V_{in}$  is swept over values that only include the transition region. Use the following voltage specification for the  $V+$  and the  $V-$  terminal.

$V+$	DC voltage : inp, AC magnitude : 1 V
$V-$	DC voltage: 0 , AC magnitude: 0 V

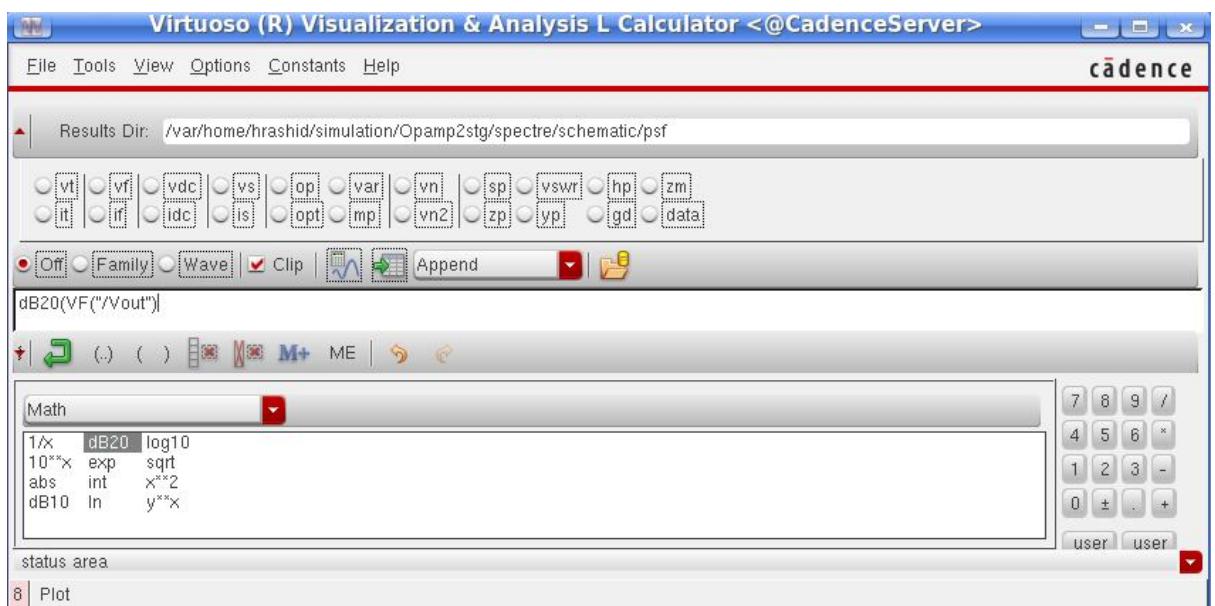
From the transfer characteristics curve determine the output off-set voltage as well as the input off-set voltage.



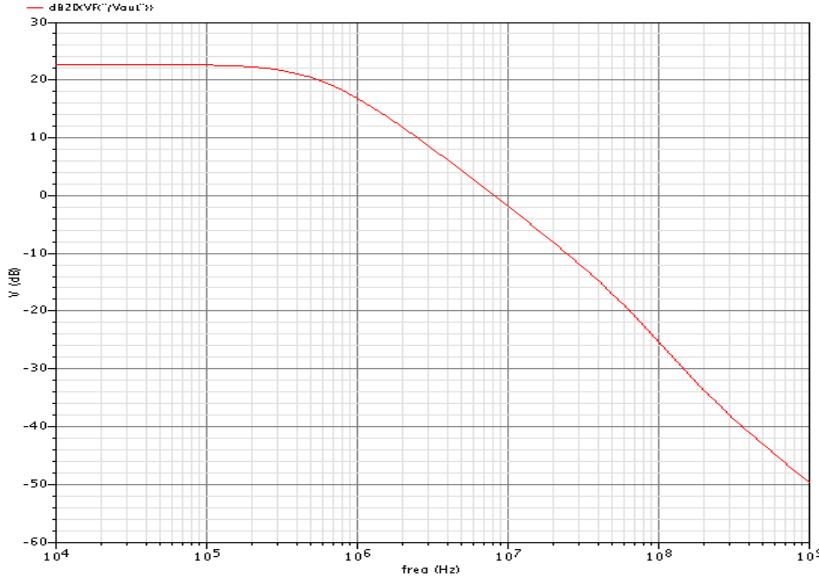
**Fig. Transfer characteristics of the Op-Amp**

## 2. Frequency response of the Op-AMP

For this purpose we will perform the ac analysis in spectre. In the ac sweep use *frequency* as a *sweep variable*, and vary the frequency from 10 kHz to 1 GHz. Set *sweep type* as *automatic*. Now a plot of output voltage magnitude in V versus frequency in Hz will be plotted out. However, we want to plot the output voltage in dB. For this purpose in the graph window execute *Tools* → *Calculator*. The Virtuoso Visualization and Analysis Calculator window will appear as follows. In the calculator window execute *Options* → *Algebraic*. Now double click the dB20 function and it will appear in the evaluation window.



Click the *vf* variable to select the ac voltage expression. The cursor will automatically move to the virtuoso schematic window. Now select the voltage *Vout* in the schematic window and it will be automatically loaded to the expression. You need to close the bracket of the expression manually. Now the expression is ready to be plotted. In the calculator window execute **Tools → plot**. The output voltage *Vout* in dB versus frequency curve will be plotted.

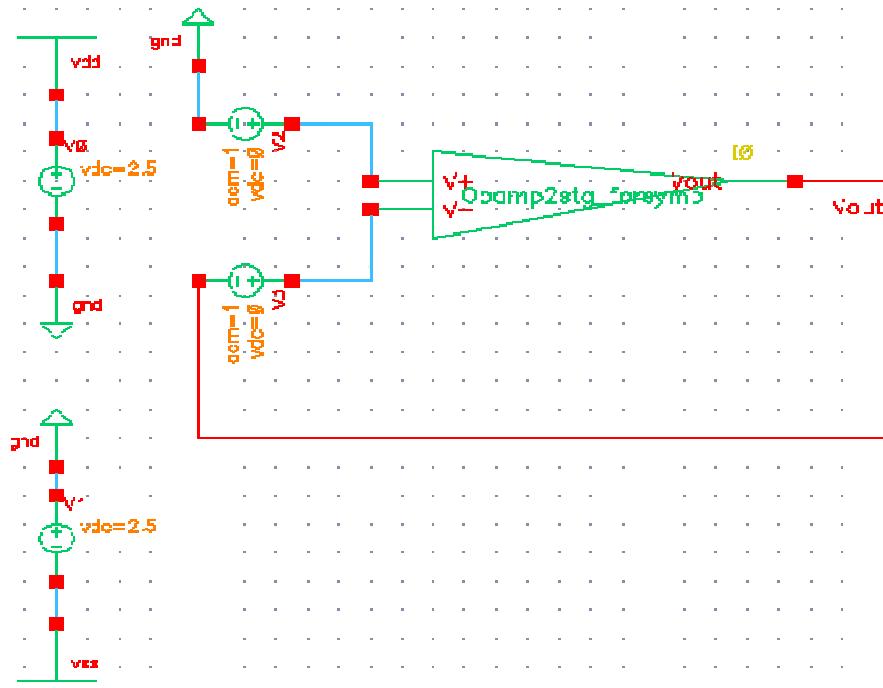


**Fig. Frequency response of Output Voltage**

From the graph determine the location of the dominant pole and the gain band width (GB) of the amplifier. Also determine the phase margin and the gain margin of the amplifier from the calculator. You can find the *gainBWProd*, *PhaseMargin* and the *GainMargin* in the special function of the calculator.

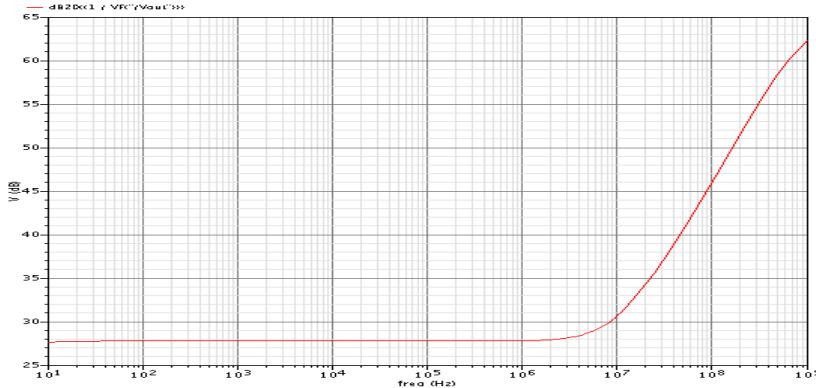
### 3. simulation of Common Mode Rejection Ratio

The objective of this simulation is to get an output that is equal to CMRR or can be related to CMRR. The following figure shows a method that can accomplish this objective. Two identical voltage sources designated as *Vcm* are placed in series with both op amp inputs where the op amp is connected in the unity gain configuration. To accomplish this first create a symbol of the Op-Amp and then connect it as shown below. For this circuit it can be shown that  $\frac{V_{out}}{V_{cm}} = \frac{1}{CMRR}$



**Fig. Set-up for the simulation of the CMRR**

The frequency response of  $1/\text{CMRR}$  is given in figure below.



**Fig. Frequency response of ( $1/\text{CMRR}$ )**

#### 4. Simulation of $\text{PSRR}^+$ and $\text{PSRR}^-$

Use similar circuits to measure the  $\text{PSRR}^+$  ( $\nabla V_{DD}/\nabla V_{out}$ ) and  $\text{PSRR}^-$  ( $\nabla V_{SS}/\nabla V_{out}$ )

#### Report :

1. Provide the circuit diagram and performance curves of the circuits
2. Compare in a table the design specifications with the simulated specifications.
3. Discuss about the discrepancy between the design specifications and the simulated specifications.
4. Simulate the design at fast fast and slow slow corner and compare the specification again.