VLSI DESIGN LABORATORY

LAB MANUAL

Course Code : AEC112

Regulations : IARE-6

Class : IV Year I Semester (ECE)

Department of Electronics and Communication Engineering

INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)
Dundigal – 500 043, Hyderabad

INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)
Dundigal, Hyderabad - 500 043

ELECTRONICS AND COMMUNICATION ENGINEERING

Program Outcomes	
PO1	**Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO2	**Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated RESULT s using first principles of mathematics, natural sciences,and engineering sciences.
PO3	**Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO4	**Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid RESULT s.
PO5	**Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
PO6	**The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
PO7	**Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO8	**Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO9	**Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary Settings.
PO10	**Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write

Program Outcomes	
	effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO11	**Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO12	**Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.
Program Specific Outcomes	
PSO 1	**Problem Solving**: Exploit the knowledge of high voltage engineering in collaboration with power systems in innovative, dynamic and challenging environment, for the research based team work.
PSO 2	**Professional Skills:** Identify the scientific theories, ideas, methodologies and the new cutting edge technologies in renewable energy engineering, and use this erudition in their professional development and gain sufficient competence to solve the current and future energy problems universally.
PSO 3	**Modern Tools in Electrical Engineering:** Comprehend the technologies like PLC, PMC, process controllers, transducers and HMI and design, install, test, maintain power systems and industrial applications.

INDEX

ATTAINMENT OF PROGRAM OUTCOMES & PROGRAM SPECIFIC OUTCOMES

Exp. No.	Experiment	Program Outcomes Attained	Program Specific Outcomes Attained
1	To plot the output characteristics, Transfer characteristics of an n-channel and p-channel MOSFET..	PO1, PO2	PSO1
2	To design and plot the static (VTC) and dynamic characteristics of a digital CMOS inverter.	PO1, PO2	PSO1
3	To design and plot the output characteristics of a 3-inverter ring oscillator	PO1, PO2	PSO1
4	To design and plot the dynamic characteristics of 2-input NAND, NOR, XOR and XNOR logic gates using CMOS technology.	PO1, PO2	PSO1, PSO2
5	To design and plot the characteristics of a 4x1 digital multiplexer using pass transistor logic.	PO1, PO2	PSO1
6	To design and plot the characteristics of a positive and negative latch based on multiplexers.	PO1, PO2, PO3	PSO1, PSO2
7	To design and plot the characteristics of a master-slave positive and negative edge triggered registers based on multiplexers.	PO1, PO2, PO3	PSO1
8	Design and simulation of a simple 5 transistor differential amplifier. Measure gain, ICMR, and CMRR.	PO1, PO2	PSO1
9	To design layout of NMOS and CMOS inverter.	PO1, PO2	PSO1
10	To design the layout of 2-input NAND, NOR gates.	PO1, PO2, PO3	PSO1
11	Analysis of Frequency response of Common source amplifiers.	PO1, PO2	PSO1
12	Analysis of Frequency response of Common drain amplifiers	PO1, PO2	PSO1
13	Design and Simulation of Single Stage Cascode Amplifier	PO1, PO2	PSO1, PSO2
14	Design and Simulation of Basic Current Mirror, Cascode Current Mirror Amplifier	PO1, PO2, PO3	PSO1

VLSI DESIGN LABORATORY

OBJECTIVE:

The objective of the VLSI DESIGN LAB is to expose the students to the circuit design of analog and digital circuit using Cadence Virtuoso tools. It also aims to understand how to measure different performance parameters of the circuits, Create some innovative ideas for the students to design various circuits to satisfy the performance parameters of the design.

.

OUTCOMES:

1. Study transfer, dynamic characteristics of various analog and digital circuits .

2. Learn the circuit design using cadence tools.

3. Draw layouts using Cadence for various circuits and doing simulations.

4. Generates interest for the students to do work on core.

EXPERIMENT NO: 1

1.1 AIM: Plotting the (i) output characteristics (ii) Transfer characteristics of an n-channel and p-channel MOSFET with Cadence.

1.2 LEARNING OBJECTIVE: To understand basic characteristic and operation of MOSFET .

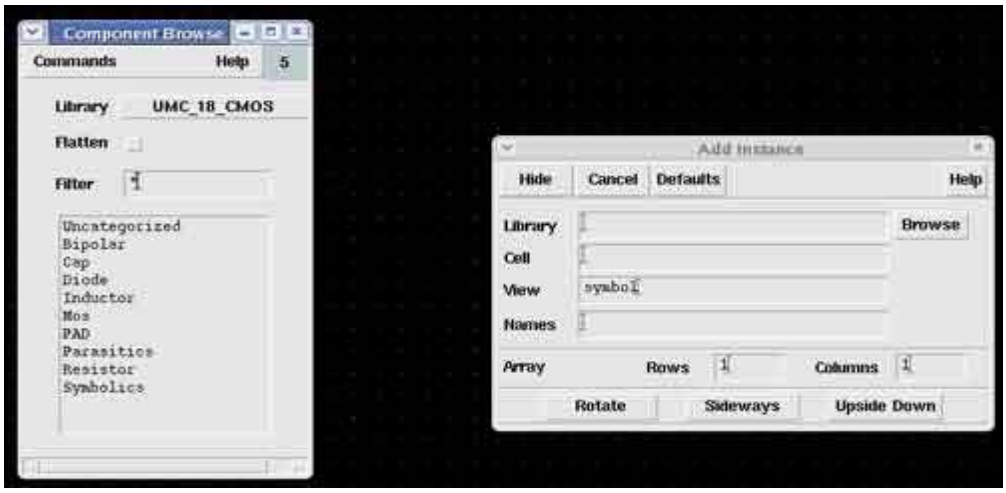
1.3 TOOLS REQUIRED: PC,CADENCE TOOLS

1.4 PROCEDURE:

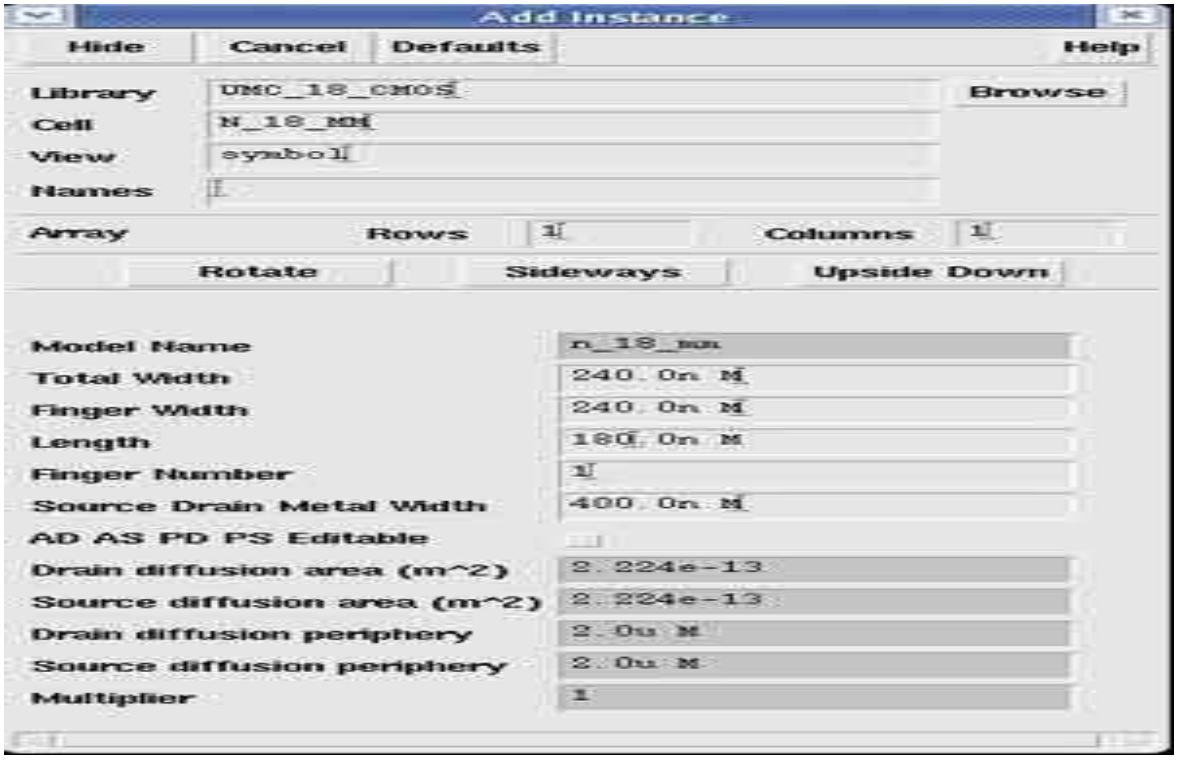
Start by creating a new schematic cell view in you existing or newly created library. Creation of new library and cell view is already covered in **"First Look at Cadence"** page.

Schematic Creation:

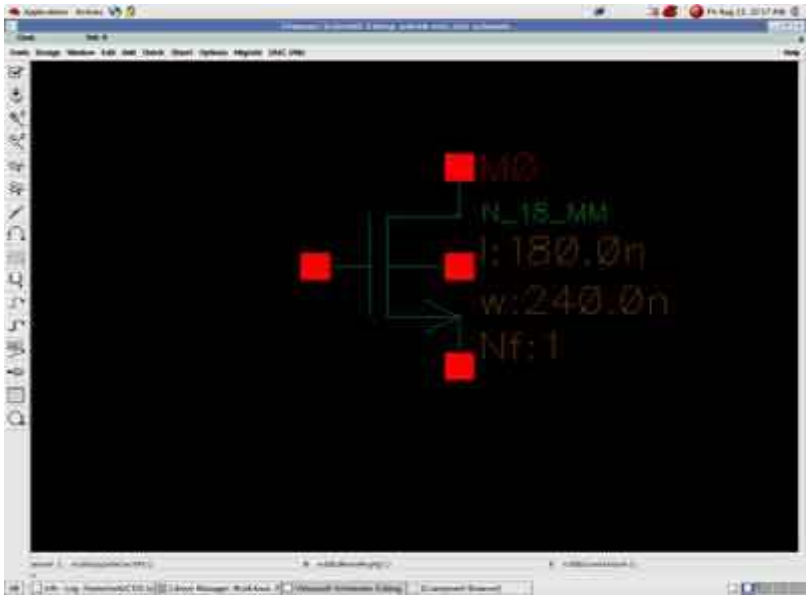
- Create a new schematic cell view where we shall instantiate a NMOS and apply some Vgs and Vds and plot the drain currents at different operating points.
- In a new schematic editor window, **press "i"**. This will invoke a new subwindow called Add an instance window.
- Here we can select what we wish to add to the schematic.

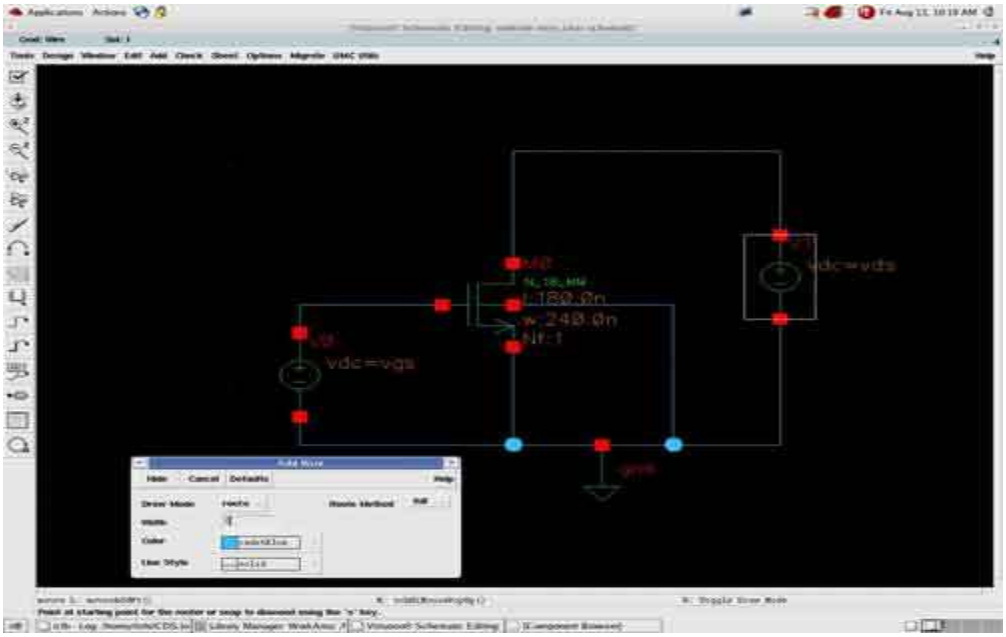


- We can browse for an instance called **N_18_MM** inside the **UMC_18_CMOS** library, and select the **symbol view** from the browser window.
- Now the NMOS is attached to our mouse cursor and we can place the NMOS by just clicking on an empty space on the schematic editor window.
- A window shown below will appear where can can change the W and L of the transistor and even rotate the transistor in all ways and direction by the Rotate, Sideways and Upside Down keys.

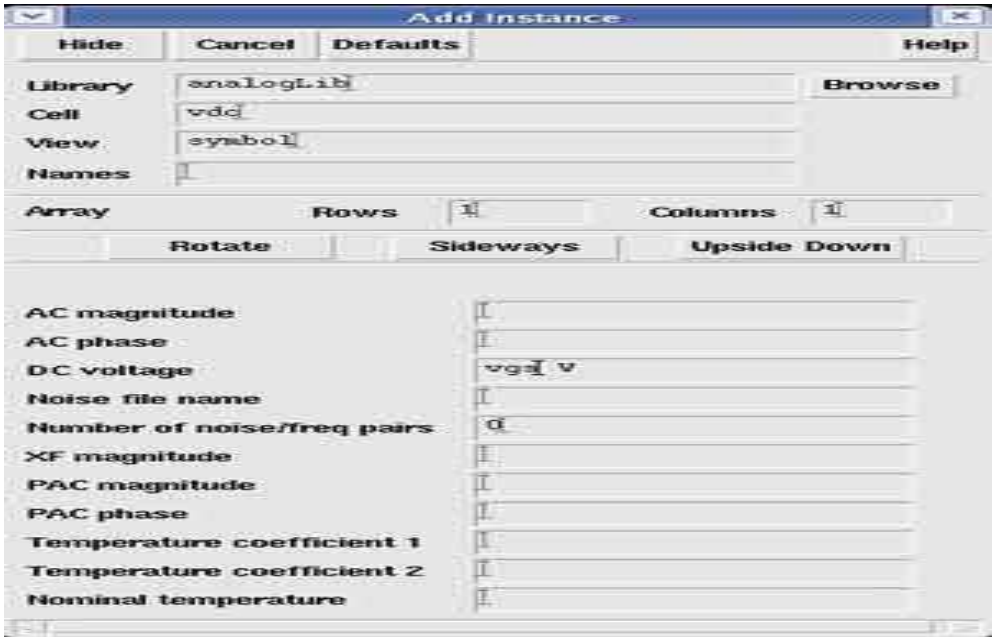


- The device is still seen as attached to the mouse, which can be removed by pressing **"ESC"** key. After placing the transistor, the schematic would look something like this.
- The top terminal of the NMOS is the drain, bottom one is the source (clear from the arrow), the terminal on the left is gate and on centre right is body.
- Now we have to add dc supply sources. One Vdc source for gate to source voltage and one for drain to source voltage.
- Again invoke add an instance menu by pressing i and browse for an instance called **"vdc" inside analogLib.**
- Note that analogLib can be sorted by categories by ticking the show category option at the top of the browser window.
- Vdc can be found under **analogLib > Sources > Independent > Vdc**. Draw the schematic as shown below.
- The wires can be drawn by **pressing "w"** then click on starting point, then click on ending point. NOTE that a **gnd! instance has to be added** to the schematic.
- Else the simulator will not be able to resolve the voltages as no reference would be specified then.





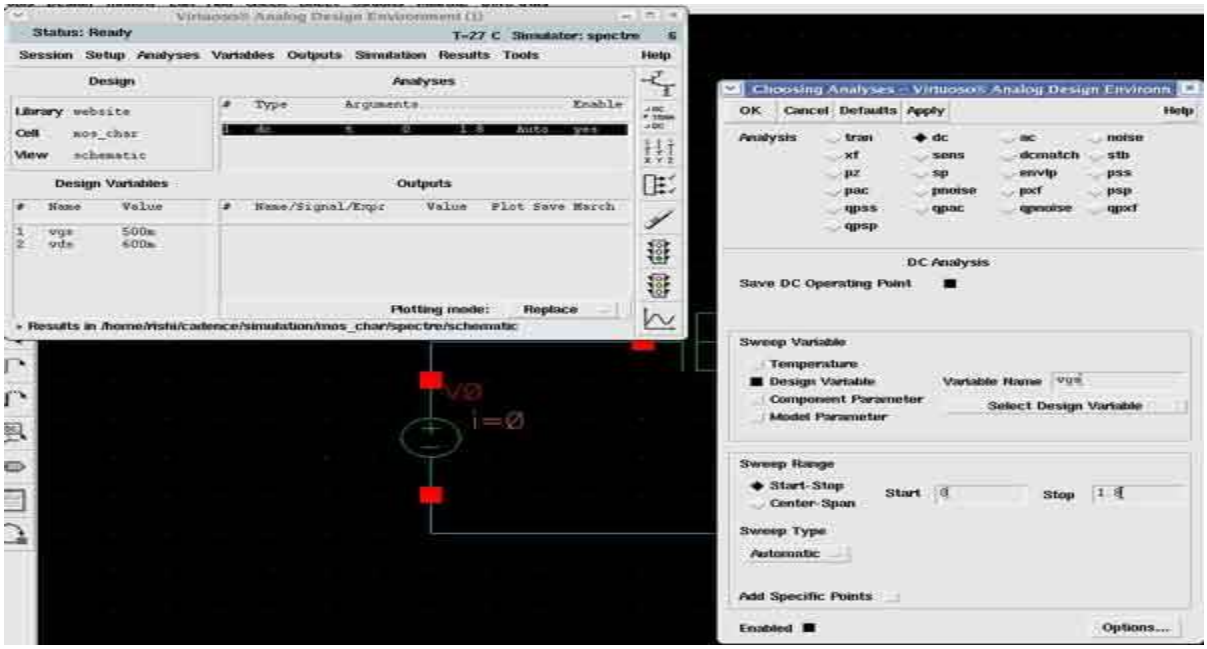
- Now the value of the dc sources as to be set.
- Choose a dc source, and press **"q"**. This is open the query page.
- In the row DC Voltage, fill the values **"vgs"** and **"vds"** for the two voltage sources correctly.
- Note that no units are to be added. Cadence will automatically take it in voltage.



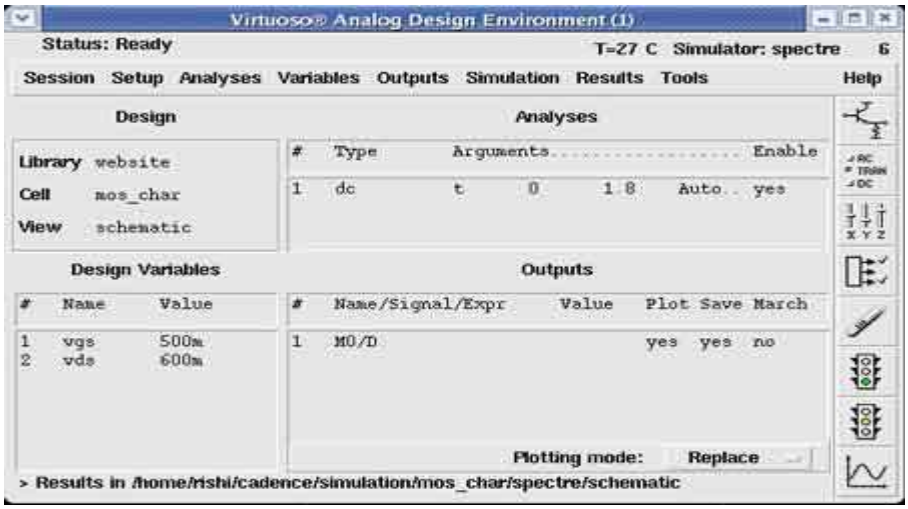
- Also the W and L of the transistor can be changed at any time by selecting the transistor and pressing q. the query page "q" is generally used to set properties of all the components and devices invoked from the library manager.
- Once the schematic is ready, press the **"Check and Save"** button on top left in the schematic editor window (tick symbol button).
- This will check for errors and save and will report if there are any errors or warnings.
- Errors cannot be ignored but warnings may be ignored if you are aware and sure that the warning is harmless.
- Now its time to simulate.

DC Analysis

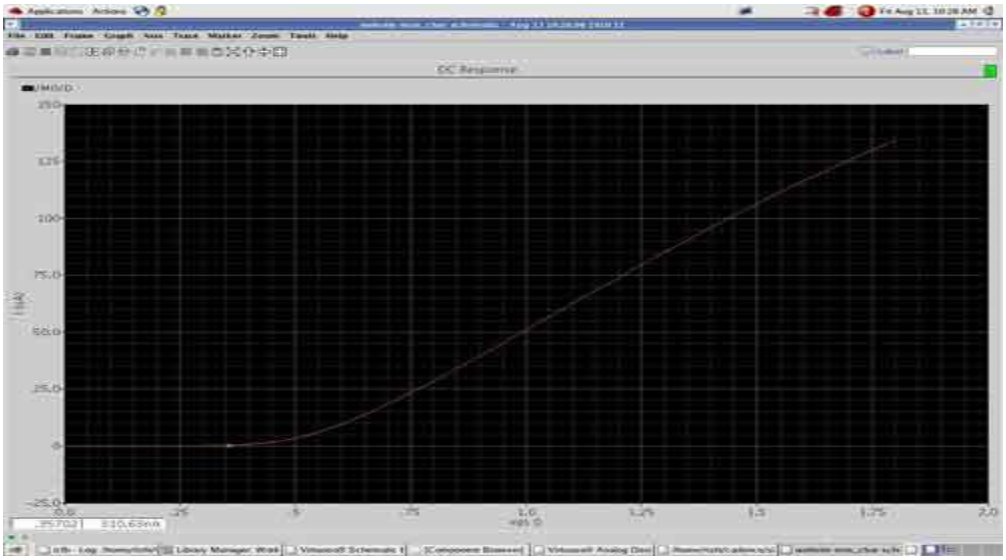
- Select **Tools > Analog Environments**.
- A new window opens up. On the menu on top, select **variable > copy from cellview**.
- Immediately, vgs and vds would appear on the low left side of this window.
- Double click them and assign some initial value, like vgs=0.5 V and vds=0.6 V.
- from Menu, click **Analysis > Choose**. click on **dc** and click on **save operating points**.
- Also select component parameter below. this will make some more options appear.
- Click on **select component** twice.
- This will take you to schematic, click on a voltage source for vgs, then in new popup window select **dc voltage** and then **OK**.
- Come back to the analysis window by using ALT-TAB and give the start = 0 and stop 1.8V (Since maximum supply is 1.8V for our process).
- Then press **OK** on analysis window.



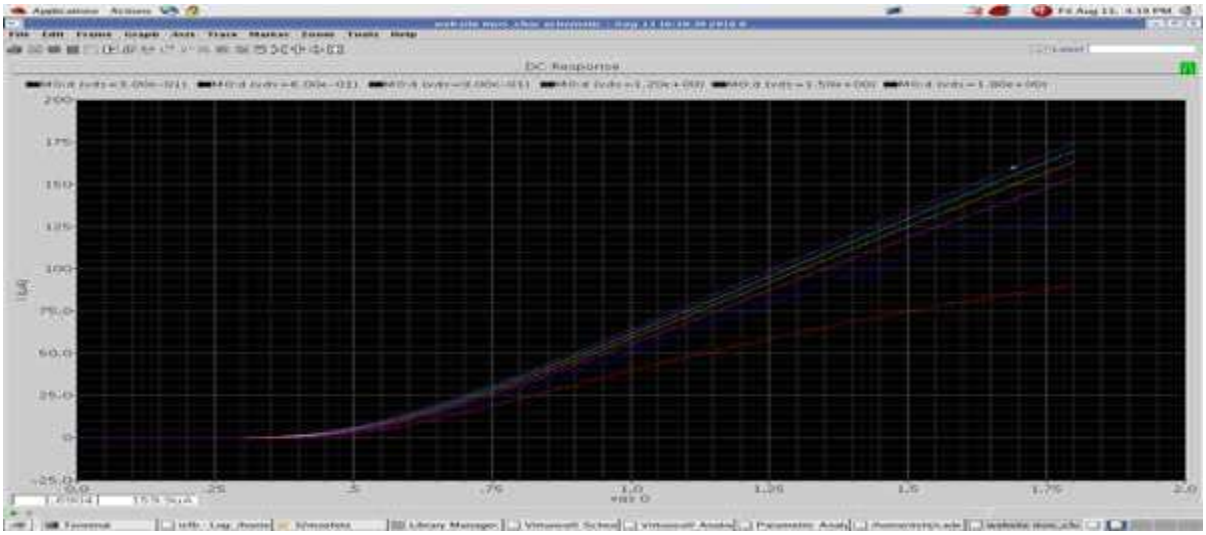
- Coming back to the Analog Environments, select **output > to be plotted > select on schematic**.
- Now select the drain terminal of the NMOS transistor by clicking on it. then press Esc.
- The Final analysis window will look like shown below.



- Come back to Analog Environment and notice that the output is added and
- select **Simulation > Netlist and Run** or just press the Netlist and Run button on the right (third button from down).
- Now simulation will start and a plot window will appear as shown below.



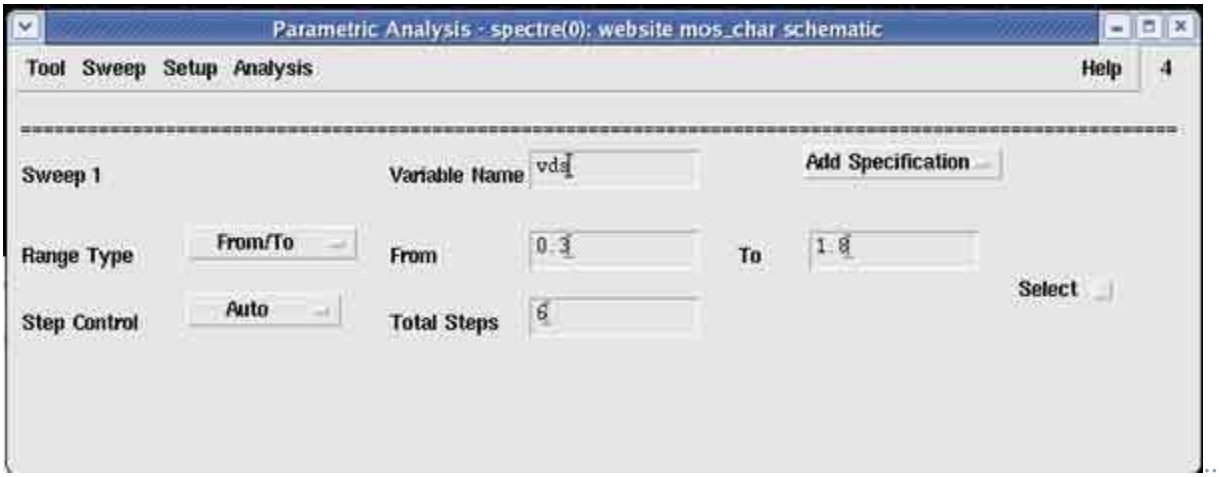
- The X-axis is Vgs and the Y-Axis is Id.
- The Id Vgs curve shown above is for the specified value of vds (specified to variable vds in analog environ ment window).



Parametric Analysis

- We can also plot Id Vgs characteristics for more than one value of Vds on the same graph at the same time. Such plots can be achieved by parametric analysis.
- Let us consider that we wish to plot the below given graph.
- We have Vgs on the X Axis and Id on the Y Axis. Each curve on the plot is for different values of Vds. Therefore we select vgs as the sweep variable in dc analysis and vds as the variable of parametric analysis.
- Just like earlier, from analog environment, we select vgs voltage source in component parameter sweep in DC Analysis. Sweep it from 0 to 1.8V.
- Select the drain terminal of the transistor as the current plot.

- Then from Analog Environment window, we select Tools > Parametric. This will open up a new window as shown below.



- we fill up the above window as shown.
- Note that the variable name "vds" is same as the variable name given to the dc voltage value of the voltage source which applies the vds of the transistor.
- To eliminate variable name errors, in this window, choose **Setup > Variable name > sweep 1**.
- Then select vds as the parametric sweep variable. Give in the range and the number of steps as shown above.
- Then click **Analysis > Start.** Simulation will run, and the above shown graph for Id Vs. Vgs for various vds will be plotted.

1.5 RESULT : Understand the basic operation and characteristics of MOS transistors.

1.6 PRE LAB VIVA QUESTIONS:

1. What is the difference between MOSFET and BJT
2. What are the advantages of MOSFET?
3. What are different mode of operations of MOSFET?

1.7 POST LAB VIVA QUESTIONS:

1. What are the characteristics of Enhancement mode?
2. Define cutoff region?
3. Define pinch off region?
4. Define linear region?

EXPERIMENT NO: 2

2.1 AIM: To design and plot the static (VTC) and dynamic characteristics of a digital CMOS inverter.

2.2 LEARNING OBJECTIVE: To understand the characteristics of CMOS inverter.

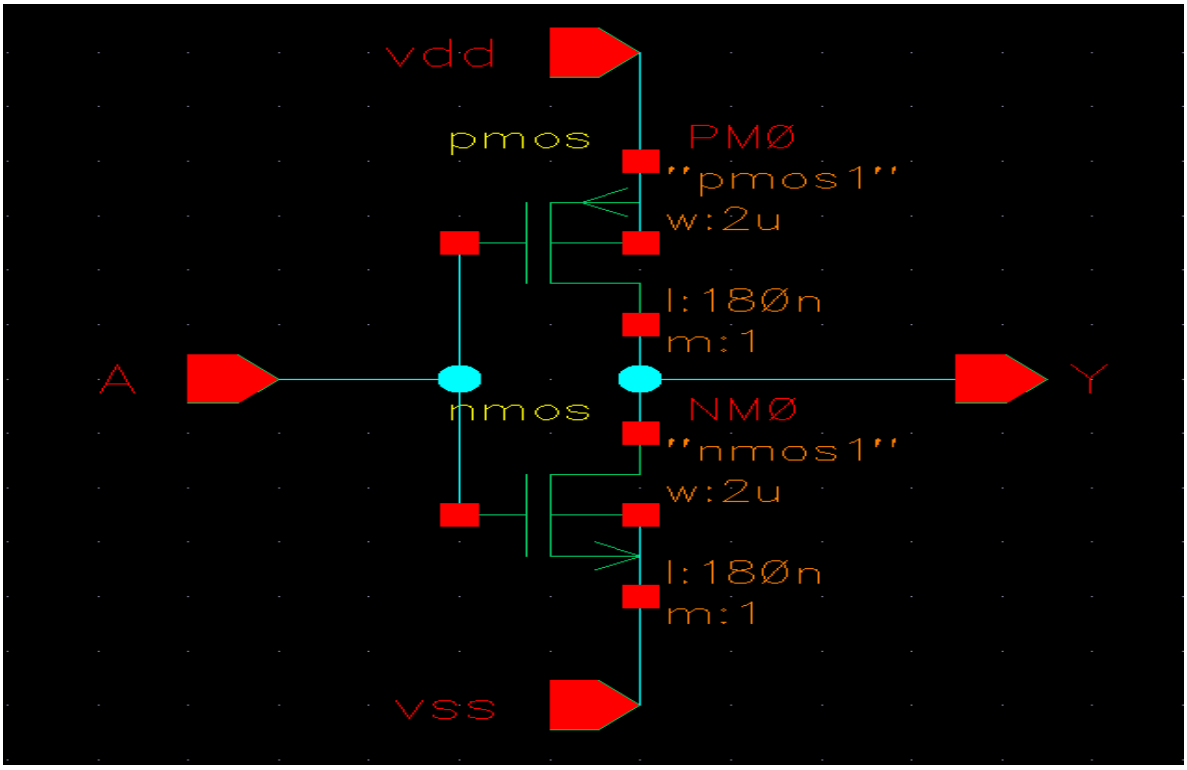
2.3 TOOLS REQUIRED: PC,CADENCE TOOLS

2.4 PROCEDURE:

Schematic Entry:

Objective: To create a library and build a schematic of an Inverter

Below steps explain the creation of new library "**myDesignLib**" and we will use the same throughout this course for building various cells that we going to create in the next labs. Execute **Tools – Library Manager** in the CIW or Virtuoso window to open Library Manager.



Creating a New library:

- In the Library Manager, execute **File - New – Library**. The new library form appears.

- In the "New Library" form, type "**myDesignLib**" in the Name section.In the field of Directory section, verify that the path to the library is set to **~/Database/cadence_analog_labs_613** and click **OK**.

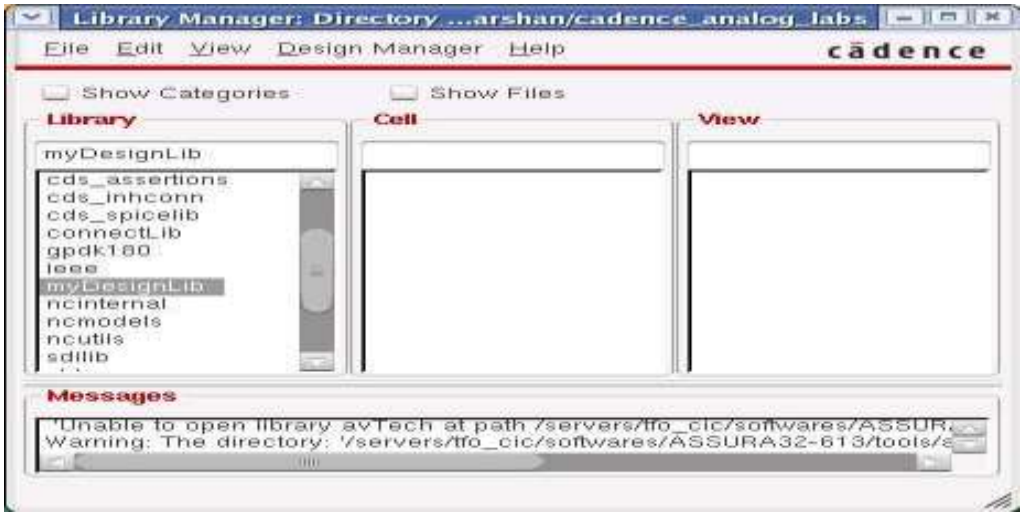
  Note: A technology file is not required if you are not interested to do the layouts for the design.

- In the next "**Technology File for New library**" form, select option **Attach to an existing techfile** and click **OK**.

- In the "**Attach Design Library to Technology File**" form, select **gpdk180** from the cyclic field and click **OK**.

- After creating a new library you can verify it from the library manager.
- If you right click on the "**myDesignLib**" and select properties, you will find that **gpdk180** library is

attached as techlib to "**myDesignLib**".



Creating a Schematic Cellview

In this section we will learn how to open new schematic window in the new **myDesignLib**" library and build the inverter schematic as shown in the figure at the start of this lab.

- In the CIW or Library manager, execute **File – New – Cellview**.
- Set up the New file form as follows:Do not edit the **Library path file** and the one above might be different from the path shown in your form.

- Click **OK** when done the above settings. A blank schematic window for the **Inverter** design appears.

Adding Components to schematic

- In the Inverter schematic window, click the **Instance** fixed menu icon to display the Add Instance form. 
- **Tip:** You can also execute **Create — Instance** or press **i**.
- Click on the **Browse** button. This opens up a Library browser from which you can select components and the **symbol** view .You will update the Library Name, Cell Name, and the property values given in the table on the next page as you place each component.
- After you complete the Add Instance form, move your cursor to the schematic window and click **left** to place a component.
- This is a table of components for building the Inverter schematic.

Library name	**Cell Name**	**Properties/Comments**
gpdk180	pmos	For M0: Model name = pmos1, **W= wp**, L=180n
gpdk180	nmos	For M1: Model name = nmos1, W= 2u, L=180n

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

- You can rotate components at the time you place them, or use the **Edit— Rotate**
- command after they are placed.

Adding pins to Schematic

- Click the **Pin** fixed menu icon in the schematic window. You can also execute **create — Pin** or press ***p***.  The Add pin form appears.
- Type the following in the Add pin form in the exact order leaving space between the pin names.

Pin Names	**Direction**
vin	Input
vout	Output

Make sure that the direction field is set to **input/output/inputOutput** when placing the **input/output/inout** pins respectively and the Usage field is set to **schematic**.

- Select **Cancel** from the Add – pin form after placing the pins. In the schematic window, execute **Window— Fit** or press the **f** bindkey.

Adding Wires to a Schematic

Add wires to connect components and pins in the design.

- Click the **Wire (narrow)** icon in the schematic window.
- You can also press the ***w*** key, or execute **Create — Wire (narrow).**
- In the schematic window, click on a pin of one of your components as the first point for your wiring. A diamond shape appears over the starting point of this wire.
- Follow the prompts at the bottom of the design window and click **left** on the destination point for your wire. A wire is routed between the source and destination points.
- Complete the wiring as shown in figure and when done wiring press **ESC** key in the schematic window to cancel wiring.

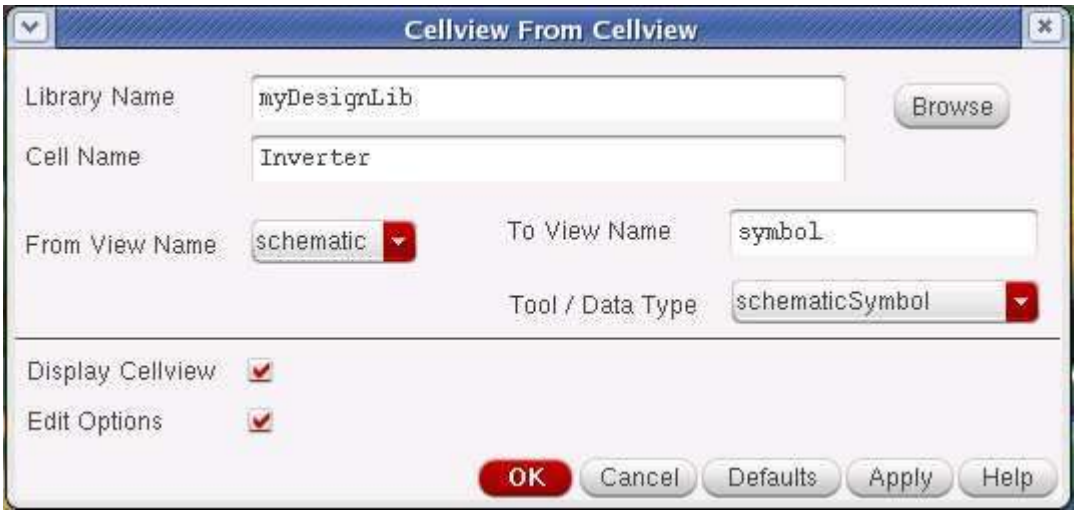
Saving the Design

- Click the **Check and Save** icon in the schematic editor window.
- Observe the CIW output area for any errors.

Symbol Creation

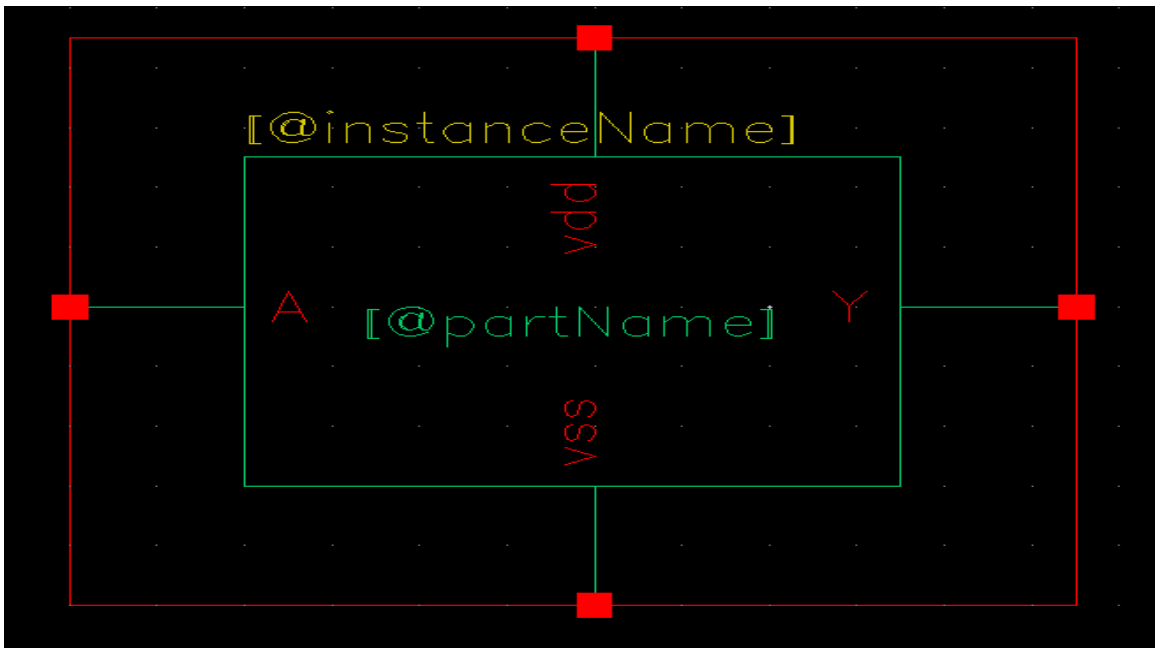
In this section, you will create a symbol for your inverter design so you can place it in a test circuit for simulation. A symbol view is extremely important step in the design process. The symbol view must exist for the schematic to be used in a hierarchy. In addition, the symbol has attached properties (cds Param) that facilitate the simulation and the design of the circuit.

- In the Inverter schematic window, execute **Create — Cellview— From Cellview**.
- The **Cellview From Cellview** form appears. With the Edit Options function active, you can control the appearance of the symbol to generate.
- 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 **Schematic Symbol**.



- Click **OK** in the **Cellview From Cellview** form.
- The Symbol Generation Form appears.
- Modify the **Pin Specifications** as follows:

- Click **OK** in the Symbol Generation Options form.
- A new window displays an automatically created Inverter symbol as shown here.



Editing a Symbol

In this section we will modify the inverter symbol to look like a Inverter gate symbol.

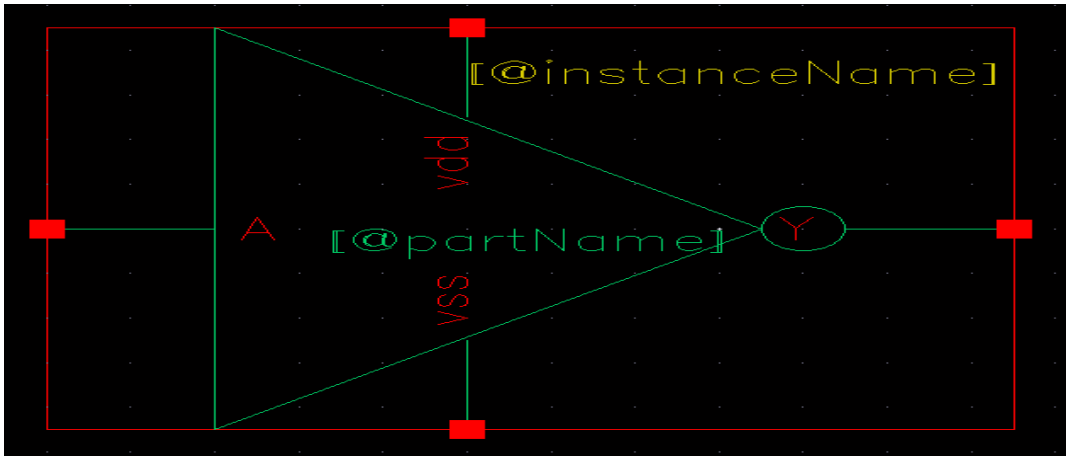
Move the cursor over the automatically generated symbol, until the green rectangle is highlighted, click **left** to select it.

- Click **Delete** icon in the symbol window, similarly select the red rectangle and delete that.
- Execute **Create – Shape – polygon**, and draw a shape similar to triangle.
- After creating the triangle press ***ESC*** key.
- Execute **Create – Shape – Circle** to make a circle at the end of triangle.
- You can move the pin names according to the location.
- Execute **Create — Selection Box**. In the Add Selection Box form, click
- **Automatic**.A new red selection box is automatically added.
- After creating symbol, click on the ***save*** icon in the symbol editor window to save the symbol. In the symbol editor, execute **File — Close** to close the symbol view window.

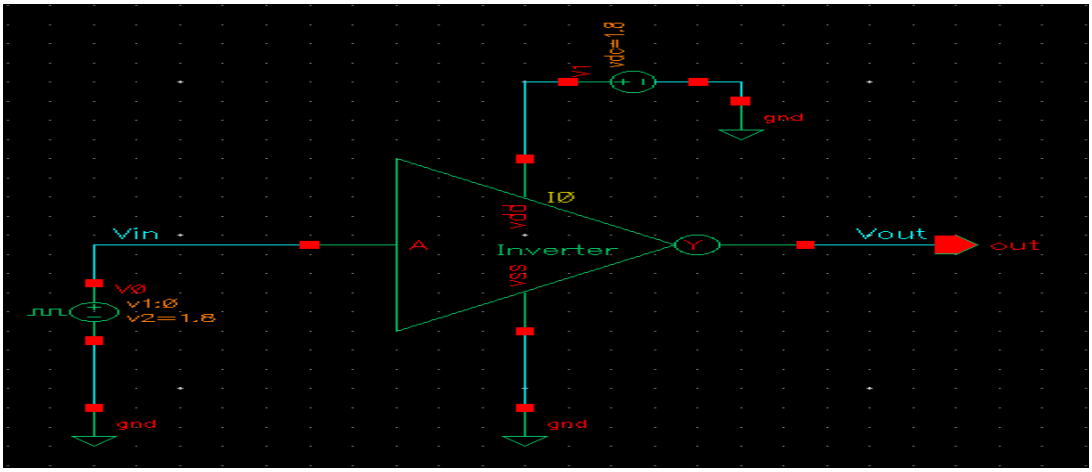
Building the Inverter_Test Design:Creating the Inverter_Test Cellview

You will create the Inverter_Test cellview that will contain an instance of the Inverter cellview. In the next section, you will run simulation on this design

- In the CIW or Library Manager, execute **File— New— Cellview**.
- Set up the New File form as follows:



- Click **OK** when done. A blank schematic window for the **Inverter_Test** design appears.



Building the Inverter_Test Circuit

- Using the component list and Properties/Comments in this table, build the Inverter_Test schematic.

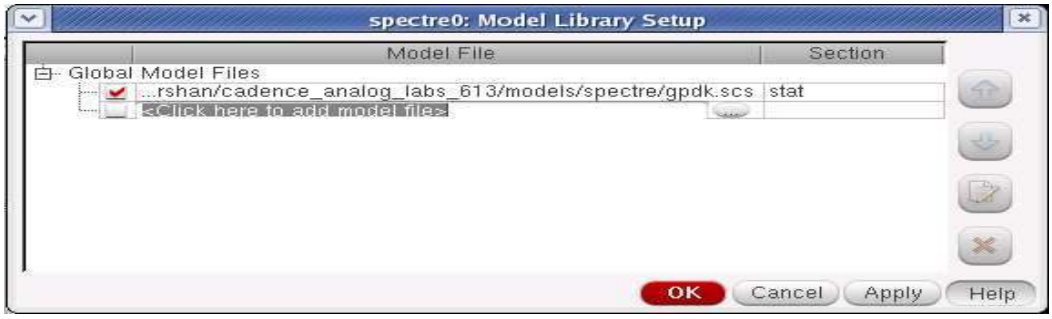
Library name	Cellview name	Properties/Comments
myDesignLib	Inverter	Symbol
analogLib	vpulse	v1=0, v2=1.8,td=0 tr=tf=1ns,ton=10n, T=20n
analogLib	vdc, gnd	vdc=1.8

Note: Remember to set the values for **VDD** and **VSS**. Otherwise, your circuit will have no power.

- Add the above components using **Create — Instance** or by pressing **I**. 
- Click the **Wire (narrow)** icon and wire your schematic. 
- **Tip:** You can also press the **w** key, or execute **Create— Wire (narrow)**.
- Click **Create — Wire Name** or press **L** to name the input **(Vin)** and output **(Vout)** wires as in the below schematic. 
- Click on the **Check and Save** icon to save the design. 
- The schematic should look like this.
- Leave your **Inverter_Test** schematic window open for the next section.
- **Analog Simulation with Spectre:** To set up and run simulations on the Inverter_Test design
- In this section, we will run the simulation for Inverter and plot the transient, DC characteristics and we will do Parametric Analysis after the initial simulation.
- **Starting the Simulation Environment:**Start the Simulation Environment to run a simulation.
- In the **Inverter_Test** schematic window, execute
- **Launch – ADE L:**The Virtuoso Analog Design Environment (ADE) simulation window appears.
- **Choosing a Simulator**
- Set the environment to use the **Spectre® tool**, a high speed, highly accurate analog simulator. Use this simulator with the **Inverter*_Test*** design, which is made-up of analog components.
- In the simulation window (ADE), execute Setup— Simulator/Directory/Host.
- In the Choosing Simulator form, set the Simulator field to **spectre** (Not spectreS) and click **OK**.

Setting the Model Libraries:

- The Model Library file contains the model files that describe the nmos and pmos devices during simulation.
- In the simulation window (ADE), Execute **Setup - Model Libraries.** The Model Library Setup form appears. Click the **browse** button  to add **gpdk.scs** if not added by default as shown in the **Model Library Setup** form.
- **Remember** to select the section type as **stat** in front of the gpdk.scs file. Your Model Library Setup window should now looks like the below figure.



To view the model file, highlight the expression in the Model Library File field and Click **Edit File**. 

- To complete the Model Library Setup, move the cursor and click **OK**. The Model Library Setup allows you to include multiple model files. It also allows you to use the Edit button to view the model file.

Choosing Analyses

This section demonstrates how to view and select the different types of analyses to complete the circuit when running the simulation.

- In the Simulation window (ADE), click the **Choose - Analyses** icon.

  You can also execute **Analyses - Choose.**

  The Choosing Analysis form appears. This is a dynamic form, the bottom of the form changes based on the selection above.

- To setup for transient analysis
  - **a.** In the Analysis section select **tran**
  - **b.** Set the stop time as **200n**
  - c. Click at the **moderate** or **Enabled** button at the bottom, and then click Apply.

- To set up for DC Analyses:
  - a. In the Analyses section, select **dc**.
  - b. In the DC Analyses section, turn on **Save DC Operating Point.**

c. Turn on the **Component Parameter**.

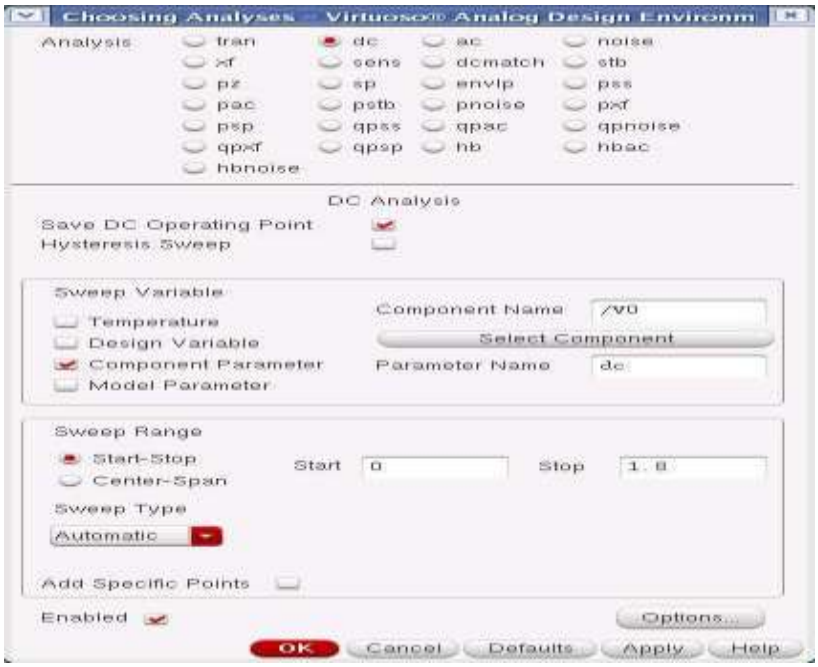
d. Double click the **Select Component**, Which takes you to the schematic window.

e. Select input signal **vpulse source** in the test schematic window.

f. Select **"DC Voltage"** in the **Select Component Parameter** form and click OK.

g. In the analysis form type **start** and **stop** voltages as **0** to **1.8** respectively.

h. Check the enable button and then click **Apply.**



- Click ***OK*** in the Choosing Analyses Form.

Setting Design Variables

Set the values of any design variables in the circuit before simulating. Otherwise, the simulation will not run.

- In the Simulation window, click the **Edit Variables** icon.
- The Editing Design Variables form appears.
- Click **Copy From** at the bottom of the form. The design is scanned and all variables found in the design are listed. In a few moments, the **wp** variable appears in the Table of Design variables section.
- Set the value of the **wp** variable: With the **wp** variable highlighted in the Table of Design Variables, click on the variable name **wp** and enter the following:

Value(Expr)	2u

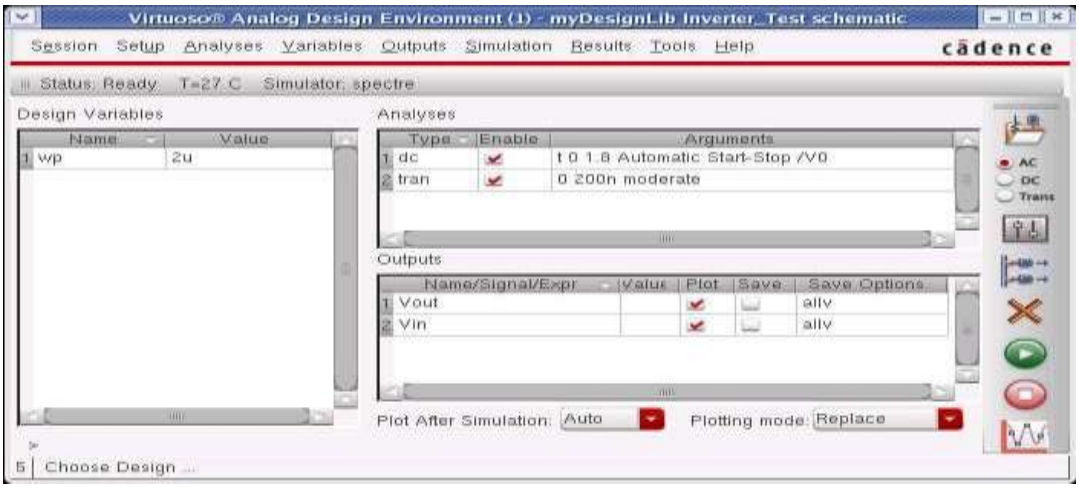
Click **Change** and notice the update in the Table of Design Variables.

- Click **OK** or **Cancel** in the Editing Design Variables window.

Selecting Outputs for Plotting

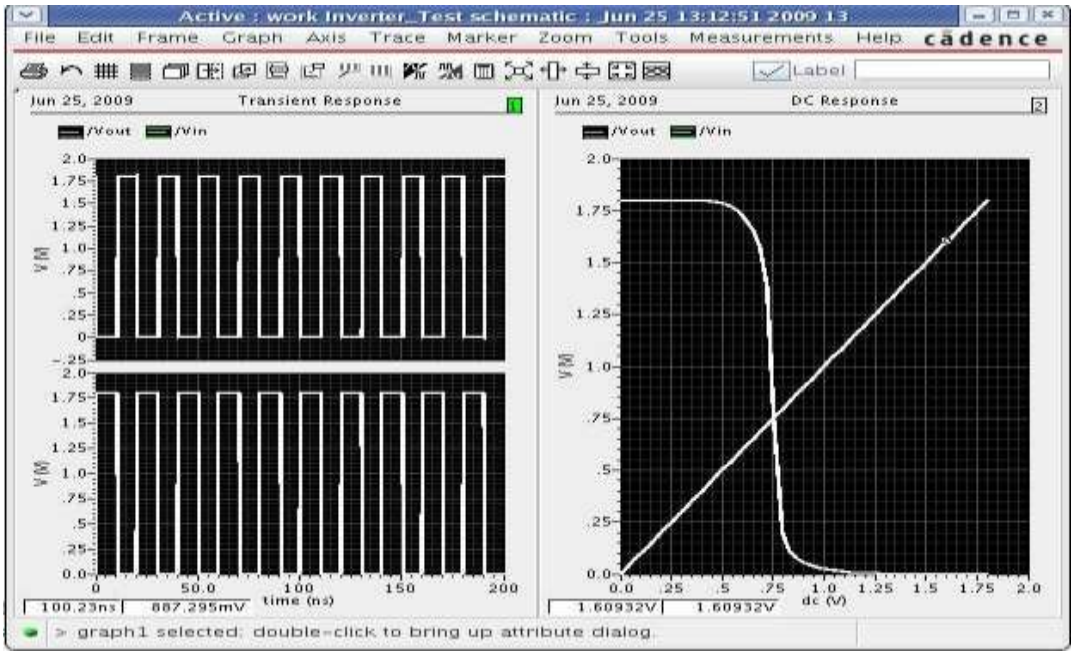
- Execute **Outputs – To be plotted – Select on Schematic** in the simulation window.
- Follow the prompt at the bottom of the schematic window, Click on output net **Vout**, input net **Vin** of the Inverter. Press **ESC** with the cursor in the schematic after selecting it.

Does the simulation window look like this?



Running the Simulation:

- Execute **Simulation – Netlist and Run** in the simulation window to start the
- Simulation or the icon, this will create the netlist as well as run the simulation.
- When simulation finishes, the Transient, DC plots automatically will be popped up along with log file.



Saving the Simulator State

We can save the simulator state, which stores information such as model library file, outputs, analysis, variable etc. This information restores the simulation environment without having to type in all of setting again.

- In the Simulation window, execute **Session – Save State**. The Saving State form appears.
- Set the **Save as** field to **state1_inv** and make sure all options are selected under what to save field.
- Click **OK** in the saving state form. The Simulator state is saved.

Loading the Simulator State

- From the ADE window execute **Session – Load State.**
- In the Loading State window, set the State name to **state1_inv** as shown

- Click **OK** in the Loading State window.

Parametric Analysis

- Parametric Analysis yields information similar to that provided by the Spectre® sweep feature, except the data is for a full range of sweeps for each parametric step. The Spectre sweep feature provides sweep data at only one specified condition.

- You will run a parametric DC analysis on the **wp** variable, of the PMOS device of the Inverter design by sweeping the value of **wp**.

- Run a simulation before starting the parametric tool. You will start by loading the state from the previous simulation run.

- Run the simulation and check for errors. When the simulation ends, a single waveform in the waveform window displays the DC Response at the **Vout** node.

Starting the Parametric Analysis Tool

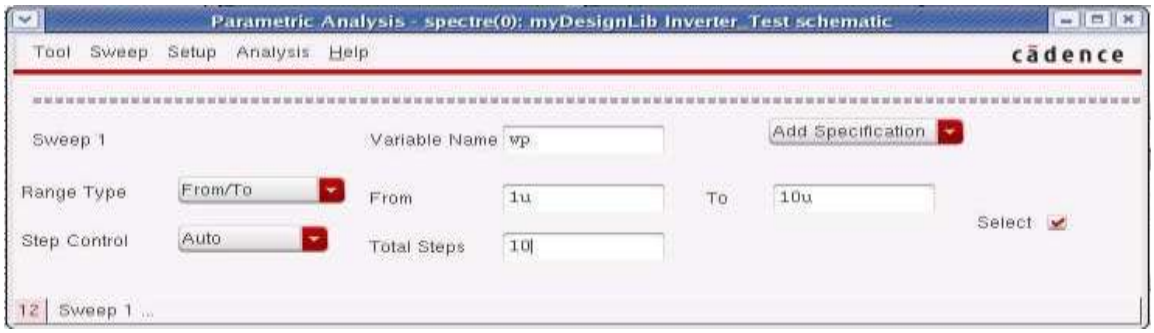
- In the Simulation window, execute **Tools—Parametric Analysis**. The Parametric Analysis form appears.
- In the Parametric Analysis form, execute **Setup—Pick Name For Variable—Sweep 1**.

A selection window appears with a list of all variables in the design that you can sweep. This list includes the variables that appear in the Design Variables section of the Simulation window.

- In the selection window, double click left on **wp**.The Variable Name field for Sweep 1 in the Parametric Analysis form is set to **wp**.
- Change the Range Type and Step Control fields in the Parametric Analysis form as shown below:

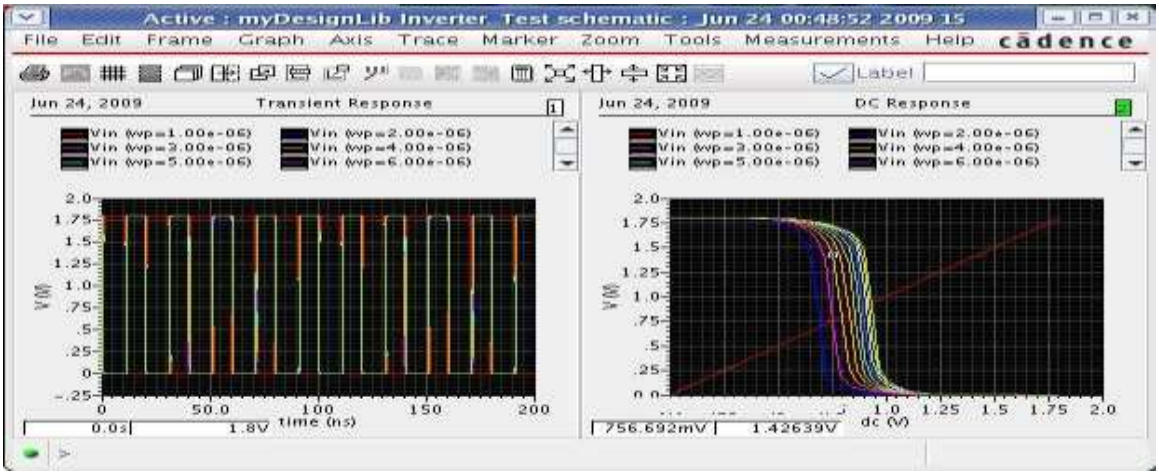
Range Type	**From/To**	From	**1u**	To	**10u**
Step Control	**Auto**	Total Steps	**10**		

These numbers vary the value of the **wp** of the pmos between 1um and 10um at ten evenly spaced intervals.



- Execute **Analysis—Start**.

The Parametric Analysis window displays the number of runs remaining in the analysis and the current value of the swept variable(s). Look in the upper right corner of the window. Once the runs are completed the wavescan window comes up with the plots for different runs.



Note: Change the wp value of pmos device back to 2u and save the schematic before proceeding to the next section of the lab. To do this use edit property option.

2.5 RESULT : Designed and verified the static (VTC) and dynamic characteristics of a digital CMOS inverter.

2.6 PRE LAB VIVA QUESTIONS:

1. What is the function of inverter?
2. Define CMOS Inverter?
3. Define nMOS inverter?

2.7 POST LAB VIVA QUESTIONS:

1. What is the advantage of this tool?
2. What do you observe from characteristics of Inverter?

EXPERIMENT NO: 3

3.1 AIM: To design and plot the output characteristics of a 3-inverter ring oscillator.

3.2 LEARNING OBJECTIVE: To understand the characteristics of ring oscillator.

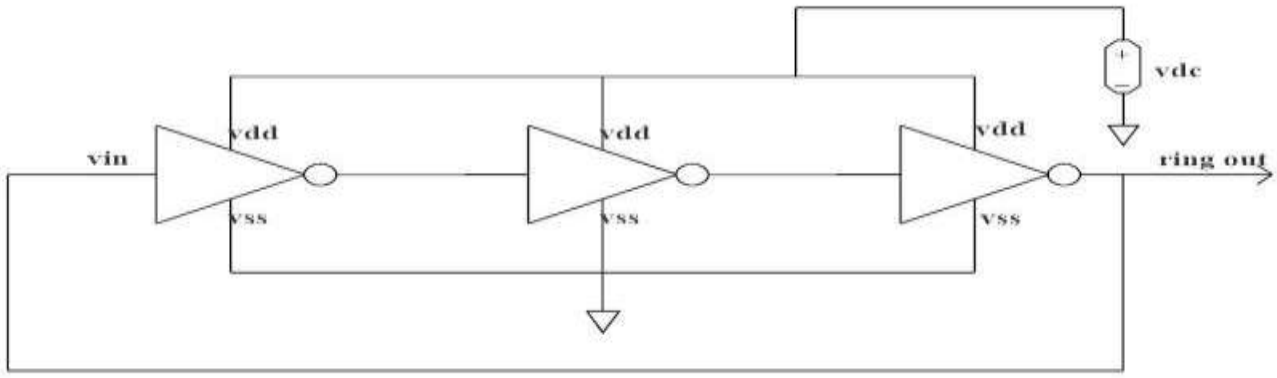
3.3 TOOLS REQUIRED: PC

CADENCE TOOLS

3.4 INTRODUCTION: A device that consists of odd number of NOT gates is referred to as Ring Oscillator. The output of these gates oscillates between two voltage levels (between 0 and 1). The immunity to external disturbances is provided by means of the Ring Oscillator. The output of the last Inverter is fed back to the Input. The input is same as the last output. A Ring Oscillator requires power above threshold Voltage to operate. At this voltage, oscillation starts spontaneously. The frequency of oscillation and the current usage can be decreased by decreasing the applied voltage.

Ring Oscillator is one of the members of class time delay oscillators. The Ring oscillator uses odd number of Inverters so that gain can be increased greater than 1. Instead of having one delay element, each inverter contributes delay around the ring of Inverters. Hence, the name Ring Oscillator is given.

PROCEDURE: Design the Ring Oscillator schematic as shown in Figure with following parameters. Design procedure is similar to CMOS inverter (Exp 2).

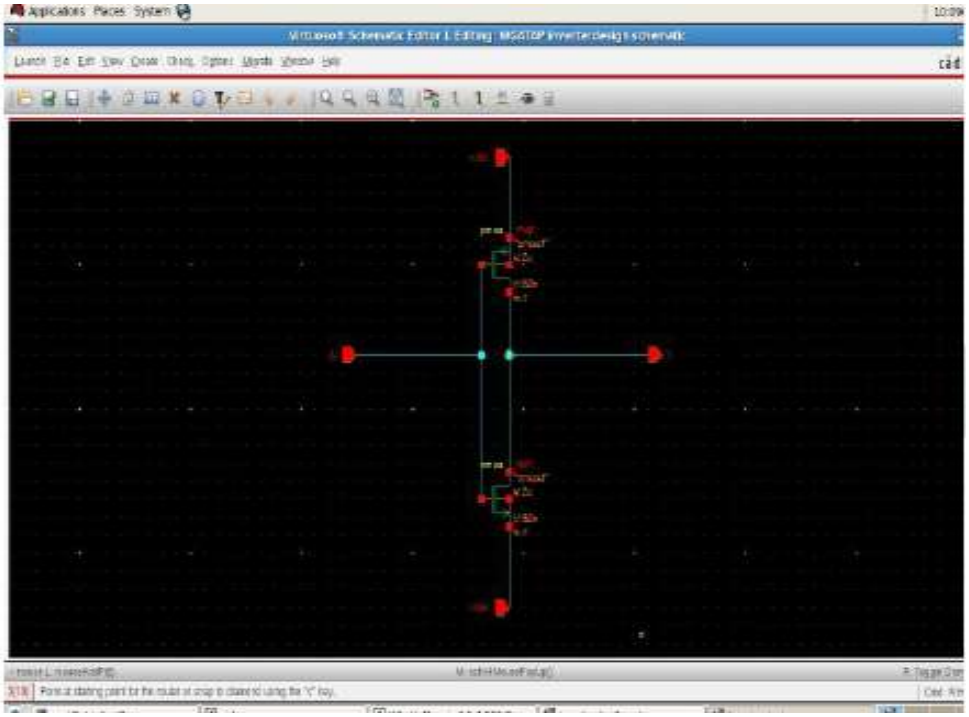


Sl. No	Parameters	Values
1	Supply Voltage	1.2 v
2	Technology	Cadence gpdk180 nm
3	Total width	2 um
4	Threshold Value	800 nm
5	Transient time	0 to 200 n
6	Clock Rise Time	1.8 ns
7	Clock Fall Time	1.8 ns
8	Clock Pulse Width	50 ns

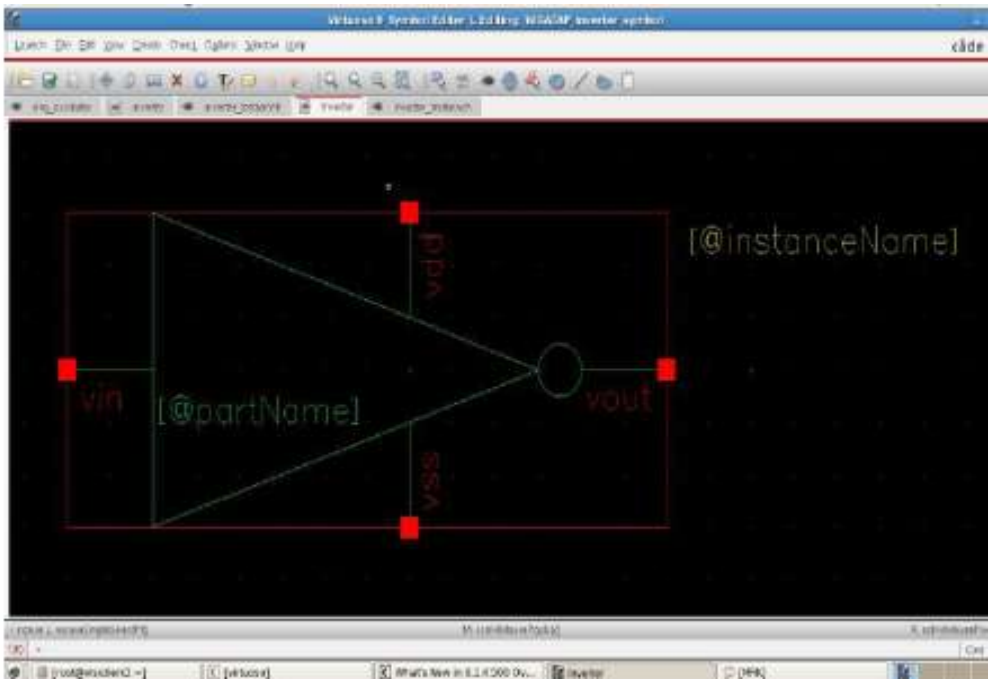
- For this first design basic CMOS Inverter as shown below.

The **schematic** of an CMOS Inverter in which the PMOS transistor and NMOS transistor connected together to form

CMOS Inverter. When low input is given, for example (0), PMOS gets ON and high output (1) is obtained . Similarly, when high input (1) is given, NMOS gets ON and low output (0) is obtained. Thus this device

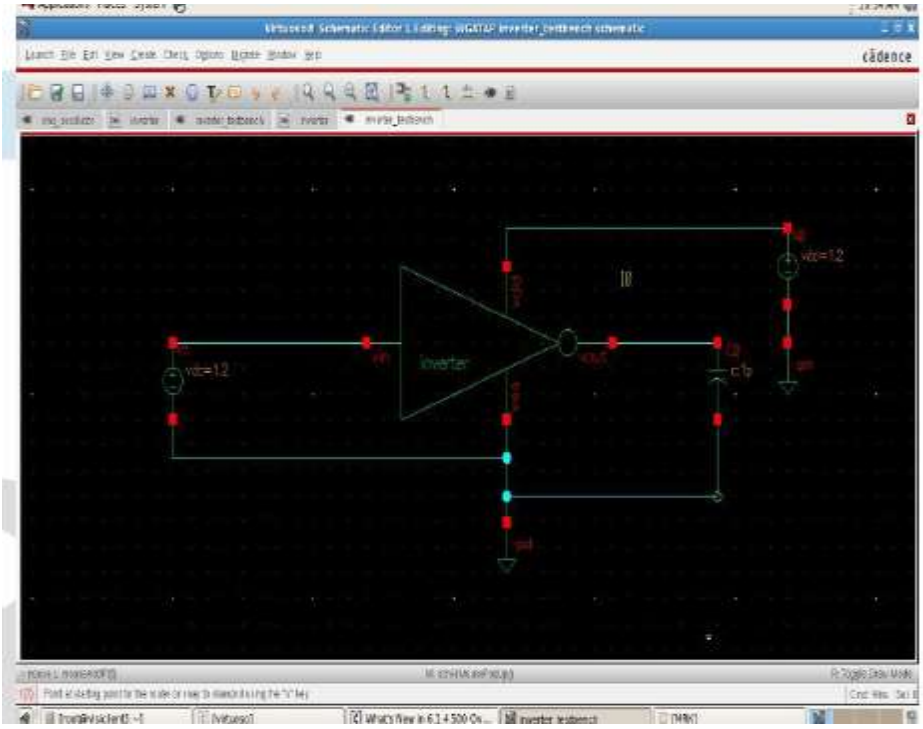


- Create the **symbol** for the inverter.

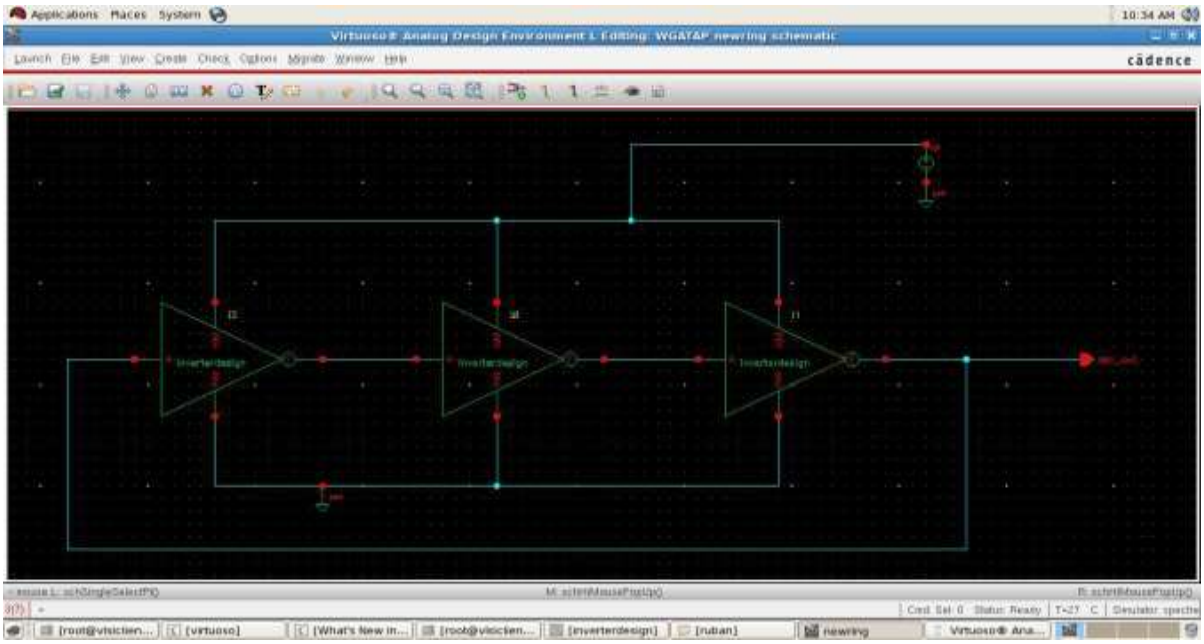


- Input pin (Vin) is formed at the left side of the Inverter. Supply Voltage pin (Vdd) is given at the top, Ground pin is provided at the bottom. Output pin (Vout) is at the right side of the Inverter.

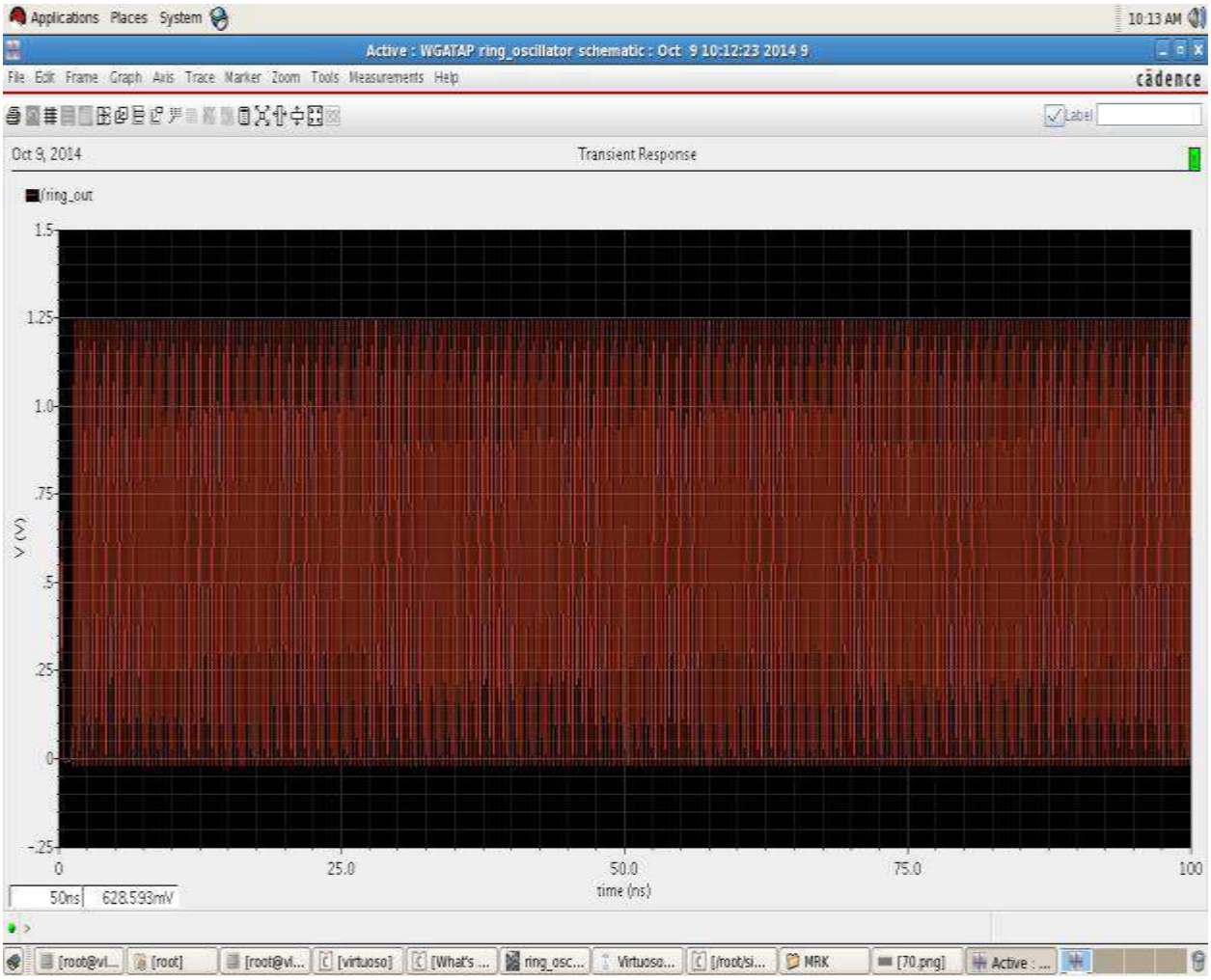
Test setup of an Inverter:



Test setup of an Inverter is shown in above Figure .The supply voltage and the input voltage is given as 1.2 volt. The Capacitor C is held at the output for the purpose of storing the charges. Now form the ring oscillator using INVERTER as shown below.



The Ring Oscillator shown in above Figure has three stages Inverter. In this ring oscillator, the output of the first inverter is given to the input of the second inverter and the second inverter output is given as the input of the third inverter. The output of the third Inverter is fed back to the input of the first Inverter, since this is an oscillator. In below Figure, The Transient response of the Ring Oscillator is shown, in which the oscillations are present due to noise in the form of non uniform waveform. The waveform formed has the maximum peak voltage of 1.2 V.



3.5 RESULT : Understood the transfer characteristic of ring oscillator.

3.6 PRE LAB VIVA QUESTIONS:

1. What is the function Ring oscillator?
2. Define Oscillator?
3. Define RC Phase shift oscillator?

3.7 POST LAB VIVA QUESTIONS:

1. How this inverter design working as ring oscillator?
2. What do you observe from characteristics of ring oscillator?

EXPERIMENT NO: 4

4.1 AIM: To design and plot the dynamic characteristics of 2-input NAND and XOR logic gates using CMOS technology.

4.2 LEARNING OBJECTIVE: To understand how to design basic logic gates like NAND and XOR, understanding its characteristics.

4.3 TOOLS REQUIRED: PC

CADENCE TOOLS

4.4 PROCEDURE:

- **Schematic Entry Objective: To create a new cell view and build A NAND gate**

Use the techniques learned in the Lab2.1 to complete the schematic of NAND gate. This is a

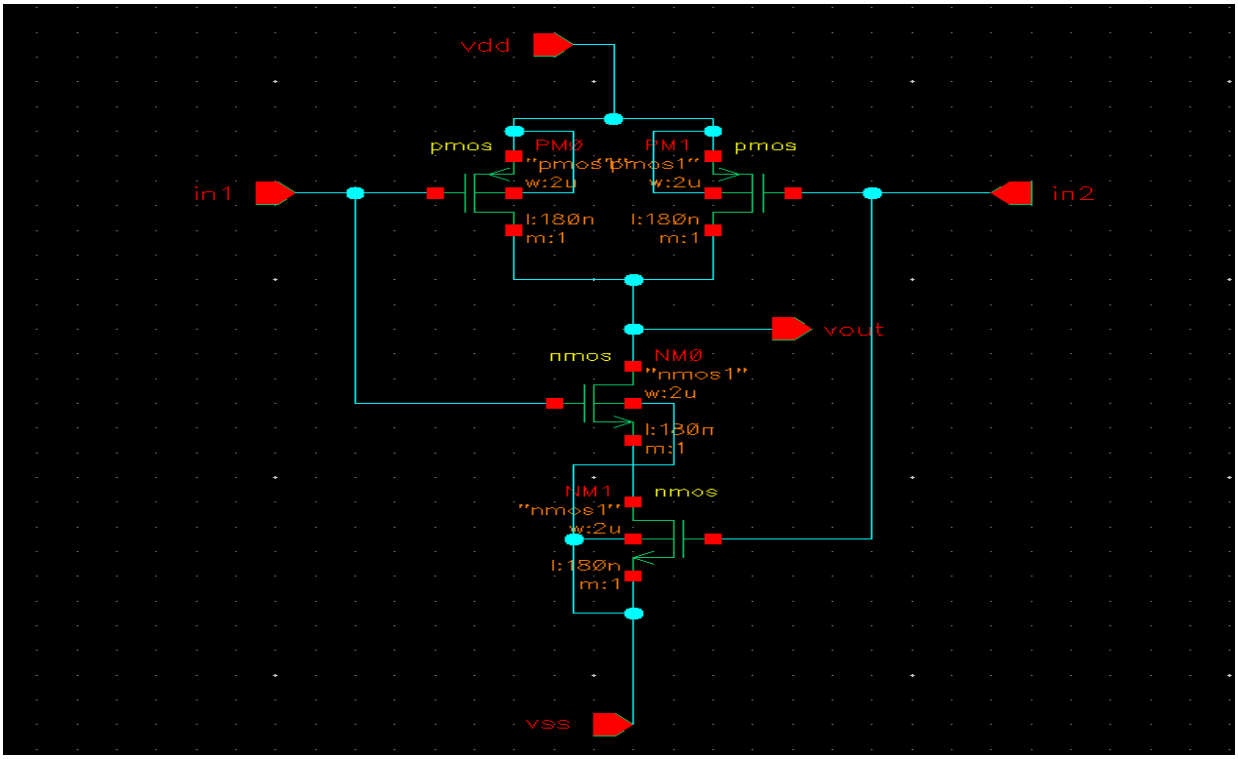


table of components for building the nand gate schematic.

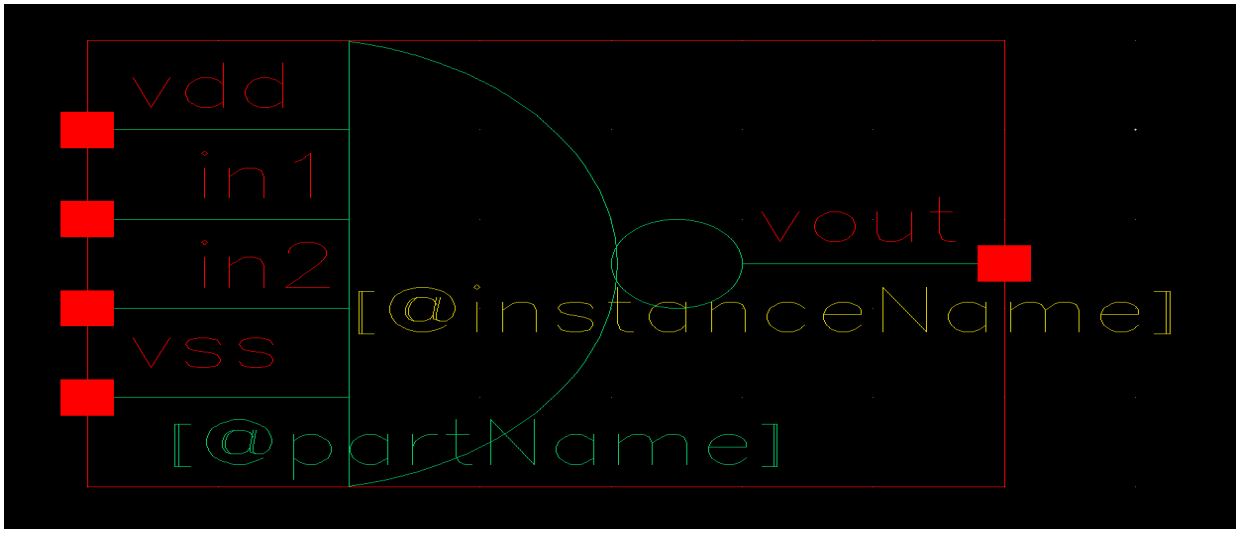
Library name	Cell Name	Properties/Comments
gpdk180	Pmos	Model Name = pmos1,pmos2;
gpdk180	Nmos	Model Name =nmos1,nmos2;

- Type the following in the ADD pin form in the exact order leaving space between the pin names.

Pin Names	Direction
Vin1 vin2	Input
vout	Output
vdd vss	Input

Symbol Creation: To create a symbol for the nand gate

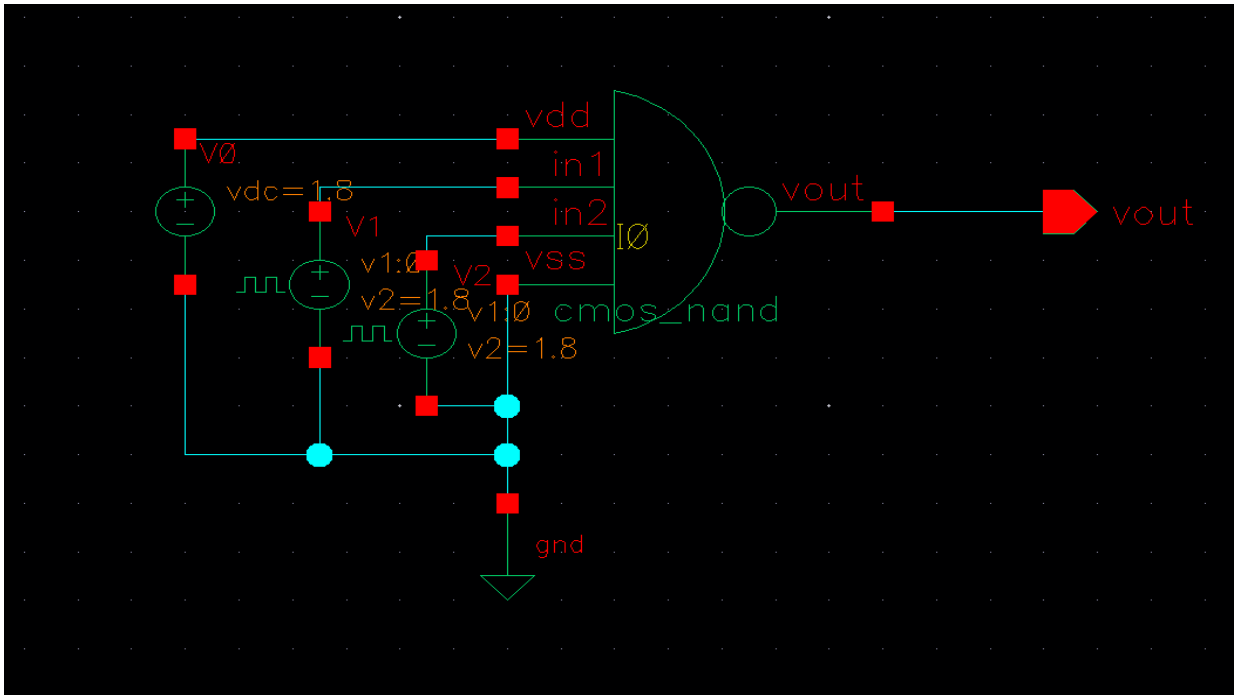
- Use the techniques learned in the Lab2.1 to complete the symbol of NAND gate



Building the NAND Test Design: To build NAND_test circuit using your NAND gate

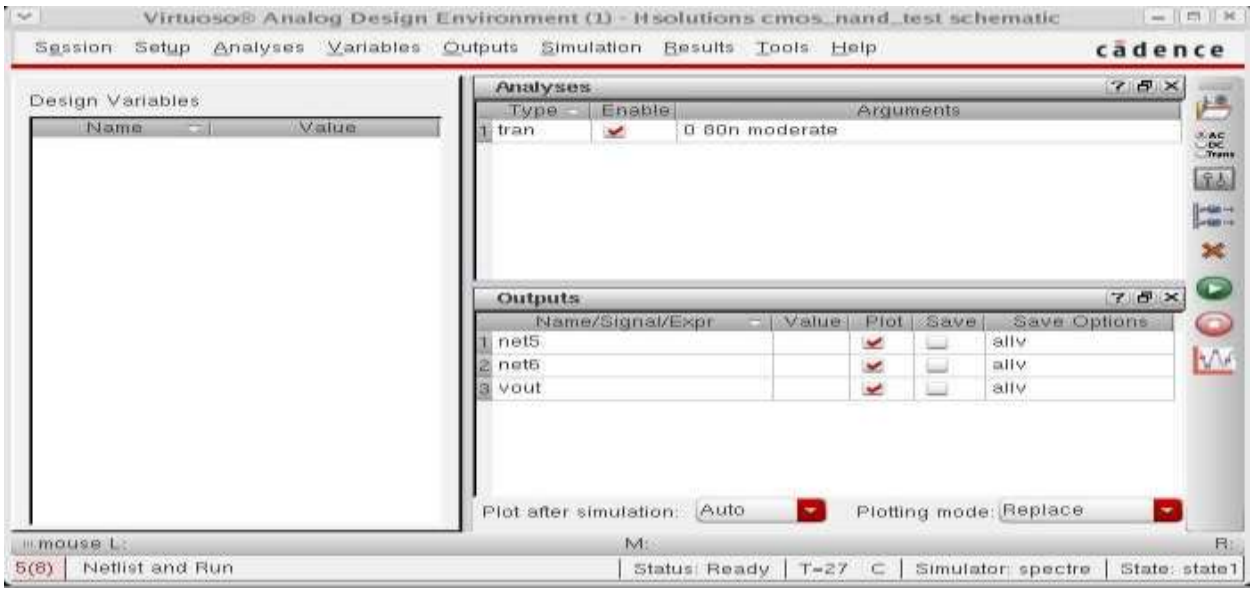
- Using the component list and Properties/Comments in the table, build the cs-amplifier_test schematic as shown below.

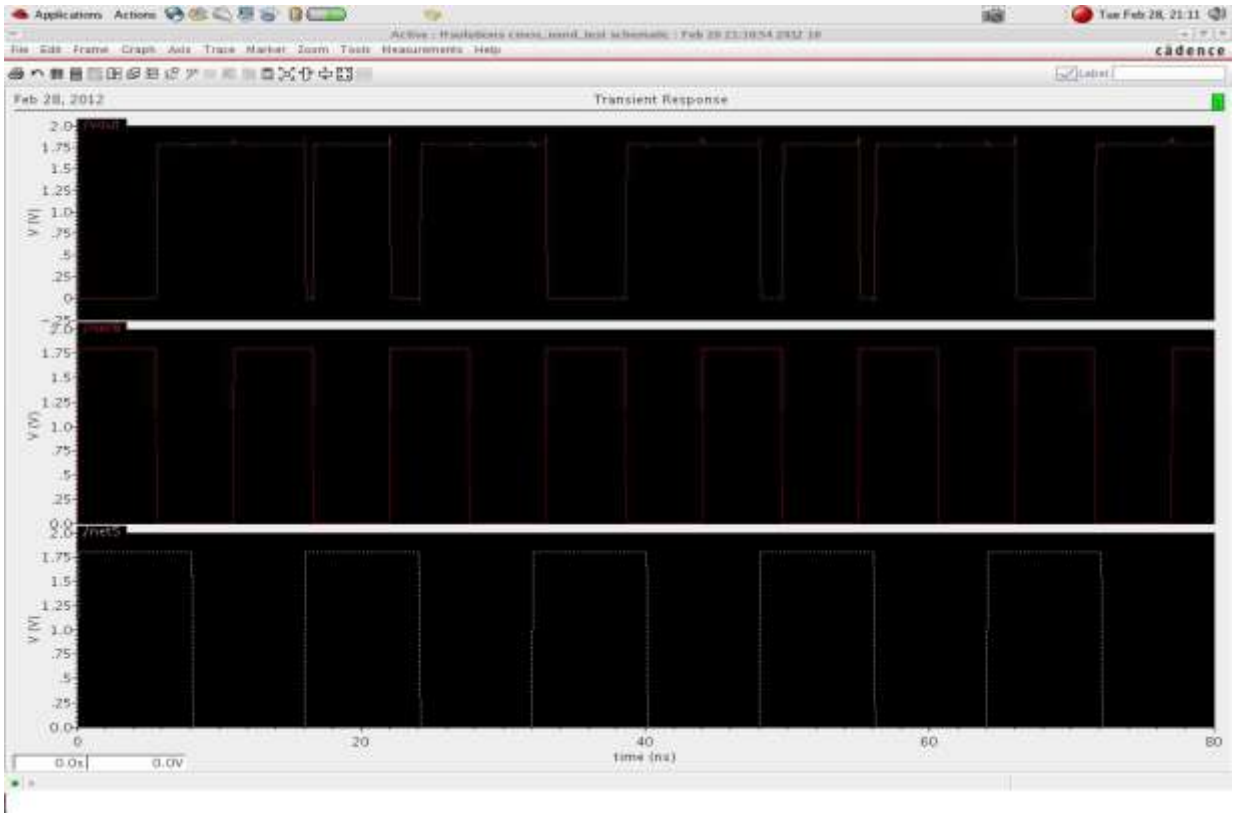
Library name	Cellview name	Properties/Comments
myDesignLib	cmos_nand	Symbol
analogLib	vpulse	Define pulse specification as In lab 2.1
analogLib	vdd,vss,gnd	vdd=1.8 ; vss= 1.8



Analog Simulation with Spectre: To set up and run simulations on the NAND gate design.

- Use the techniques learned in the Lab2.1 to complete the simulation of NAND gate, ADE window and waveform should look like below.





4.5 RESULT : Designed and verified dynamic characteristics of NAND gate.

4.6 PRE LAB VIVA QUESTIONS:

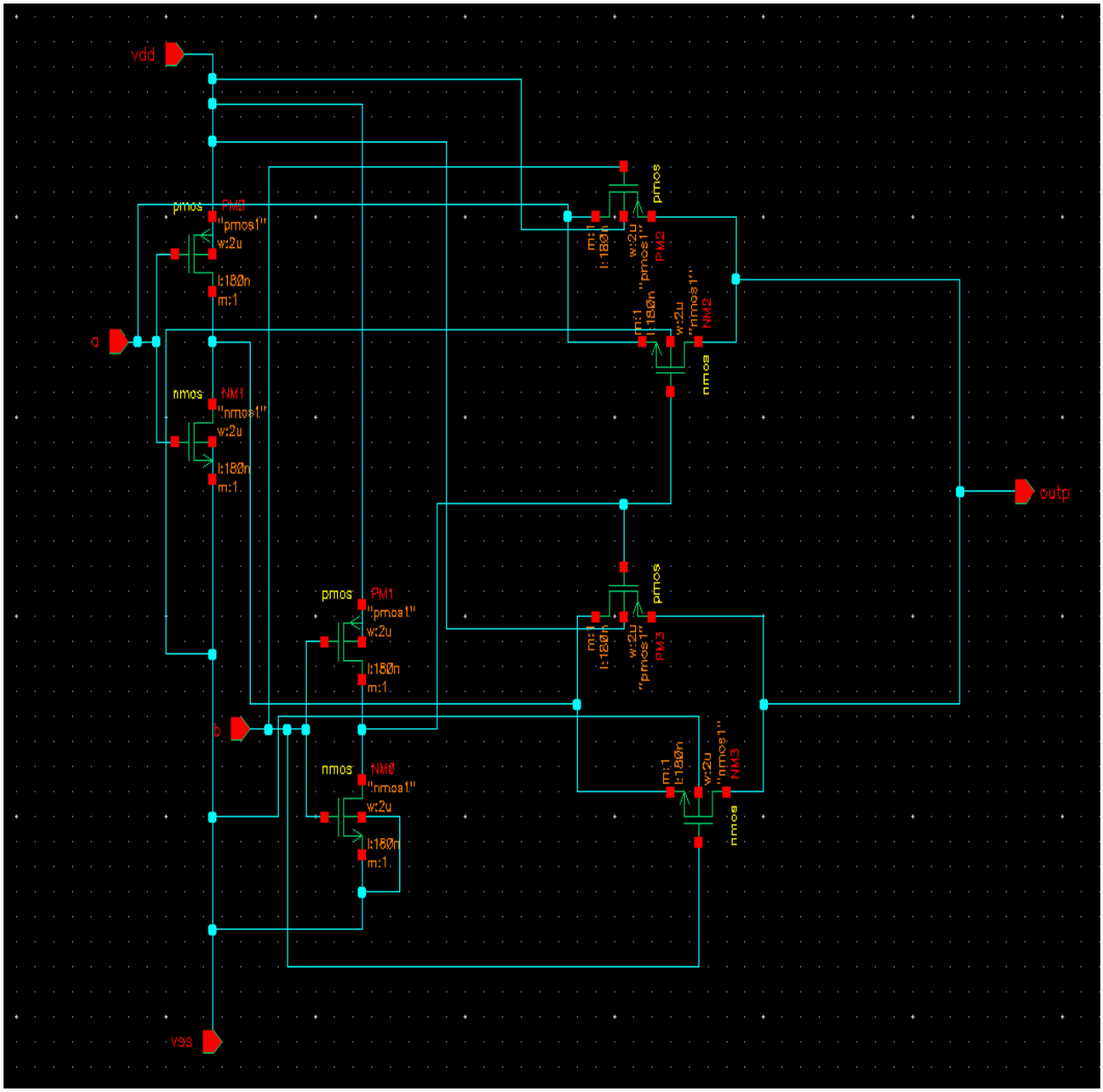
1. What is the function of NAND gate?
2. What is the function of NOR,XOR gate?

4.7 POST LAB VIVA QUESTIONS:

1. What is the advantage of NAND gate?
2. What do you observe from characteristics of NAND gate?
3. What do you observe from characteristics of XOR gate?

Design of XOR GATE

Schematic Capture



Schematic Entry: To create a new cell view and build A XOR gate

- Use the techniques learned in the Lab2.1 to complete the schematic of XOR gate. This is a table of components for building the XOR gate schematic.

Library name	**Cell Name**	**Properties/Comments**
gpdk180	Pmos	Model Name = pmos1,pmos2,pmos3,pmos4;
gpdk180	Nmos	Model Name =nmos1,nmos2,nmos3,nmos4;

- Type the following in the ADD pin form in the exact order leaving space between the pin names.

Pin Names	**Direction**
Vin1 vin2	Input
vout	Output
vdd vss	Input

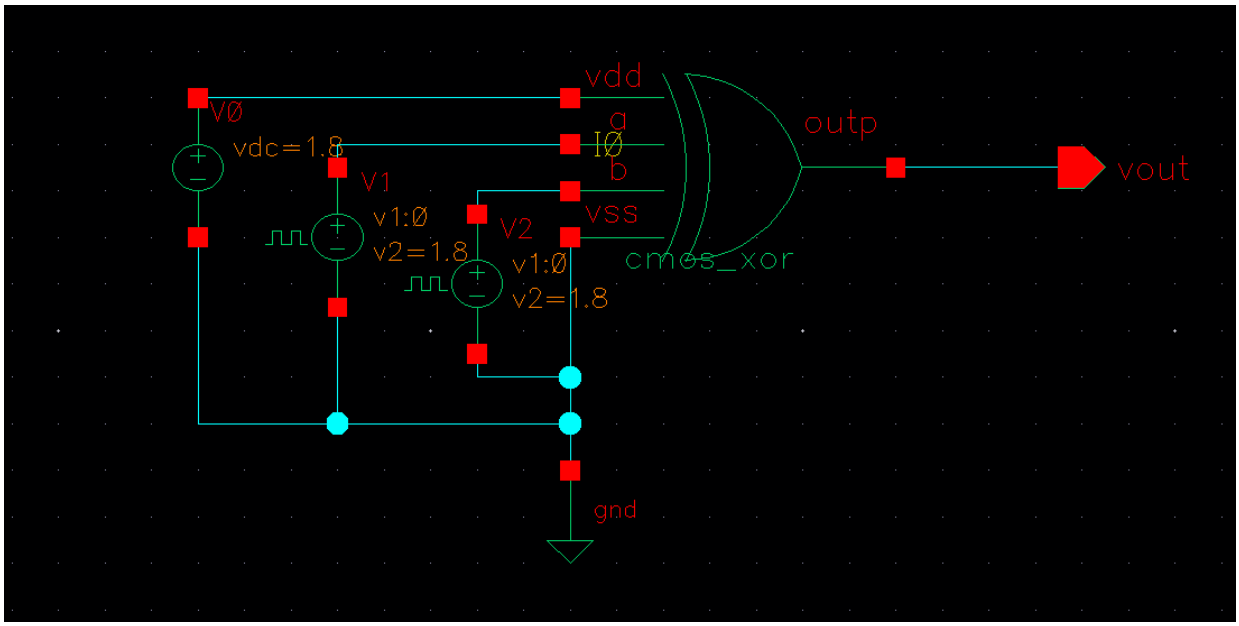
Symbol Creation: To create a symbol for the XOR gate

- Use the techniques learned in the Lab2.1 to complete the symbol of XOR gate

Building the XOR Gate Test Design: To build cmos_xor_test circuit using your cmos_xor

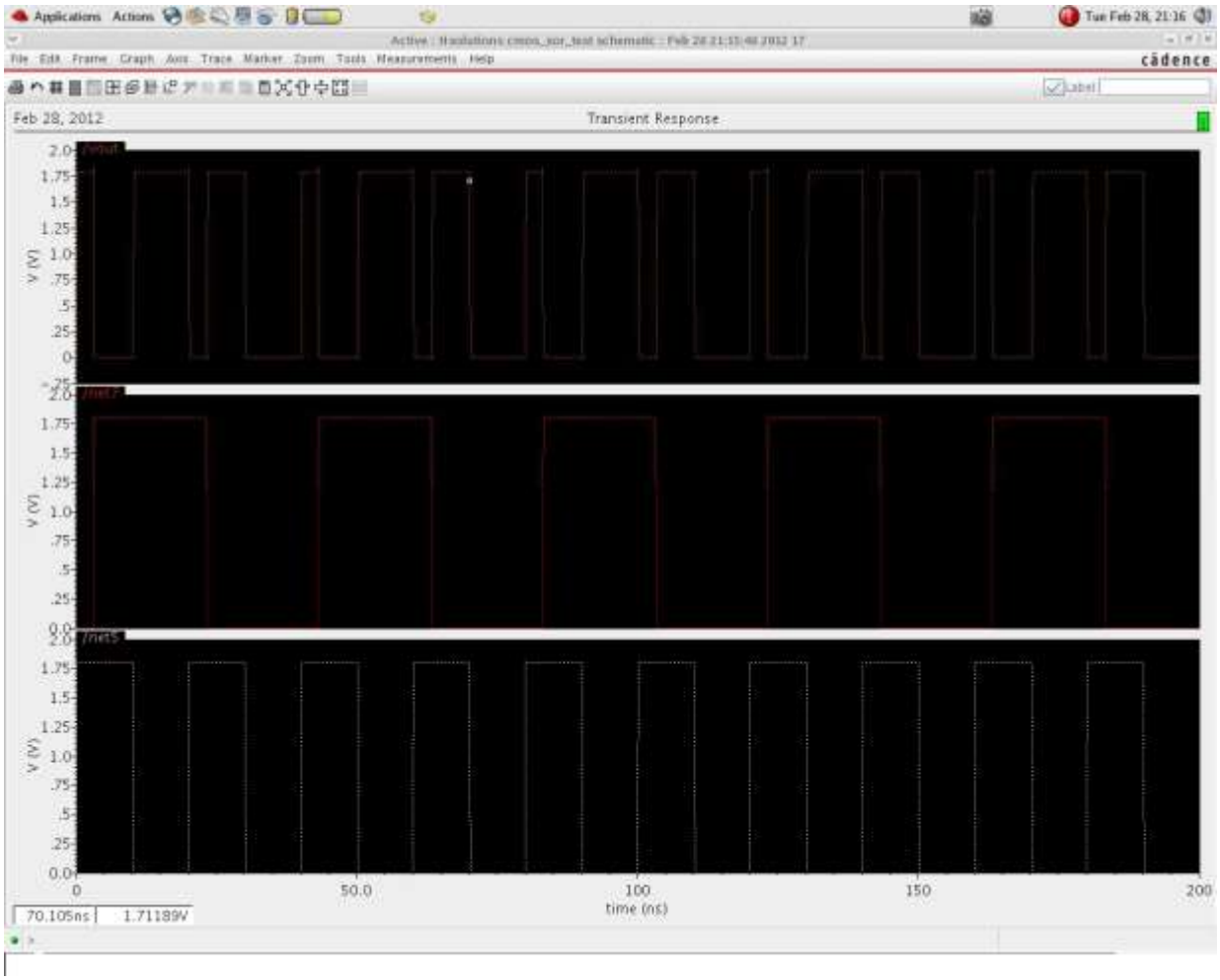
Using the component list and Properties/Comments in the table, build the cmos_xor_test schematic as shown below.

Library name	Cellview name	Properties/Comments
myDesignLib	cmos_XOR	Symbol
analogLib	vpulse	Define pulse specification as In lab 2.1
analogLib	vdd,vss,gnd	vdd=1.8 ; vss= 1.8



Analog Simulation with Spectre: To set up and run simulations on the XOR gate design.

- Use the techniques learned in the Lab2.1 to complete the simulation of XOR gate, ADE window and waveform should look like below.



RESULT : Designed and verified dynamic characteristics of XOR gate.

EXPERIMENT NO: 5

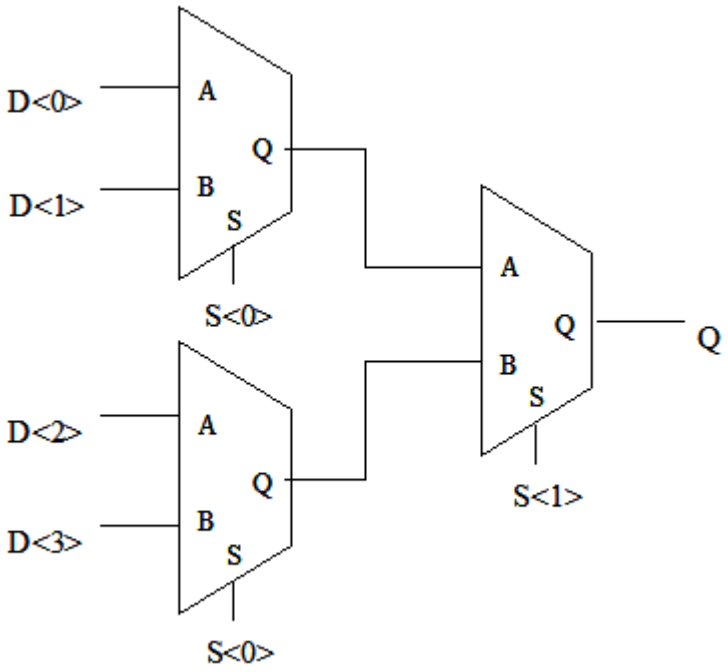
5.1 AIM: To design and plot the characteristics of a 4x1 digital multiplexer using pass transistor logic

5.2 LEARNING OBJECTIVE: To understand how to design and plot the characteristics of a 4x1 digital multiplexer using pass transistor logic

5.3 TOOLS REQUIRED: PC

CADENCE TOOLS

5.4 Theory: Consider a simple design example: a 4:1 logic multiplexer with 2 control inputs. The design is to be done by creating a 2:1 multiplexer with 1 control input, and then assembling three of them as shown below to create the 4:1 multiplexer.

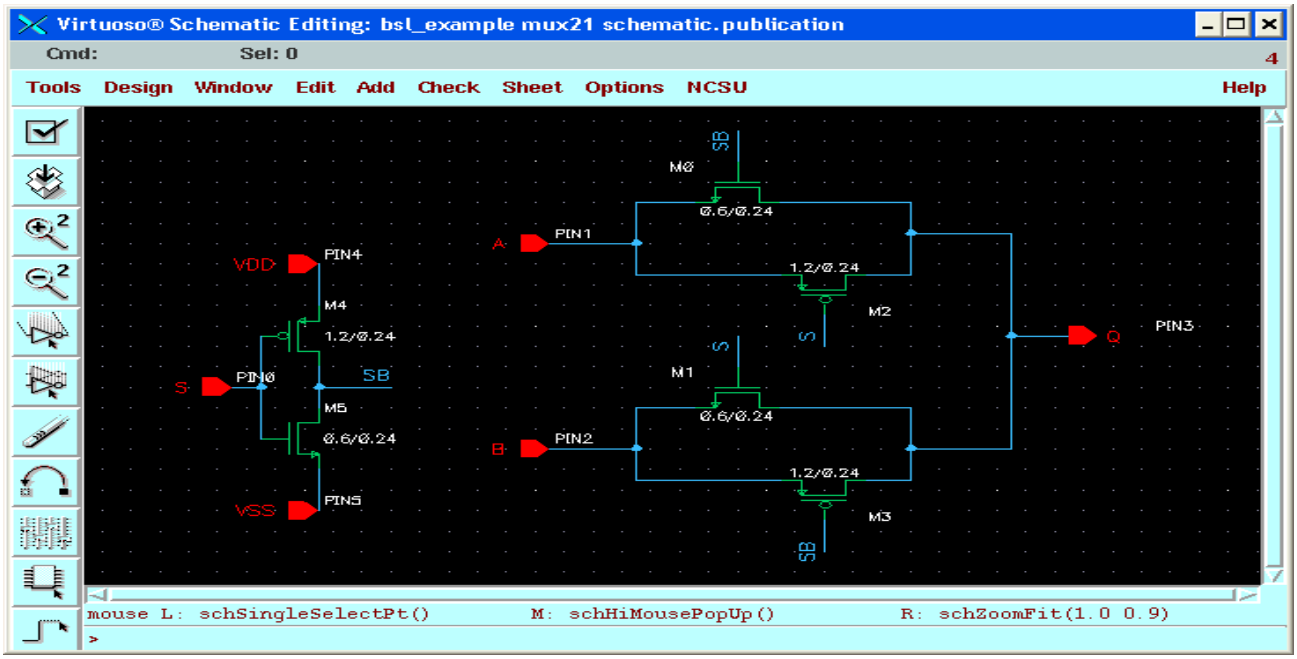


A few notations have been introduced here. First, we would like to consider the four inputs to be bits of a vector D<3:0>. But more subtle is the fact that we have used a symbol to represent a multiplexer in this schematic. There are no transistors. This is exactly what we want to do in Cadence. When we design the 2:1 multiplexer, we will create a transistor schematic and a polygon layout as you are already familiar with, but we will also create a "symbol" view that looks like the symbols used above. Then, when we create higher levels of schematics, such as the 4:1 mux, we can instantiate the 2:1 schematics, the same as we would do in the layout. So the design flow is as follows:2:1 Mux.

4.4 Procedure:

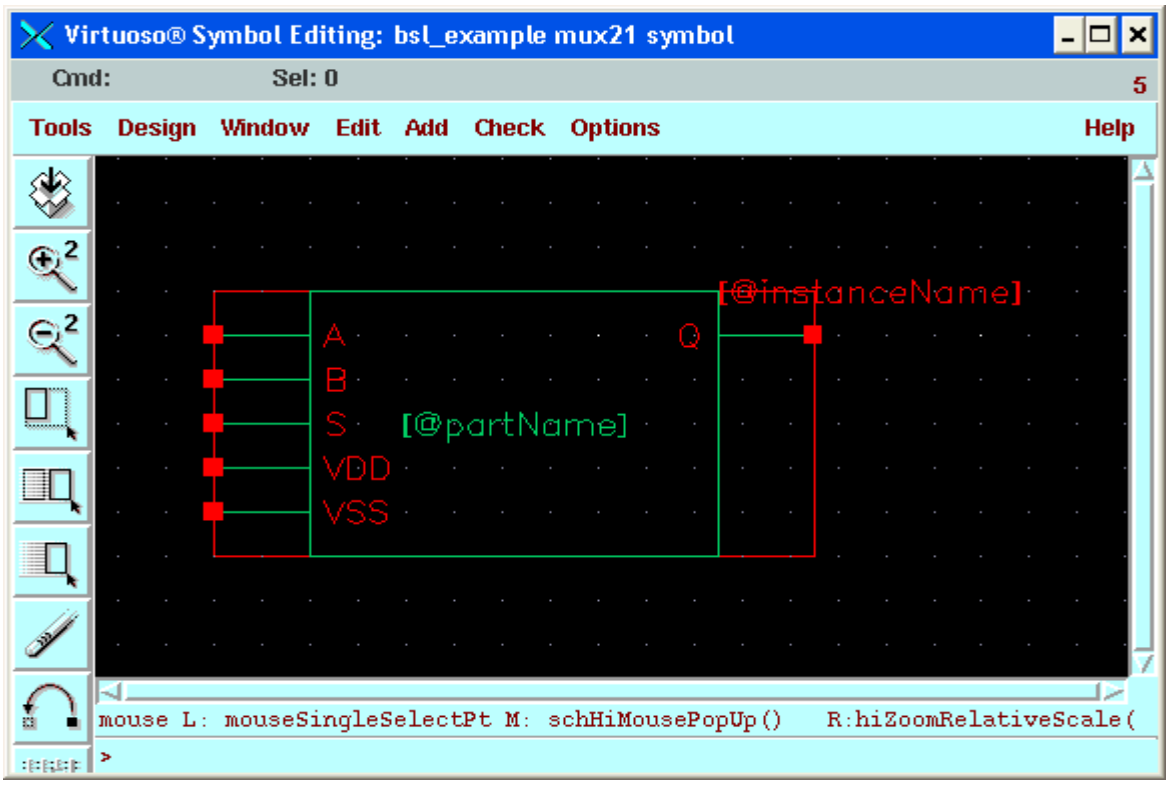
- Create transistor level schematic for 2:1 mux using transistor schematic symbols, run simulations to verify design
- Create layout for 2:1 mux by instantiating transistors, check DRC, LVS to verify layout

- Create symbol for 2:1 mux 4:1 Mux
- Create transistor level schematic for 4:1 mux using 2:1 mux schematic symbols, run simulations
- Create layout for 4:1 mux by instantiating 2:1 mux layouts, check DRC, LVS
- Create symbol for 4:1 mux
- The design flow is repeated in the same manner for each cell in the hierarchy, and this procedure can be repeated indefinitely to create very large/complex designs.
- Below is shown the schematic view for the 2:1 mux in this example.
- Remember to set the I/O type of your pins to either input or output as appropriate, and to do "Check and Save" on your schematic before creating the symbol.
- Note that the VDD and VSS pins are needed for the connections to the bulk terminals of the NMOS and PMOS devices.
- Even though they don't appear connected in the schematic, they are connected by reference when their names were used for the "bulk node" field when instantiating the transistors.
- You would also have to physically make these connections in the layout.
- Note that we have used a very useful feature in the schematics editor.
- Rather than explicitly wiring control signals S and SB around, we simply create wire stubs and label them S or SB appropriately.
- Cadence knows that all nets with the same name are considered connected. Labels are created with keystroke "l", and you must click directly on the wire being labeled when placing the labels.

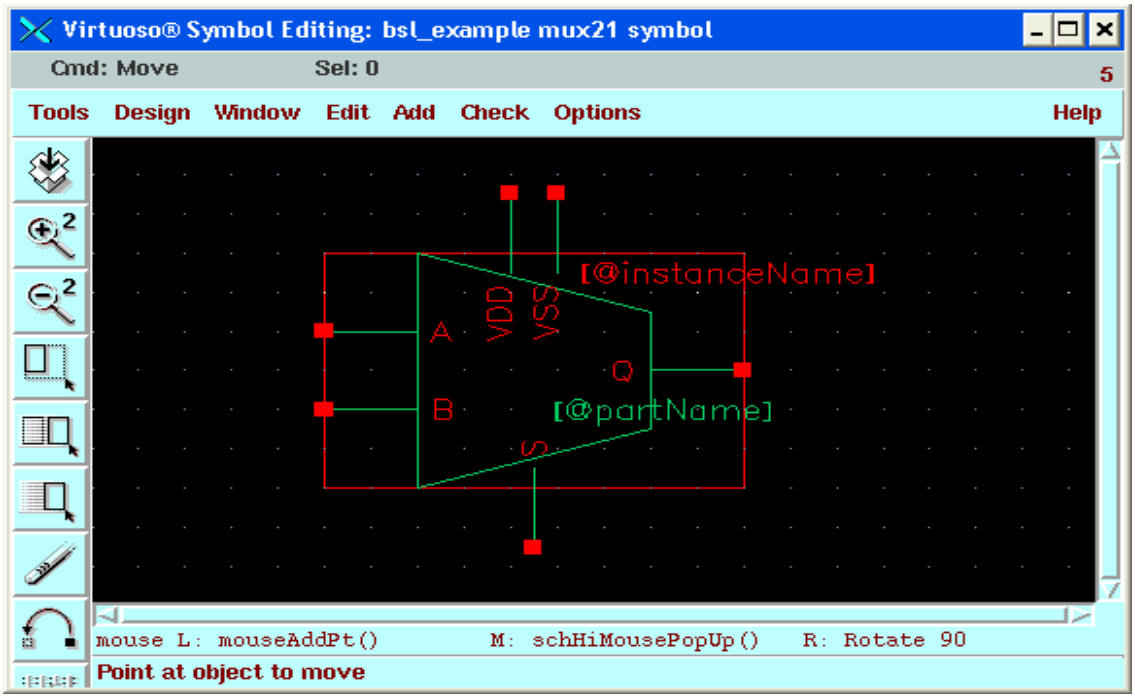


- Cadence can generate the symbol view for you automatically from the schematic.

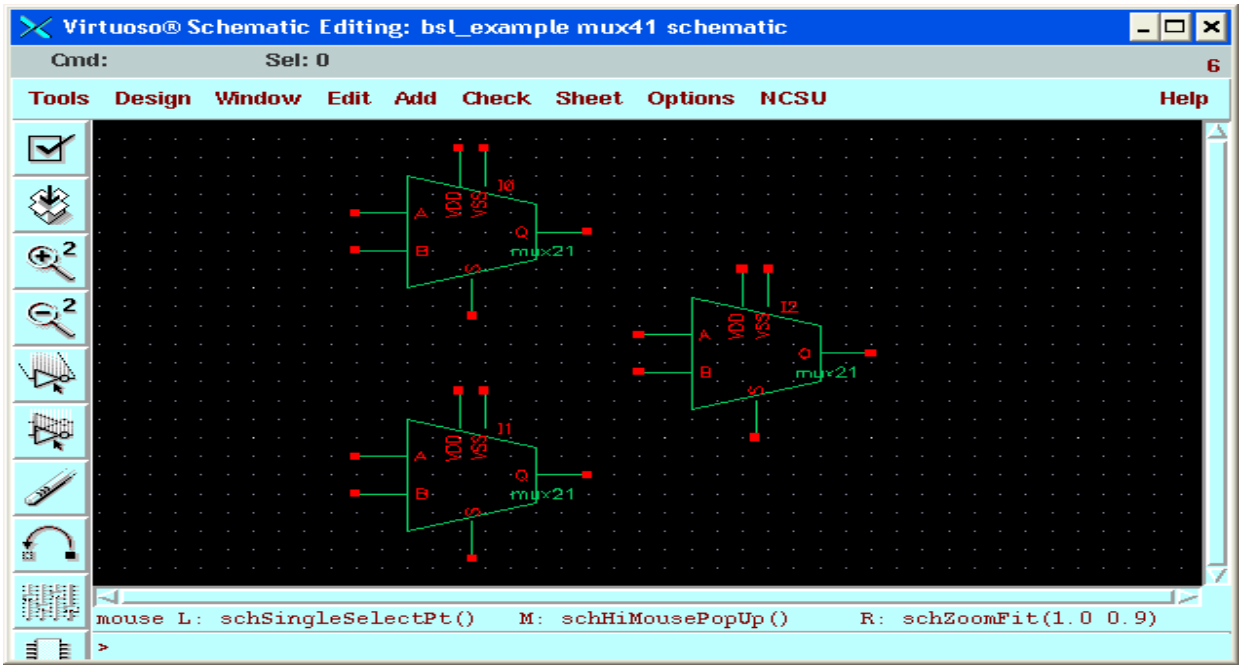
- In the Design menu, chose Create Cellview -> From Cellview and select symbol from schematic.
- It will open a dialog box asking how to position the pins in the generated symbol.
- You can just hit OK for now - it will put input pins on the left and output pins on the right by default, and you can edit the symbol later if you don't like it.
- The auto generated symbol view looks like this:



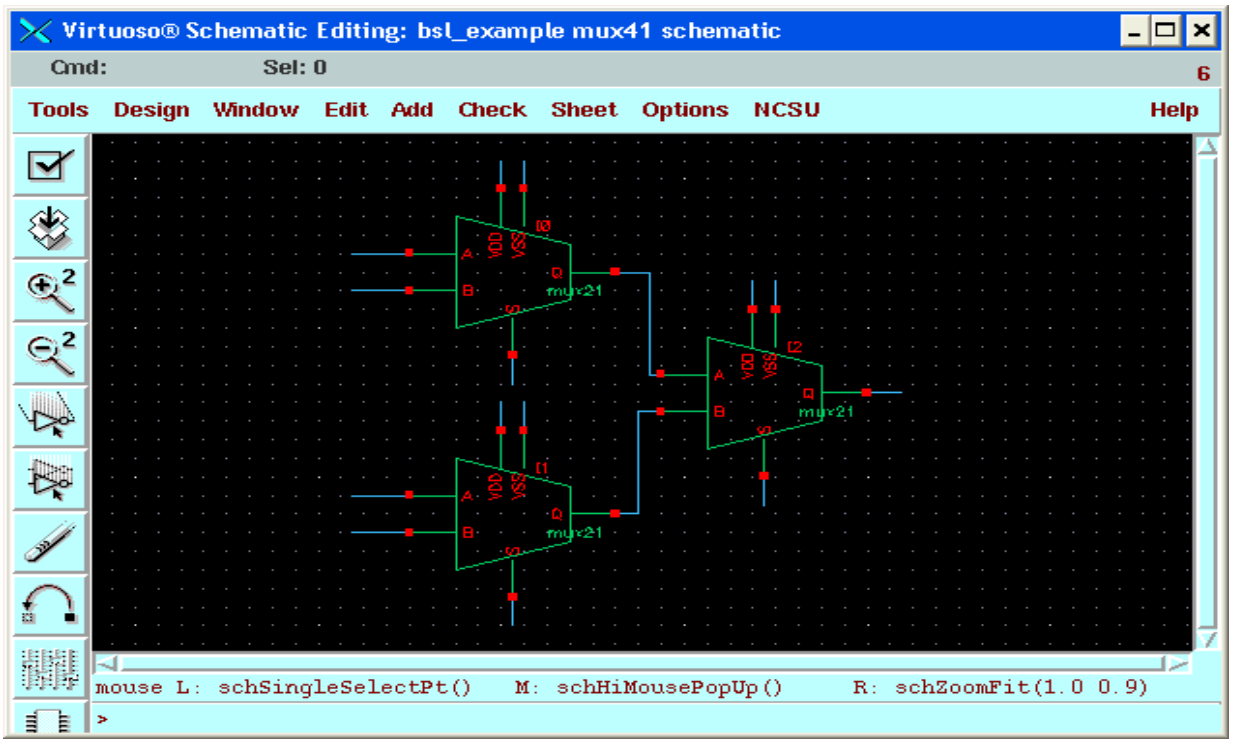
- We could just keep the symbol above, but we might like to make it look like a trapezoid so that it is more recognizable when we use it in other layouts, and separate the power supply pins from the inputs.
- We can do this by deleting the green box and redrawing four lines in the shape of a trapezoid. All of these shapes are just for visual purposes - they have no meaning in terms of electrical design.
- The only part of the symbol that is really important to the design are the red squares - these are the pins, and they are where you will connect wires to when drawing schematics.
- You can move these around, but be careful not to delete or rename them.
- It is always good to perform a "Check and Save" after making edits - it will make sure that you have exactly the same pin names in the symbol and the schematic.
- Here is the modified symbol view:



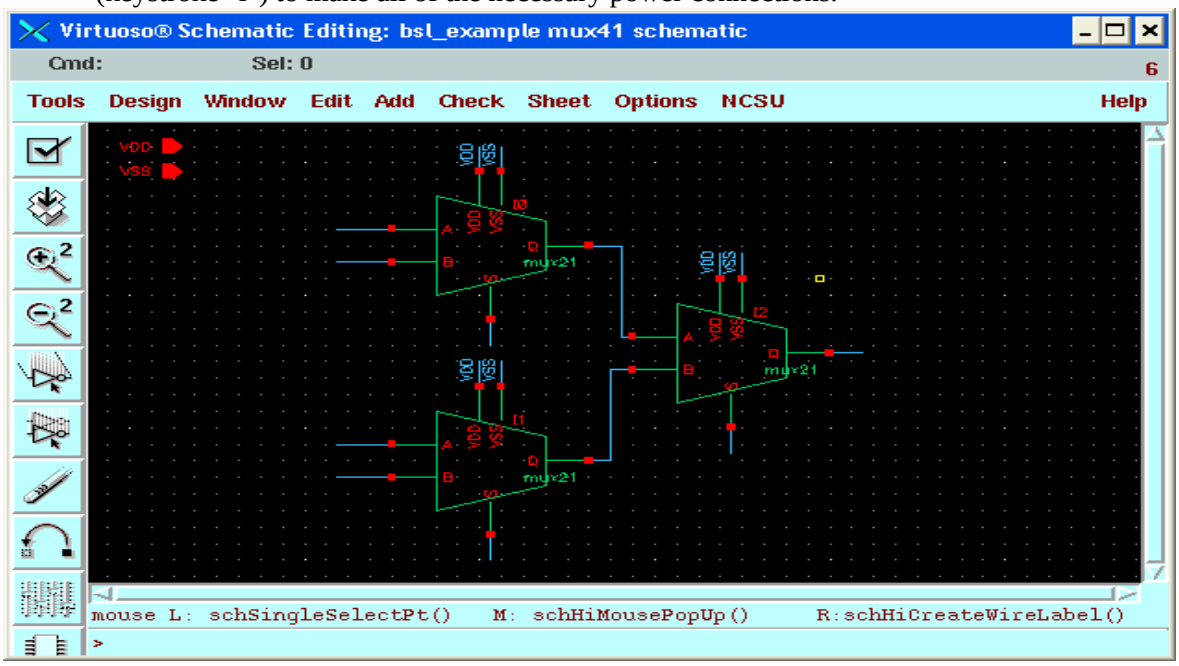
- Now we are ready to create the schematic for the 4:1 mux.
- Start by creating a new schematic view as usual. Then chose "instantiate", and browse for your 2:1 mux cell.
- Click three times in your schematic to instantiate three copies of the mux.
- You should have "symbol" selected as the cellview when doing this instantiation.
- You will then have something like this:



- When creating layouts, you will usually perform a parallel task where you instantiate 3 copies of the 2:1 mux layout into the 4:1 mux layout, but this is not shown here.
- Next we connect the muxes according to the original schematic drawing at the top of this page:

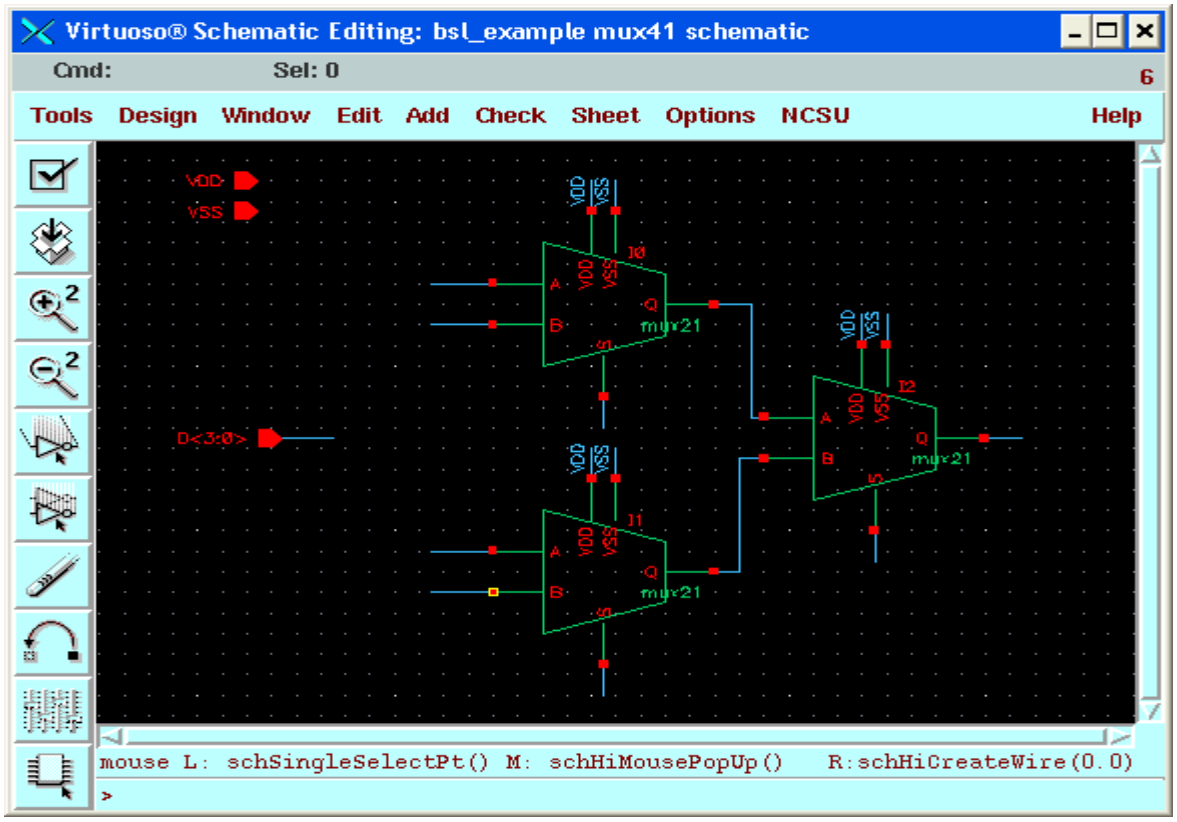


- Just as in the 2:1 mux, we need to name all of the pins.
- Let's consider power first. We create the typical pins called VDD and VSS, and again use the label tool (keystroke "l") to make all of the necessary power connections.



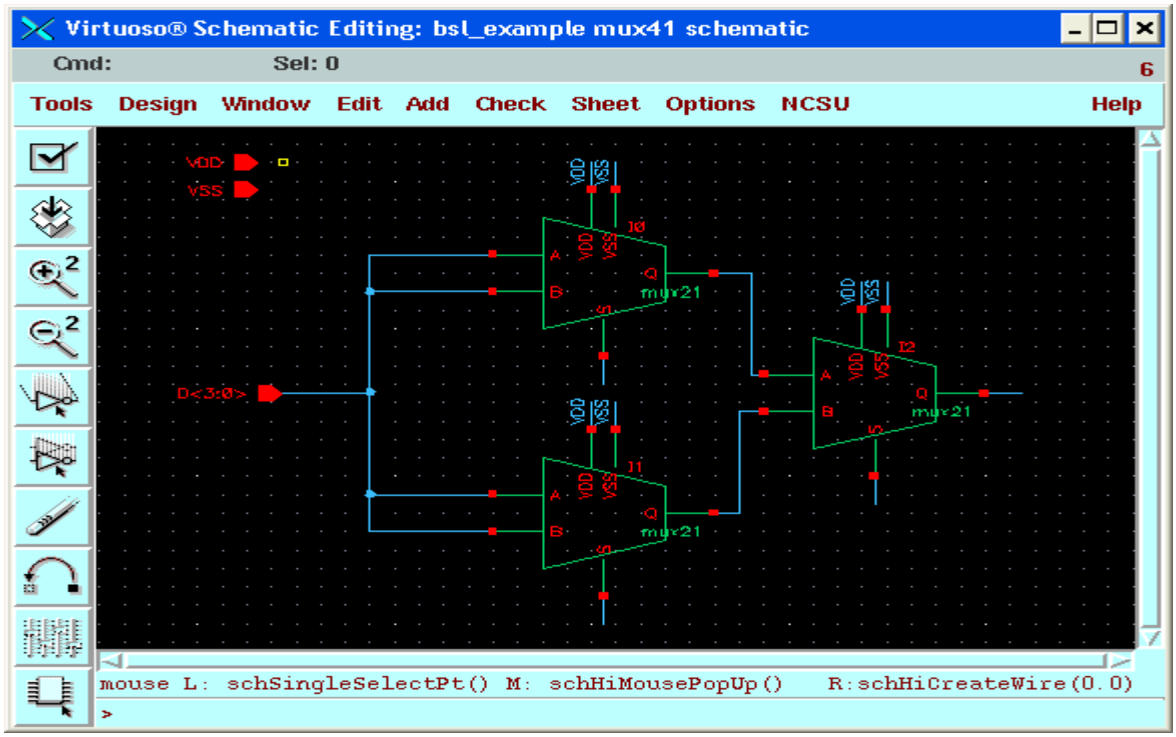
- Now consider the input pins. We could just create four input pins called A, B, C, and D and connect them to the four inputs, but this starts to become cumbersome as the number of inputs grows.

- We would like to have a single pin designation for "Inputs 0 through 3". This is done in Cadence with bus notation.
- When creating your input pins, select "input" as the I/O type, but enter D<3:0> as the pin name, and place this pin in the schematic as shown.

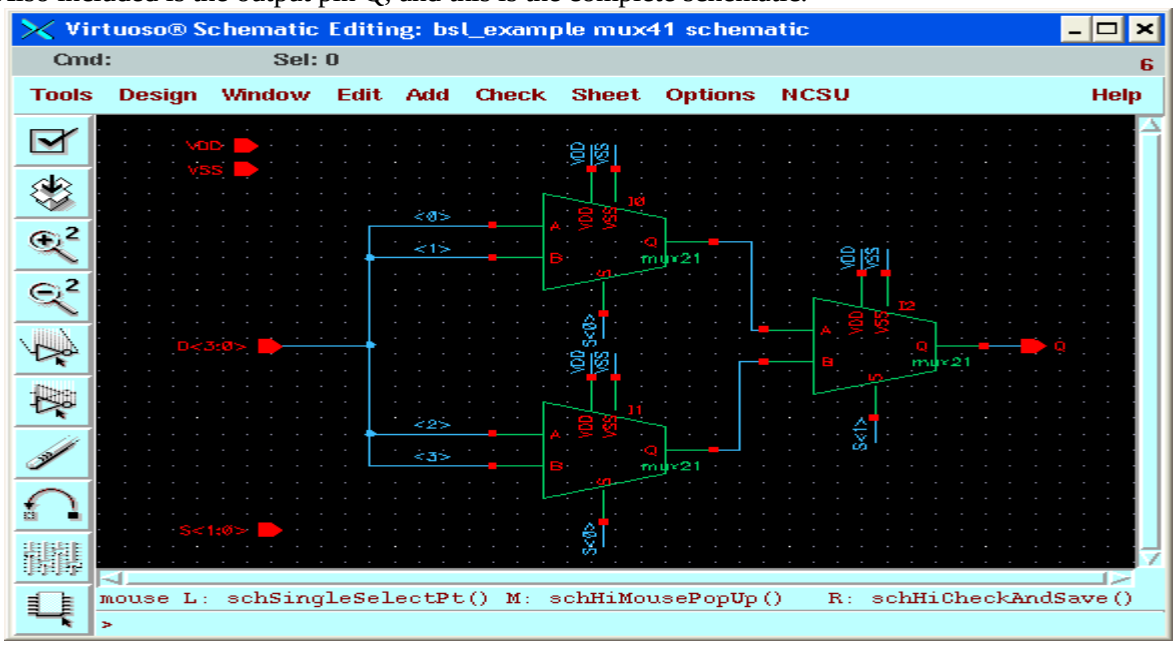


- The schematic wire connected to pin D<3:0> actually represents *four* wires. This is very important to remember.
- When you do the layout, there must actually be four physical wires drawn that correspond to the four schematic wires.
- Also, we only want to connect one wire to each of our 2:1 mux inputs, as they are single signal inputs.
- If we just wire D<3:0> to all inputs as shown below, there is no way to know which input should go to which mux.
- If you perform a "Check and Save", you will get warnings about this.
- This ambiguity is resolved by creating labels on what we want to be single wires indicating which signal from D should be used.
- Again, this is done with the label command (keystroke "l"). Labeling the wires <0>, <1>, <2>, and <3> as shown below designates which signal from vector D connects to which mux.

- There is a shortcut for this: enter <0:3> as the label name and check the "bus expansion on" box before placing the labels, and the next four clicks place labels <0>, <1>, <2>, and <3>.
- We can do the same thing for the two control bits and name the pin S<1:0> as shown below, but this time use wire labels S<0> and S<1> to indicate both the bus name and the bit number at the same time.

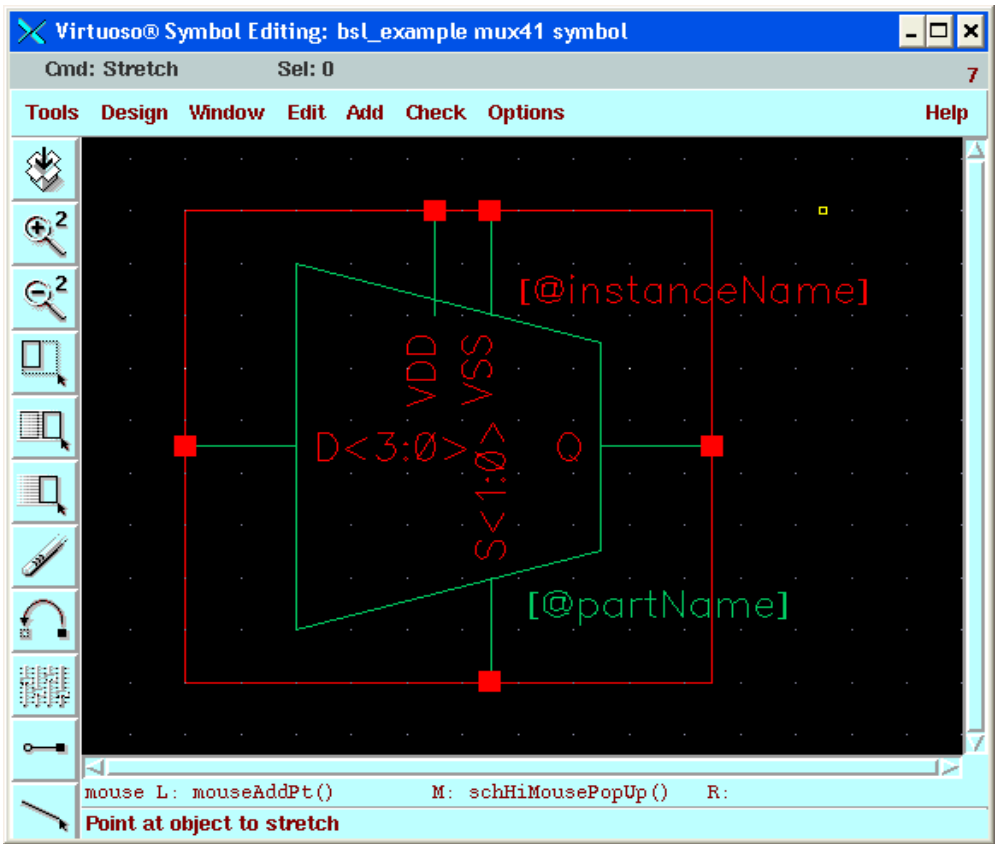


- Also included is the output pin Q, and this is the complete schematic.



- Now that the 4:1 schematic is done, we can do all the normal things: export the netlist for simulation, create the layout and run LVS to compare against this schematic, etc.

- When you export the netlist, it will actually create a netlist with all 12 transistors connected appropriately.
- Also, we will need a symbol view for this 4:1 mux for when we want to use it in turn in even higher levels of the schematic hierarchy.
- This is done the same way as for the 2:1 mux, by selecting Design->Create Cellview->From Cellview, and optionally editing the resulting symbol shapes. The result is something like this:



Schematic Capture

5.5 RESULT: Did the simulation and observed the MUX operation

5.6 PRE LAB VIVA QUESTIONS:

1. What is the function of PASS Transistor?
2. What is the application of PASS Transistor?
3. What is the function of Transmission gate?

5.7POST LAB VIVA QUESTIONS:

1. Try To design MUX by using some other gates?
2. What do you observe from characteristics of MUX gate?
3. Design DE-MUX?

EXPERIMENT NO: 6

6.1 AIM: To design and plot the characteristics of a positive and negative latch based on multiplexers.

6.2 LEARNING OBJECTIVE: To understand design and plot the characteristics of a positive and negative latch based on multiplexers.

6.3 TOOLS REQUIRED: PC

CADENCE TOOLS

6.4 THEORY:

In the proposed DETFF(**Dual-Edge Triggered Flip-Flop**), positive latch and negative latch are connected in parallel as shown in Fig. These latches are designed using one transmission gate and two inverters connected back to back and the output of both the latches are connected to 2:1Mux as input. Mux is designed using one PMOS and one NMOS connected in series and gates are connected together and derived by the inverted CLK. Output of Mux is connected to the inverter for strengthening the output. Back to back connected inverters hold the data when transmission gate is *OFF* and at the same time Mux sends the latched data to the inverter to get the correct D at the output.

6.5 PROCEDURE:

- Follow the procedure that has been followed till now to make schematic and do analysis on simulation

Library name	**Cell Name**	**Properties/Comments**
gpdk180	nmos	Model Name = nmos1 (NM0, NM1) ; W= 3u ; L= 1u ; Body Type : Detached
gpdk180	nmos	Model Name =nmos1 (NM2, NM3) ; W= 4.5u ; L= 1u ; Body Type : Integrated
gpdk180	pmos	Model Name =pmos1 (PM0, PM2); W= 15u ; L= 1u ; Body Type : Integrated

results.

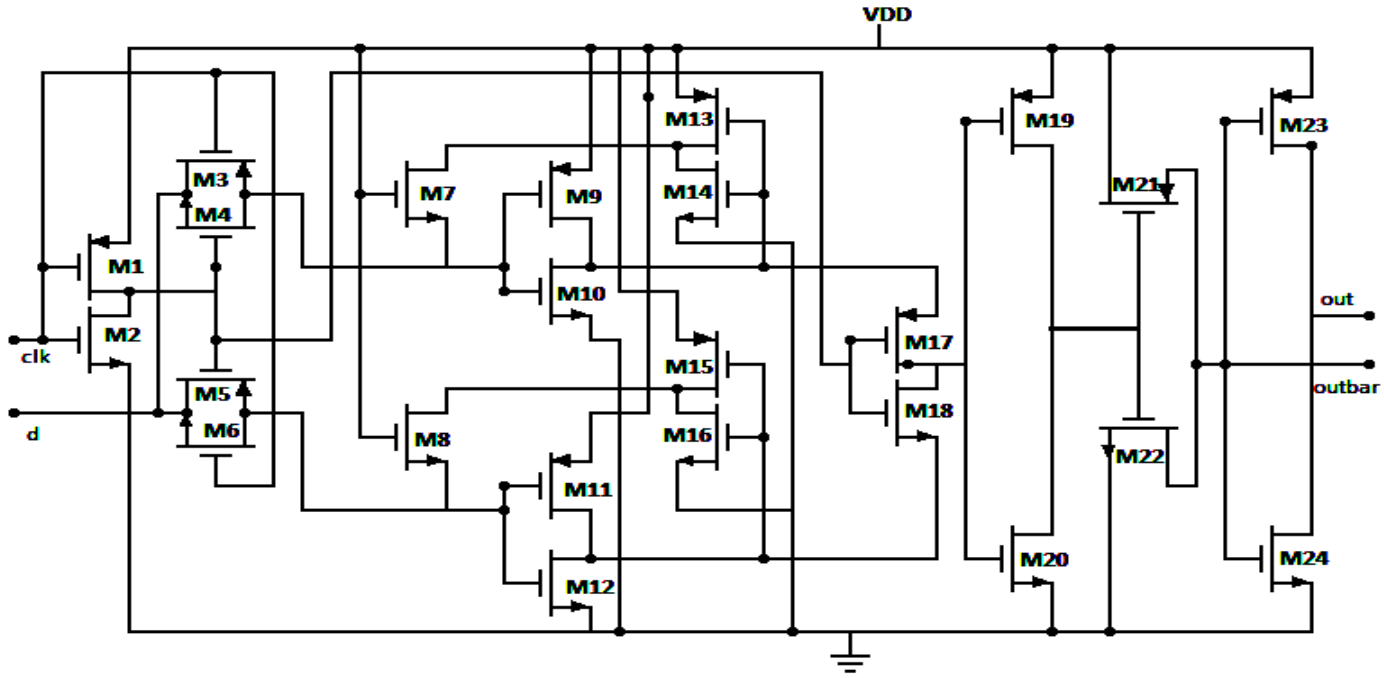
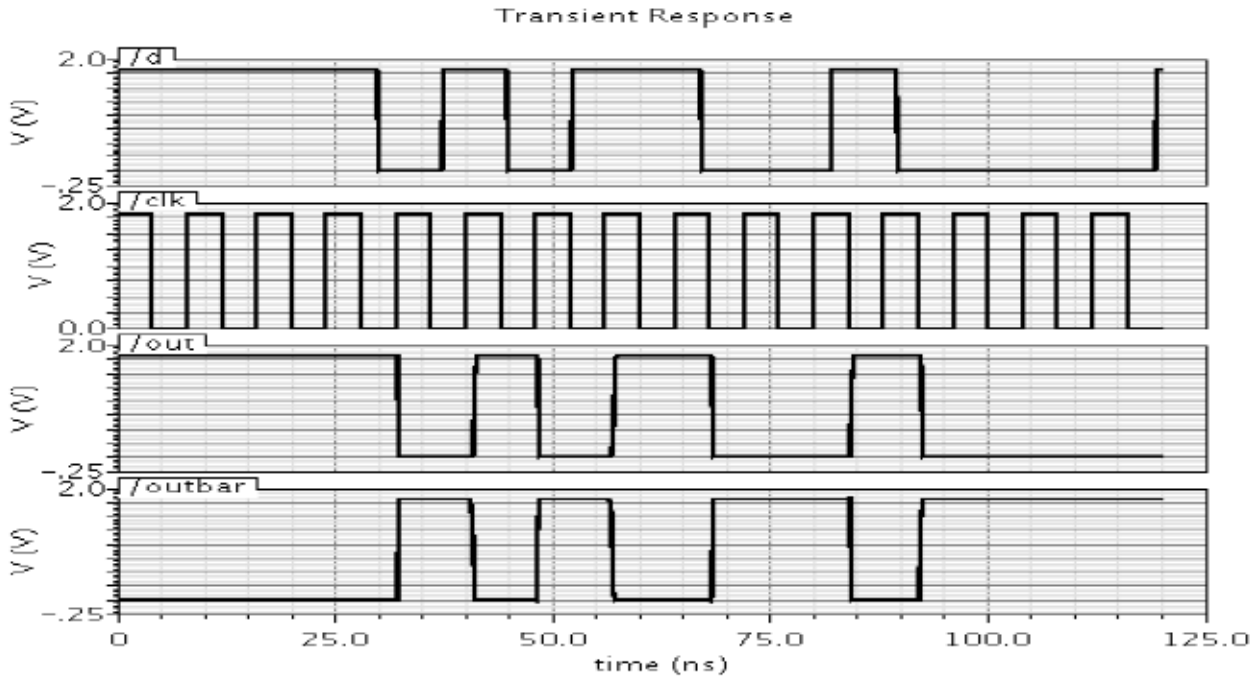


Table I
CMOS Simulation Parameters

Technology	180 nm
Min. Gate Width:	600 nm
Max. Gate Width:	1200 nm
MOSFET Model:	BSIM 3v3
Nominal Conditions:	V_{dd} = 1.8V, T=27 ^{0}C
Duty Cycle:	50 %
Nominal Clock Frequency:	125MHz



6.6 RESULT: Designed and plotted the characteristics of a positive and negative latch based on multiplexers.

6.7 PRE LAB VIVA QUESTIONS:

1. What is the function of MUX gate?
2. What is the application of MUX in real life?
3. What is the function of DE-MUX gate?

6.8 POST LAB VIVA QUESTIONS:

1. Try To design MUX by using some other gates?
2. What do you observe from characteristics of MUX gate?
3. Design DE-MUX?

EXPERIMENT NO: 7

7.1 AIM: To design and plot the characteristics of a master-slave positive and negative edge triggered registers based on multiplexers.

7.2 LEARNING OBJECTIVE: To understand characteristics of a differential amplifier

7.3 TOOLS REQUIRED: PC

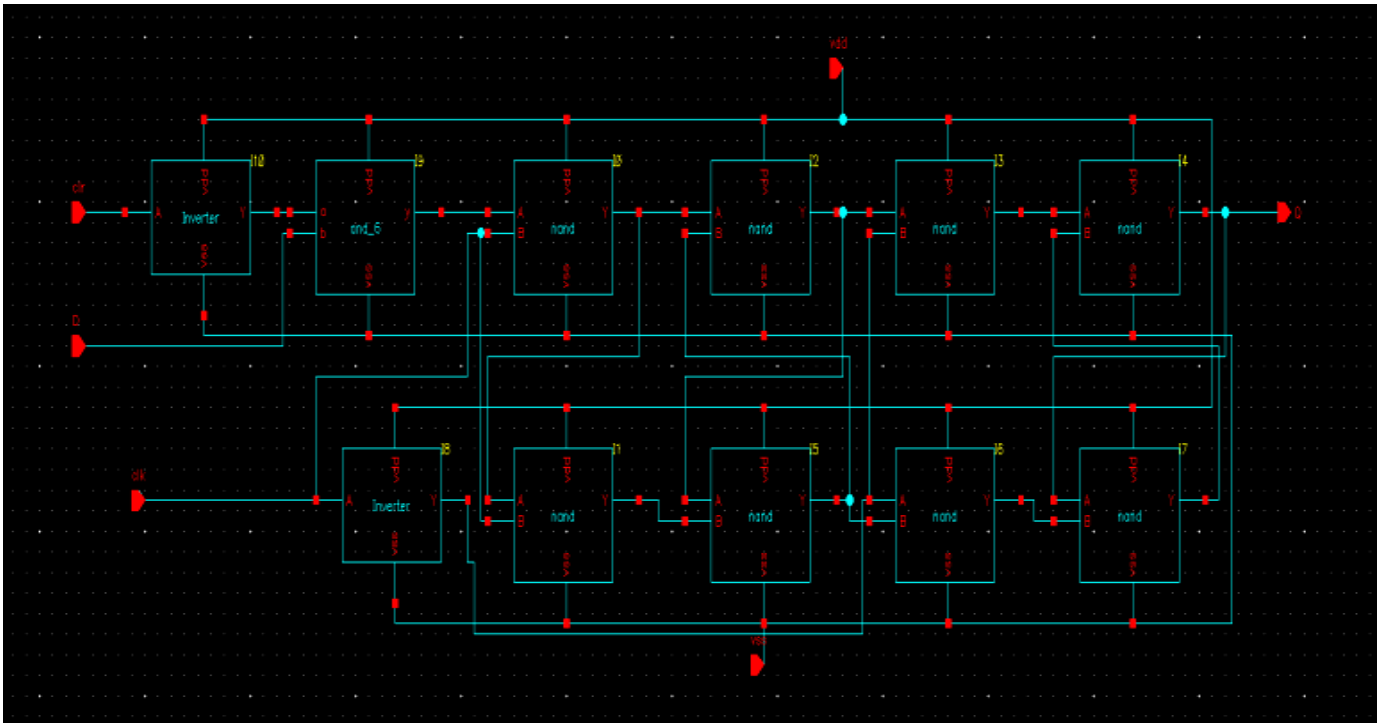
CADENCE TOOLS

7.4 THEORY:

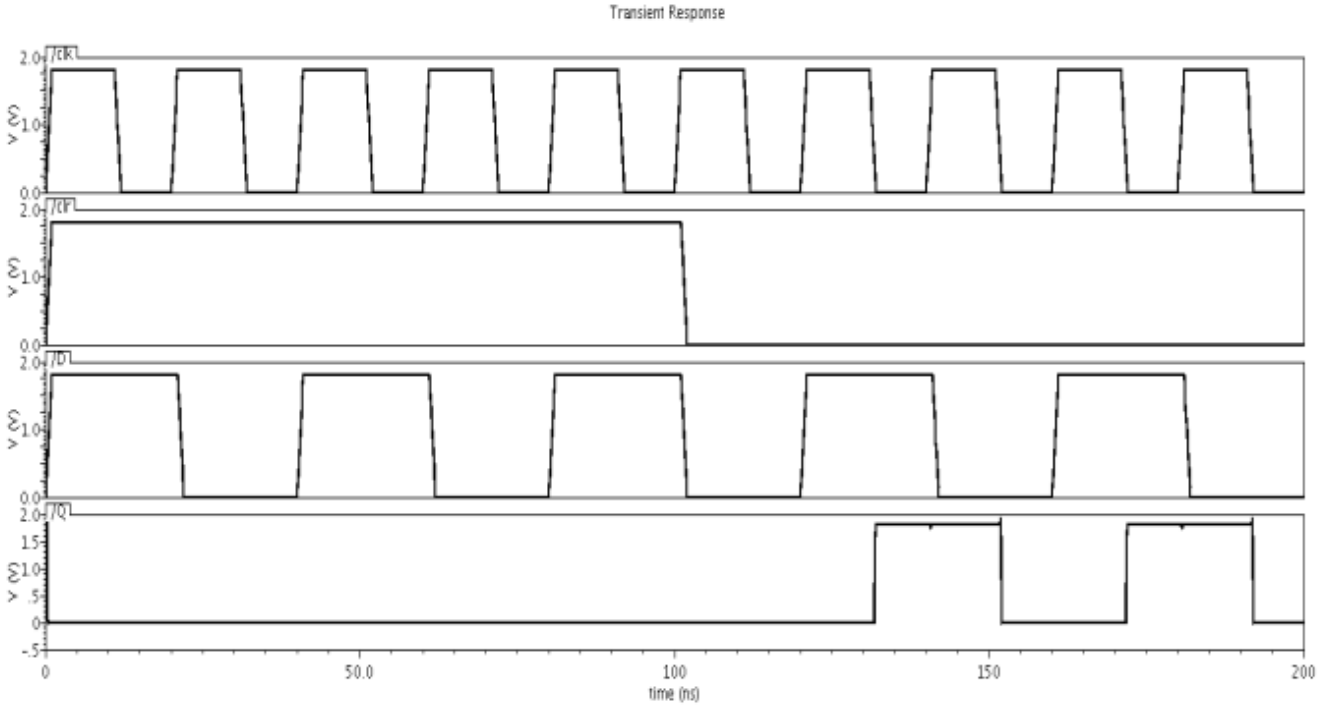
Master-Slave D Flip-Flop A master-slave D flip-flop is created by connecting two gated D latches in series and inverting the enable input to one of them. It is called master-slave because the second (slave) latch in the series only changes in response to a change in the first (master) latch [2]. The term pulse-triggered means that data is entered on the rising edge of the clock pulse, but the output does not reflect the change until the falling edge of the clock pulse. Master-slave flip-flops can be constructed to behave as a J-K, R-S, T or D flip-flop. The purpose of master-slave flip-flops is to protect a flip-flop‟s output from inadvertent changes caused by glitches on the input. Master-slave flip-flops are used in applications where glitches may be prevalent on inputs. The master-slave configuration has the advantage of being pulse-triggered, making it easier to use in larger circuits, since the inputs to a flip-flop often depend on the state of its output.

7.5 PROCEDURE:

Draw the Schematic and follow the steps what we did till now and do the simulation .



Simulation result:



7.6 RESULT: Studied the characterstics of master slave edge triggered register .

7.7 PRE LAB VIVA QUESTIONS:

1. What is the logical operation of D-Flipflop?
2. What is the logical operation of Master-Slave Flipflop?
3. What is the application of Flip-Flop?

7.8 POST LAB VIVA QUESTIONS:

1. Try To design Master-Slave flip flop using some other gates?
2. What do you observe from characteristics of Master-Slave flip flop ?

EXPERIMENT NO: 8

8.1 AIM: To study V-I characteristics of a differential amplifier

8.2 LEARNING OBJECTIVE: To understand V-I characteristics of a differential amplifier

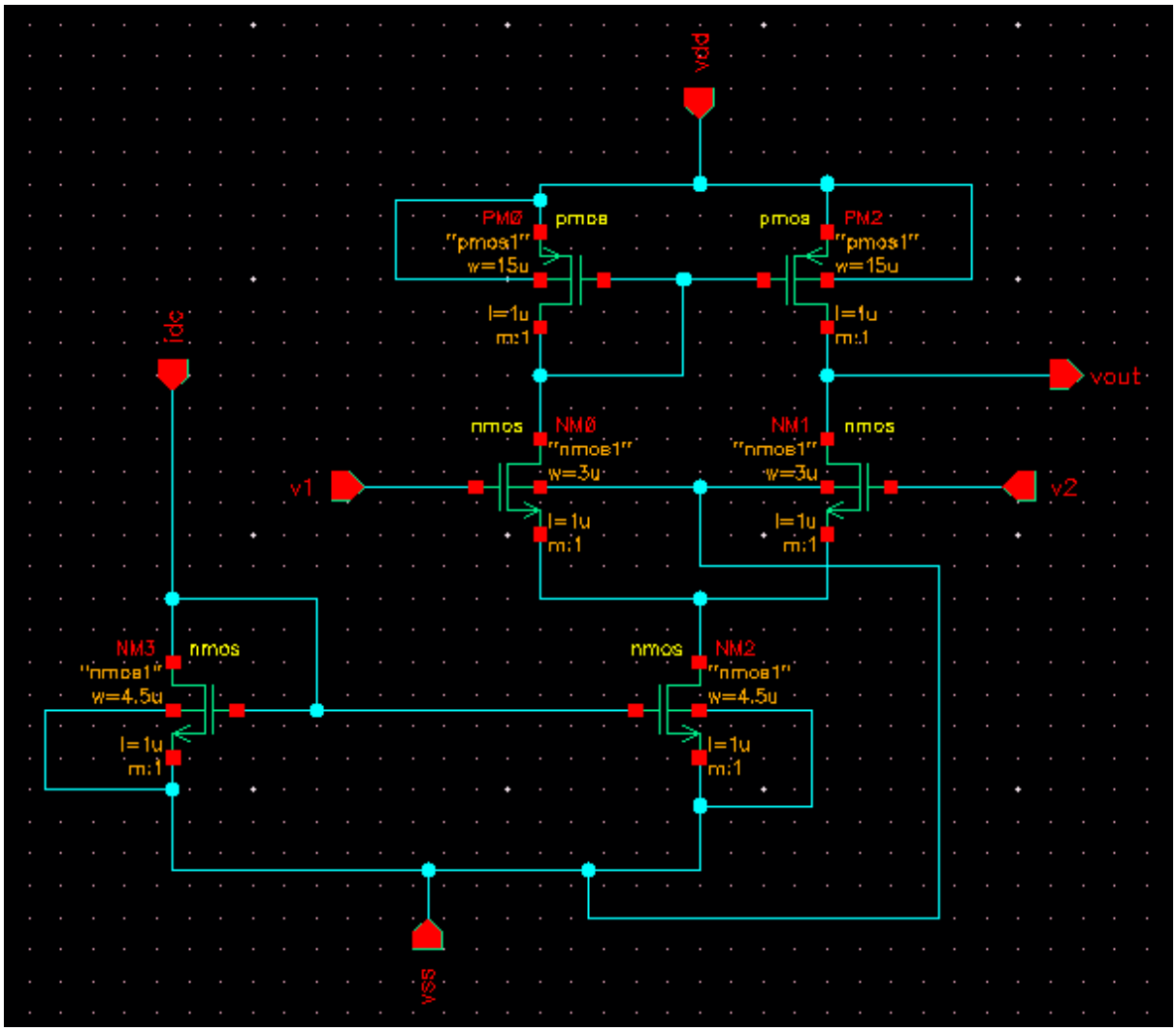
8.3 TOOLS REQUIRED: PC

CADENCE TOOLS

8.4 PROCEDURE:

Schematic Entry

Objective: To create a new cell view and build Differential Amplifier



Schematic Capture

Creating a Schematic cellview

Open a new schematic window in the **myDesignLib** library and build the Differntial_Amplifier design.

1. In the CIW or Library manager, execute **File – New – Cellview**. Set up the Create New file form as follows:

3. Click **OK** when done. A blank schematic window for the design appears.

Adding Components to schematic

1. In the Differential Amplifier schematic window, execute **Create— Instance** to display the Add Instance form.

2. Click on the **Browse** button. This opens up a Library browser from which you can select components and the **Symbol** view .You will update the Library Name, Cell Name, and the property values given in the table on the next page as you place each component.

3. After you complete the Add Instance form, move your cursor to the schematic window and click **left** to place a component.

This is a table of

components for building the Differential Amplifier schematic.

After entering components, click **Cancel** in the Add Instance form or press **Esc** with your cursor in the schematic

Library name	**Cell Name**	**Properties/Comments**
gpdk180	nmos	Model Name = nmos1 (NM0, NM1) ; W= 3u ; L= 1u ; Body Type : Detached
gpdk180	nmos	Model Name =nmos1 (NM2, NM3) ; W= 4.5u ; L= 1u ; Body Type : Integrated
gpdk180	pmos	Model Name =pmos1 (PM0, PM2); W= 15u ; L= 1u ; Body Type : Integrated

window

Adding pins to Schematic

Use **Create – Pin** or the menu icon to place the pins on the schematic window.

1. Click the **Pin** fixed menu icon in the schematic window.You can also execute **Create – Pin** or press **p**. The Add pin form appears.

2. Type the following in the Add pin form in the exact order leaving space between the pin names.

Pin Names	**Direction**
Idc,V1,V2	Input

Vout	Output
vdd, vss,	InputOutput

Make sure that the direction field is set to **input/ouput/inputoutput** when placing the **input/output/inout** pins respectively and the Usage field is set to **schematic**.

3. Select **Cancel** from the Add pin form after placing the pins.In the schematic window, execute **View— Fit** or press the **f** bindkey.

Adding Wires to a Schematic

Add wires to connect components and pins in the design.

1. Click the **Wire (narrow)** icon in the schematic window.You can also press the **w** key, or execute **Create - Wire (narrow).**

2. Complete the wiring as shown in figure and when done wiring press ***ESC*** key in the schematic window to cancel wiring.

Saving the Design

1. Click the **Check and Save** icon in the schematic editor window.

2. Observe the **CIW** output area for any errors.

Symbol Creation

Objective: To create a symbol for the Differential Amplifier

1. In the Differential Amplifier schematic window, execute **Create — Cellview— From Cellview**.

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.**

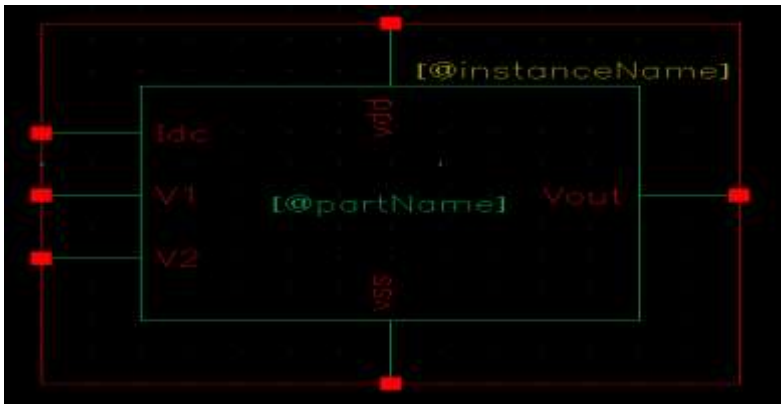
3. Click **OK** in the Cellview from Cellview form. The **Symbol Generation Form** appears.

4. Modify the **Pin Specifications** as in the below symbol.

5. Click **OK** in the Symbol Generation Options form.

6. A new window displays an automatically created Differential Amplifier symbol.

7. Modifying automatically generated symbol so that it looks like below Differential Amplifier symbol.

8. Execute **Create— Selection Box**. In the Add Selection Box form, click **Automatic.** A new red selection box is automatically added. 

9. After creating symbol, click on the **save** icon in the symbol editor window to save the symbol. In the symbol editor, execute **File— Close** to close the symbol view window.

Building the Diff_amplifier_test Design

Objective: To build Differential Amplifier Test circuit using your Differential Amplifier

Creating the Differential Amplifier Test Cellview

- In the CIW or Library Manager, execute **File— New— Cellview**.
- Set up the Create New File form as follows:

3. Click **OK** when done. A blank schematic window for the Diff_ amplifier_test design appears.

Building the Diff_amplifier_test Circuit

1. Using the component list and Properties/Comments in this table,build the Diff_amplifier_test schematic.

Library name	**Cellview name**	**Properties/Comments**
myDesignLib	Diff_amplifier	Symbol
analogLib	vsin	Define specification as AC Magnitude= 1; Amplitude= 5m; Frequency= 1K
analogLib	vdd, vss, gnd	Vdd=2.5 ; Vss= -2.5
analogLib	Idc	Dc current = 30u

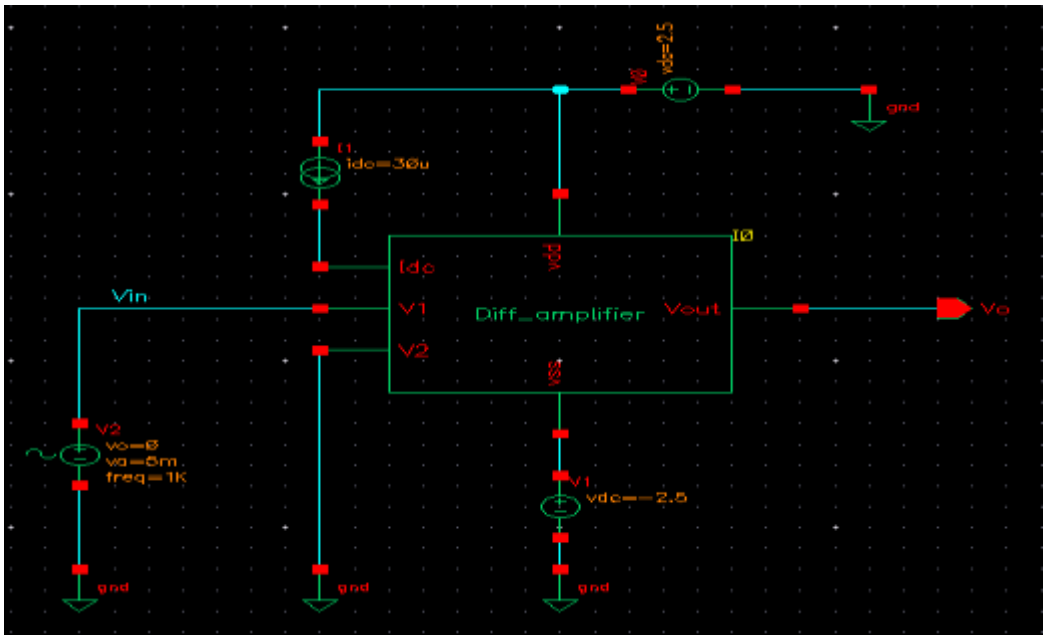
Note: Remember to set the values for **VDD** and **VSS**. Otherwise your circuit will have no power.

3. Click the **Wire (narrow)** icon and wire your schematic.

Tip: You can also press the **w** key, or execute **Create— Wire (narrow).**

4. Click on the **Check and save** icon to save the design.

5. The schematic should look like this.



6. Leave your Diff_amplifier_test schematic window open for the next section.

Analog Simulation with Spectre

Objective: To set up and run simulations on the Differential Amplifier Test design.

In this section, we will run the simulation for Differential Amplifier and plot the transient, DC and AC characteristics.

Starting the Simulation Environment

1. In the Diff_amplifier_test schematic window, execute **Launch – ADE L**. The Analog Design Environment simulation window appears.

Choosing a Simulator

1. In the simulation window or ADE, execute **Setup— Simulator/Directory/Host.**

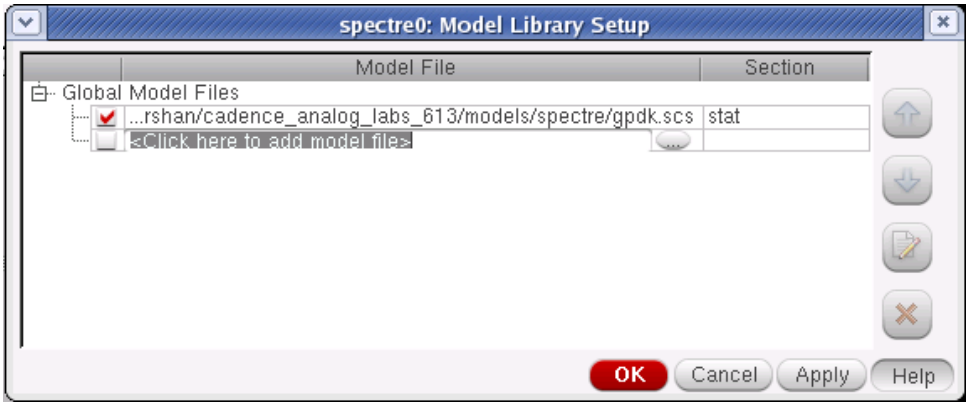
2. In the **Choosing Simulator** form, set the Simulator field to **spectre** (Not spectreS) and click **OK**.

Setting the Model Libraries

1. Click **Setup - Model Libraries.**

Note: Step 2 should be executed only if the model file not loaded by default.

2. In the Model Library Setup form, click **Browse** and find the **gpdk180.scs** file in the **./models/spectre** directory. Select **stat** in Section field, click **Add** and click **OK.**



Choosing Analyses

1. In the Simulation window, click the **Choose - Analyses** icon.You can also execute **Analyses - Choose.**

The Choosing Analysis form appears. This is a dynamic form, the bottom of the form changes based on the selection above.

2. To setup for transient analysis a. In the Analysis section select **tran**

    b. Set the stop time as **5m**

    c. Click at the **moderate** or **Enabled** button at the bottom, and then click **Apply.**

3. To set up for DC Analyses:

    a. In the Analyses section, select **dc**.

    b. In the DC Analyses section, turn **on** Save DC Operating Point.

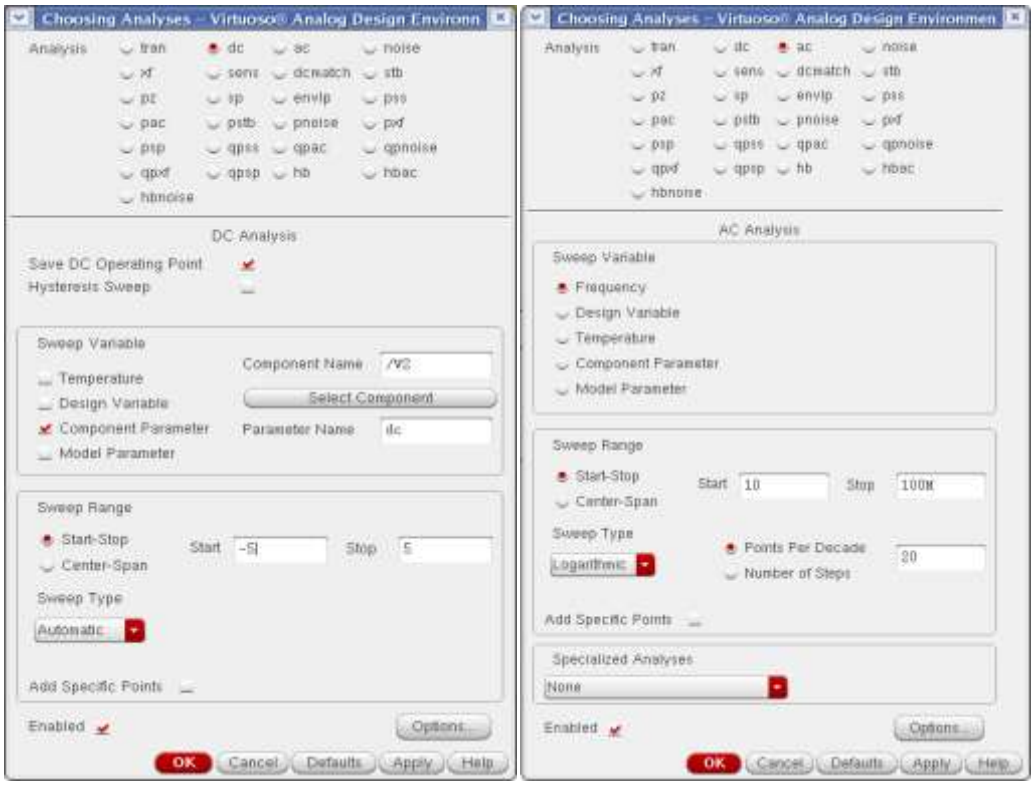
    c. Turn on the **Component Parameter**

    d. Double click the **Select Component**, Which takes you to the schematic window.

    e. Select input signal **Vsin** for dc analysis.

    f. In the analysis form, select **start** and **stop** voltages as **-5** to **5** respectively.

    g. Check the enable button and then click **Apply**.



4. To set up for AC Analyses form is shown in the previous page.

   a. In the Analyses section, select **ac**.

   b. In the AC Analyses section, turn on **Frequency.**

   c. In the Sweep Range section select **start** and **stop** frequencies as **150** to **100M**

   d. Select Points per Decade as **20**.

   e. Check the enable button and then click **Apply**.

5. Click **OK** in the Choosing Analyses Form.

Selecting Outputs for Plotting

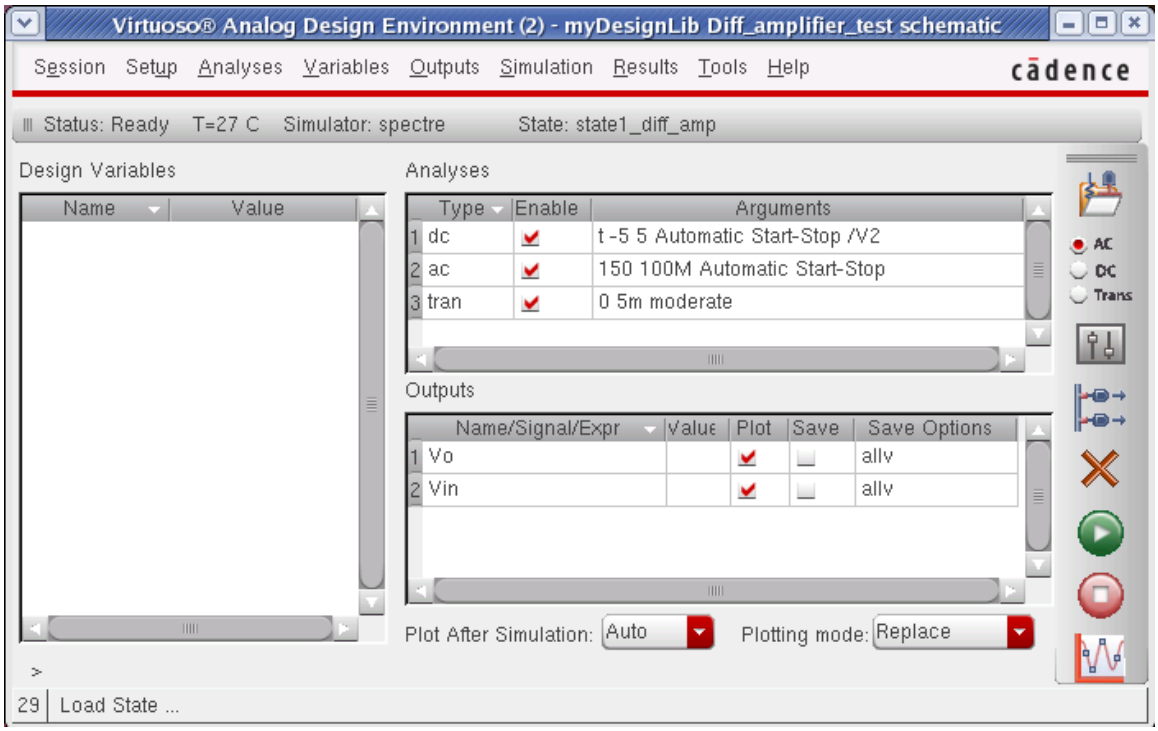
Select the nodes to plot when simulation is finished.

1. Execute **Outputs – To be plotted – Select on Schematic** in the simulation window.

2. Follow the prompt at the bottom of the schematic window, Click on output

net **Vo,** input net **Vin** of the Diff_amplifier. Press **ESC** with the cursor in the schematic after selecting node.

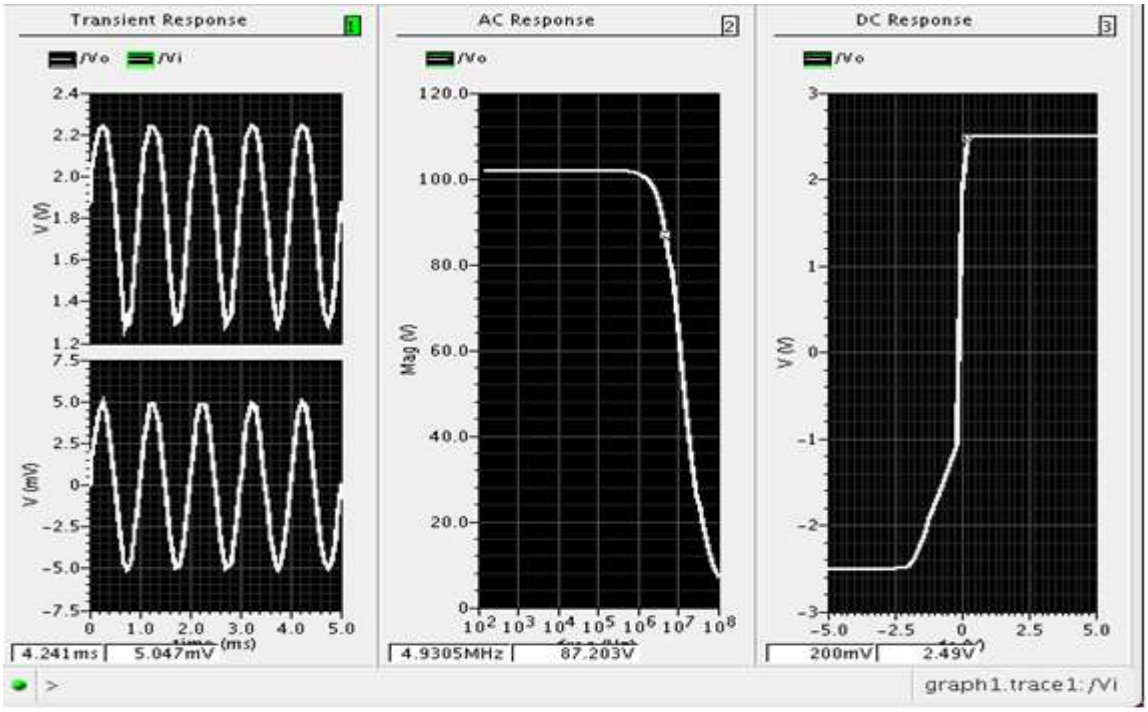
Does the simulation window look like this?



Running the Simulation

1. Execute **Simulation – Netlist and Run** in the simulation window to start the simulation, this will create the netlist as well as run the simulation.

2. When simulation finishes, the Transient, DC and AC plots automatically will be popped up along with netlist.



8. 6 RESULT: Designed and plotted the characteristics Differential amplifier

8.7 PRE LAB VIVA QUESTIONS:

1. Define differential amplifier?
2. What are the characterstics of differential amplifier?
3. What is the application of differential amplifier?

8.9 POST LAB VIVA QUESTIONS:

1. Design opamp by using differential amplifier?
2. What do you observe from characteristics of differential amplifier?

EXPERIMENT NO:9

9.1 AIM: To design layout of NMOS and CMOS inverter.

9.2 Creating Layout View of Inverter

- From the **Inverter** schematic window menu execute **Launch – Layout XL**. A **Startup Option** form appears.
- Select ***Create New*** option. This gives a New Cell View Form
- Check the Cellname **(Inverter)**, Viewname **(layout).**
- Click **OK** from the New Cellview form.
- LSW and a blank layout window appear along with schematic window.

Adding Components to Layout

- Execute **Connectivity – Generate – All from Source** or click the icon  in the layout editor window, **Generate Layout** form appears. Click **OK** which imports the schematic components in to the Layout window automatically.
- Re arrange the components with in PR-Boundary as shown in the next page.
- To rotate a component, Select the component and execute **Edit –Properties**. Now select the degree of rotation from the property edit form.

- To Move a component, Select the component and execute **Edit -Move** command.

Making interconnection

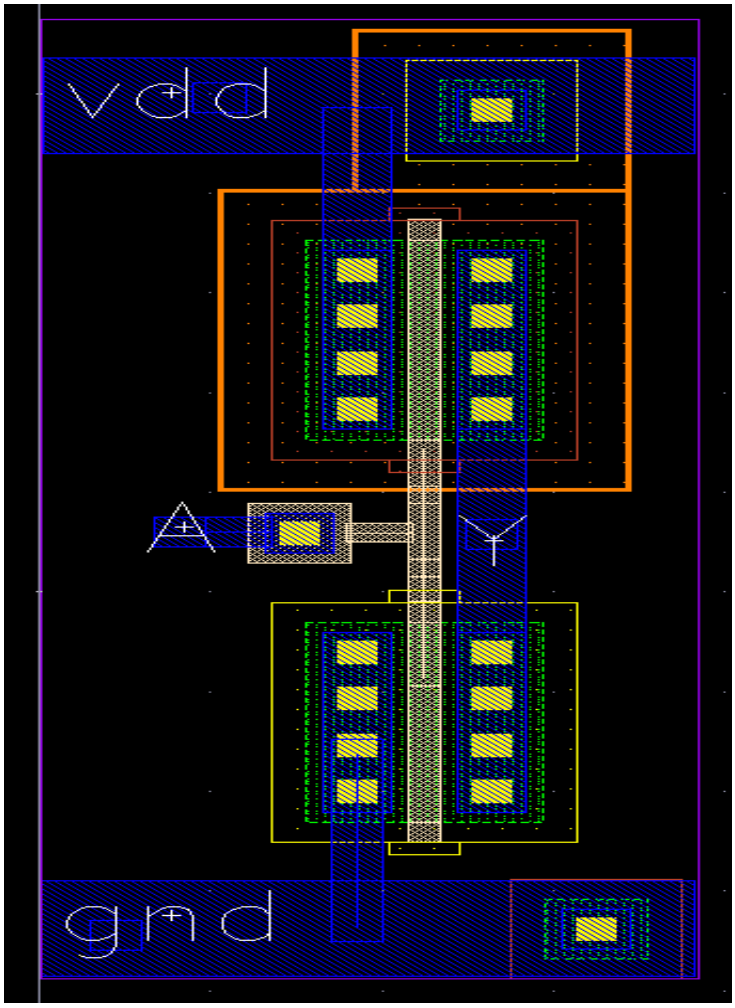
- Execute **Connectivity –Nets – Show/Hide selected Incomplete Nets** or click  the icon in the Layout Menu.
- Move the mouse pointer over the device and click **LMB** to get the connectivity information, which shows the guide lines (or flight lines) for the inter connections of the components.
- From the layout window execute **Create – Shape – Path/ Create wire** or **Create – Shape – Rectangle (**for vdd and gnd bar**)** and select the appropriate Layers from the **LSW** window and Vias for making the inter connections

Creating Contacts/Vias

You will use the contacts or vias to make connections between two different layers.

- 1. Execute **Create — Via** or select  command to place different Contacts, as given in below table

Connection	*Contact Type*
For Metal1- Poly Connection	Metal1-Poly
For Metal1- Psubstrate Connection	Metal1-Psub
For Metal1- Nwell Connection	Metal1-Nwell



Saving the design

1. Save your design next do

Physical Verification:

Assura DRC

Running a DRC

- Open the Inverter layout form the CIW or library manger if you have closed that.

Press **shift – f** in the layout window to display all the levels.

- Select **Assura - Run DRC** from layout window. The DRC form appears. The Library and Cellname are taken from the current design window, but rule file may be missing. Select the Technology as **gpdk180.** This automatically loads the rule file.

Your DRC form should appear like this

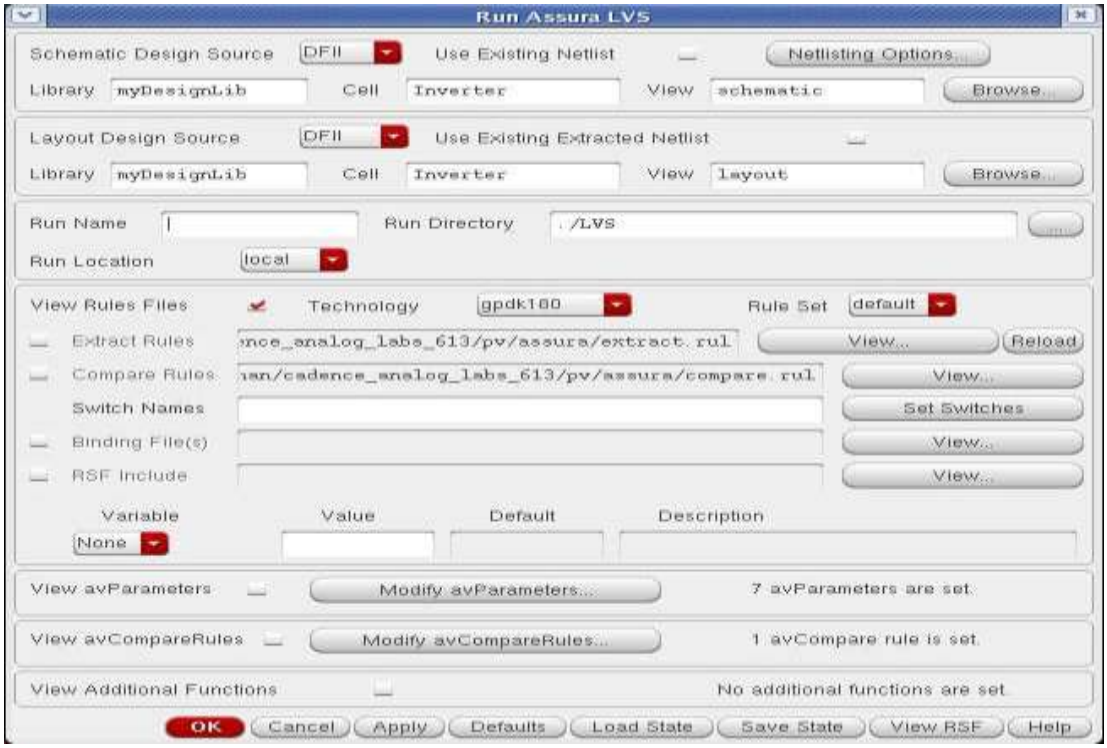
- Click **OK** to start DRC.
- A Progress form will appears. You can click on the watch log file to see the log file.
- When DRC finishes, a dialog box appears asking you if you want to view your DRC results, and then click **Yes** to view the results of this run.
- If there any DRC error exists in the design **View Layer Window** (VLW) and **Error Layer Window** (ELW) appears. Also the errors highlight in the design itself.
- Click **View – Summary** in the ELW to find the details of errors.
- You can refer to rule file also for more information, correct all the DRC errors and **Re – run** the DRC.
- If there are no errors in the layout then a dialog box appears with **No DRC errors found** written in it, click on **close** to terminate the DRC run.

ASSURA LVS

- In this section we will perform the LVS check that will compare the schematic netlist and the layout netlist.

Running LVS

- Select **Assura – Run LVS** from the layout window. The Assura Run LVS form appears. It will automatically load both the schematic and layout view of the cell.
- Change the following in the form and click **OK**.



- The LVS begins and a Progress form appears.
- If the schematic and layout matches completely, you will get the form displaying ***Schematic and Layout Match***.
  - If the schematic and layout do not matches, a form informs that the LVS completed successfully and asks if you want to see the results of this run.
  - Click **Yes** in the form LVS debug form appears, and you are directed into LVS debug environment.
  - In the **LVS debug form** you can find the details of mismatches and you need to correct all those mismatches and **Re – run** the LVS till you will be able to match the schematic with layout.

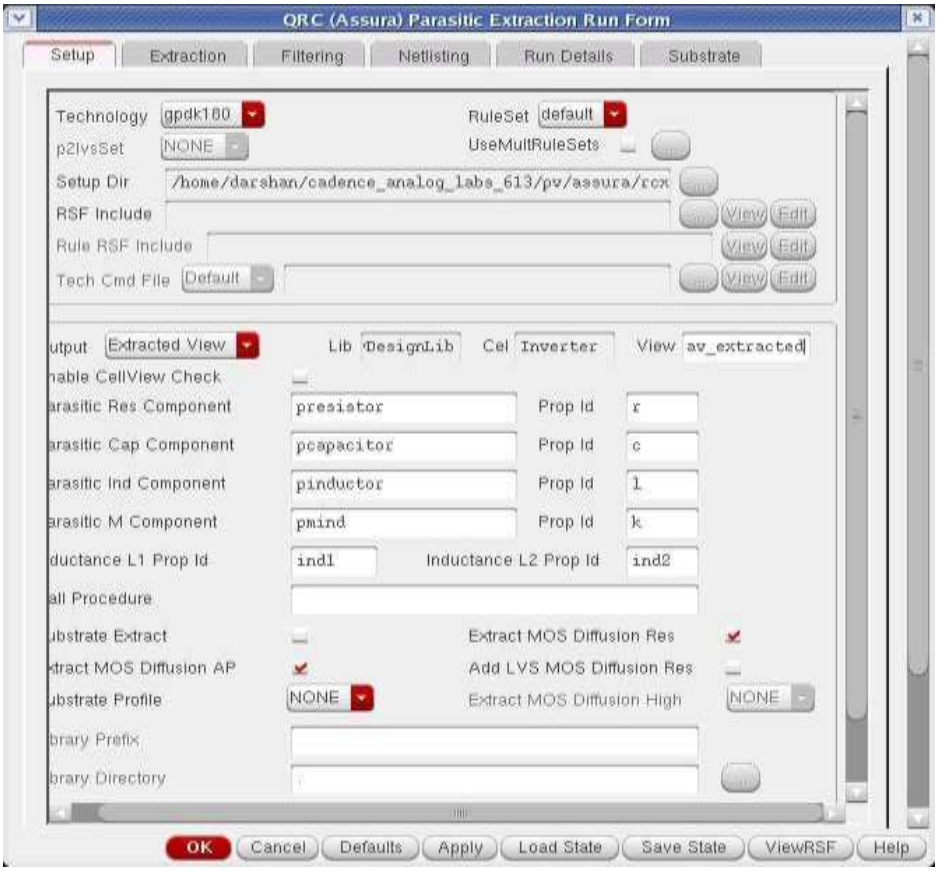
Assura RCX

In this section we will extract the RC values from the layout and perform analog circuit simulation on the designs extracted with RCX. Before using RCX to extract parasitic devices for simulation, the layout

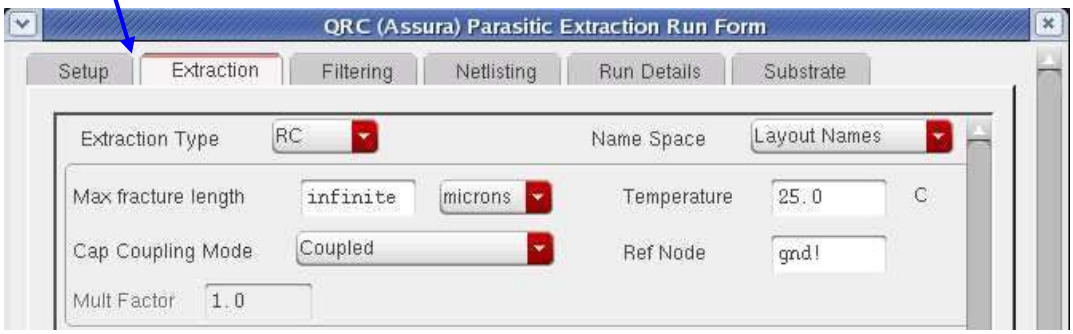
should match with schematic completely to ensure that all parasites will be backannoted to the correct schematic nets.

Running RCX

- From the layout window execute **Assura – Run RCX**.
- Change the following in the Assura parasitic extraction form. Select **output** type under **Setup** tab of the form.

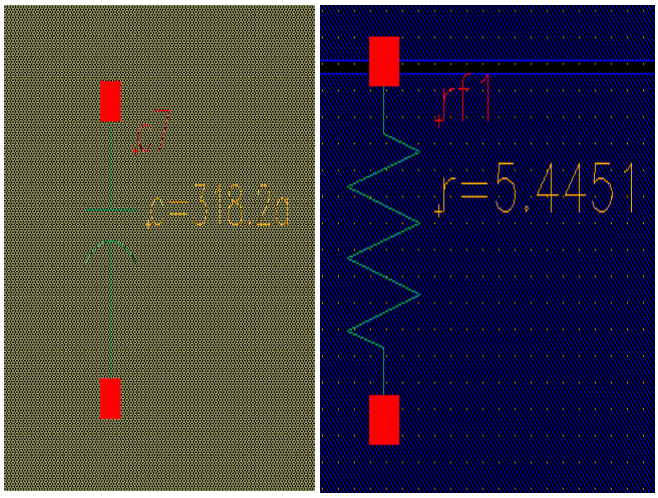


- In the **Extraction** tab of the form, choose Extraction type, Cap Coupling Mode and specify the Reference node for extraction.



- In the **Filtering** tab of the form, **Enter Power Nets** as **vdd!, vss!** and **Enter Ground Nets** as **gnd!**

- Click **OK** in the Assura parasitic extraction form when done. The RCX progress form appears, in the progress form click **Watch log file** to see the output log file.
- When RCX completes, a dialog box appears, informs you that **Assura RCX run Completed successfully.**
- You can open the **av_extracted** view from the library manager and view the parasitic.



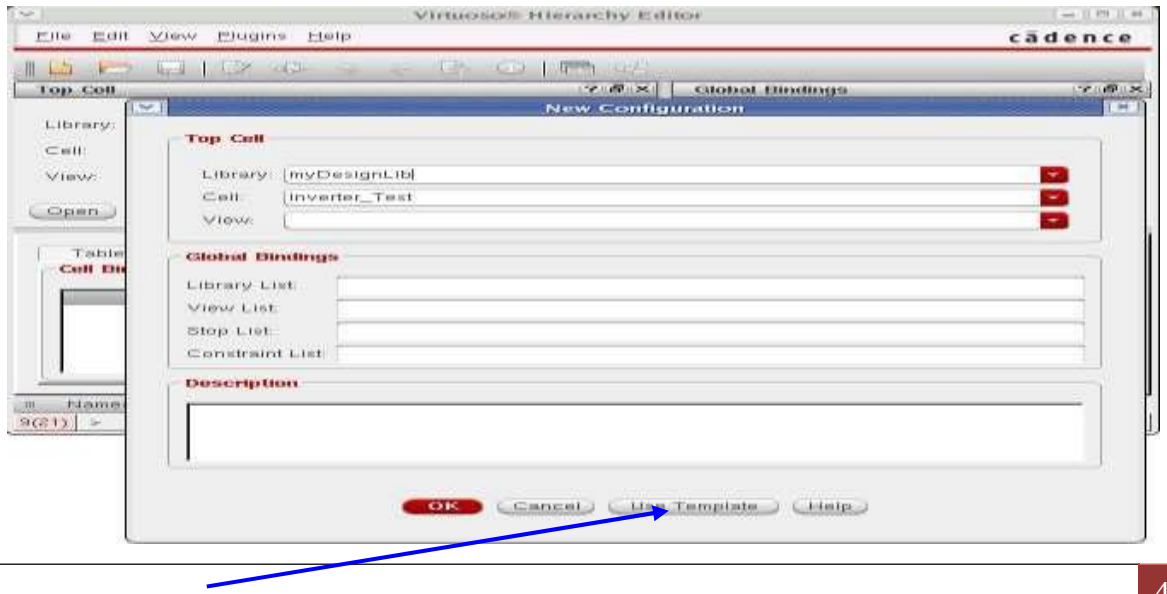
Creating the Configuration View

In this section we will create a config view and with this config view we will run the Simulation with and without parasitic.

- In the CIW or Library Manager, execute **File – New – Cellview**

- In the Create New file form, set the following:

- Click **OK** in create **New File** form. The **Hierarchy Editor** form opens and a **New Configuration** form opens in front of it.



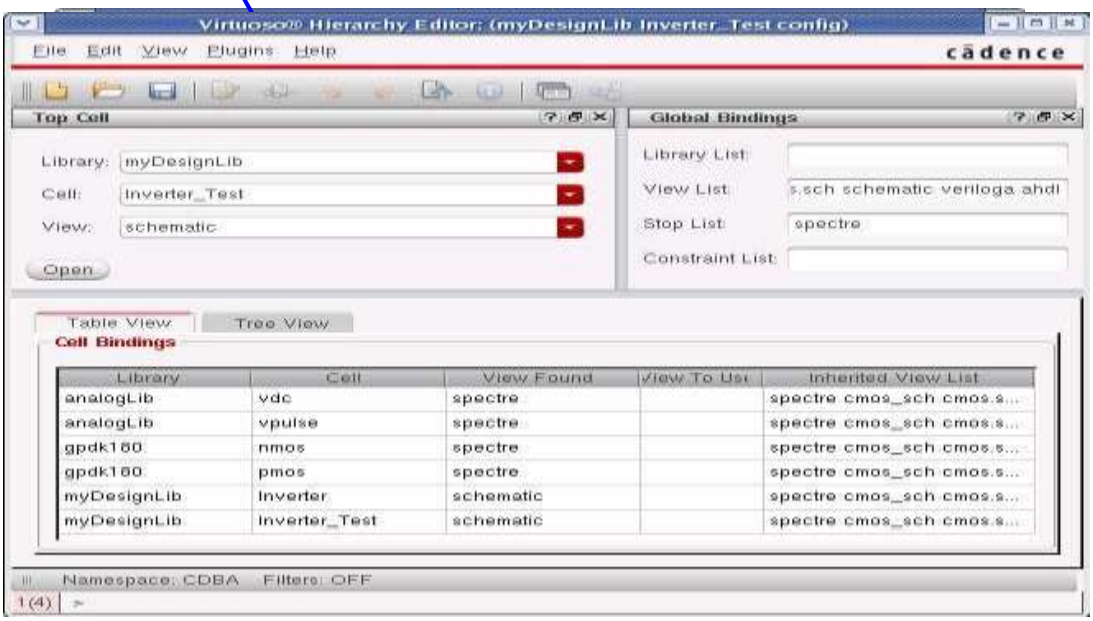
45

Click **Use template** at the bottom of the **New Configuration** form and select **Spectre** in the cyclic field and click **OK**. The Global Bindings lists are loaded from the template.

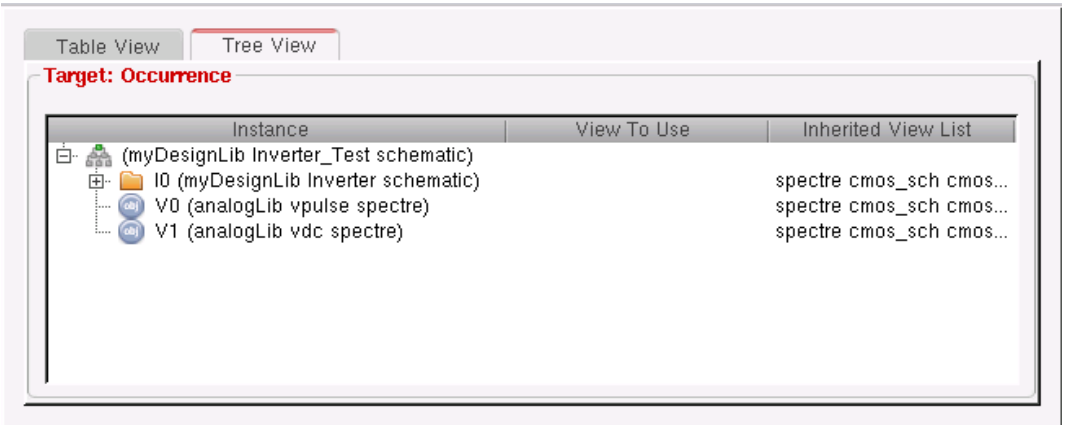
- Change the **Top Cell** View to **schematic** and remove the default entry from the **Library List** field.

- Click **OK** in the New Configuration form.

The hierarchy editor displays the hierarchy for this design using table format.



- Click the **Tree View** tab. The design hierarchy changes to tree format. The form should look like this:



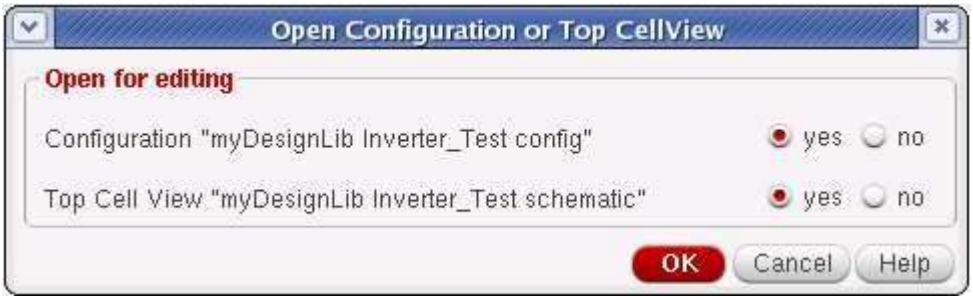
- ***Save*** the current configuration.

- Close the Hierarchy Editor window. Execute **File – Close Window**.

To run the Circuit without Parasites

- From the Library Manager open **Inverter_Test** Config view.

Open Configuration or Top cellview form appears.



- In the form, turn on the both cyclic buttons to **Yes** and click **OK.**

The Inverter_Test schematic and Inverter_Test config window appears. Notice the window banner of schematic also states **Config: myDesignLib Inverter_Test config.**

- Execute **Launch – ADE L** from the schematic window.
- Now you need to follow the same procedure for running the simulation. Executing

Session– Load state, the Analog Design Environment window loads the previous state.

- Click **Netlist and Run** icon to start the simulation.

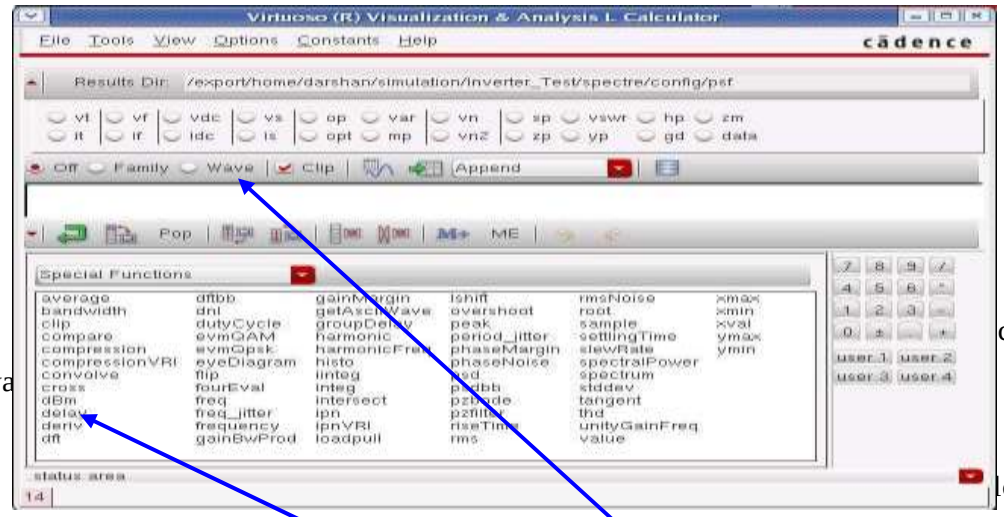
The simulation takes a few seconds and then waveform window appears.

- In the CIW, note the netlisting statistics in the **Circuit inventory** section. This list includes all nets, designed devices, source and loads. There are no parasitic components. Also note down the circuit inventory section.

Measuring the Propagation Delay

1. In the waveform window execute **Tools – Calculator.**

The calculator window appears.



- ct the input wa
-
- lculator to calculate delay at 50% i.e. at 0.9 volts.
- Execute **OK** and observe the expression created in the calculator buffer.
- Click on **Evaluate the buffer icon** to perform the calculation, note down the value returned after execution.

- Close the calculator window.

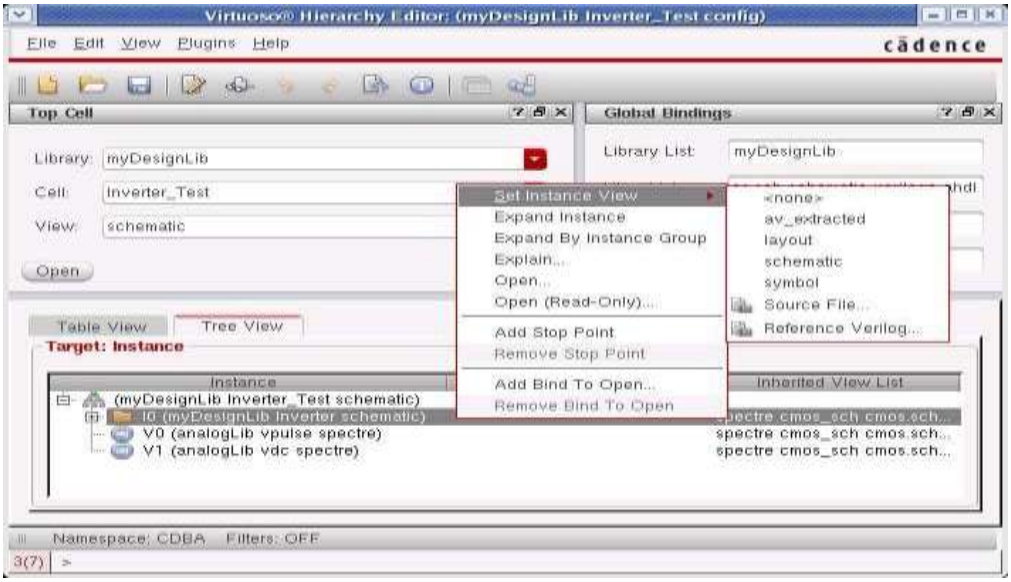
To run the Circuit with Parasites

In this exercise, we will change the configuration to direct simulation of the **av_extracted** view which contains the parasites.

- Open the same Hierarchy Editor form, which is already set for Inverter_Test config.
- Select the **Tree View** icon: this will show the design hierarchy in the tree format.
- Click **right** mouse on the Inverter schematic.

A pull down menu appears. Select **av_extracted** view from the **Set Instance view**

menu, the View to use column now shows av_extracted view.



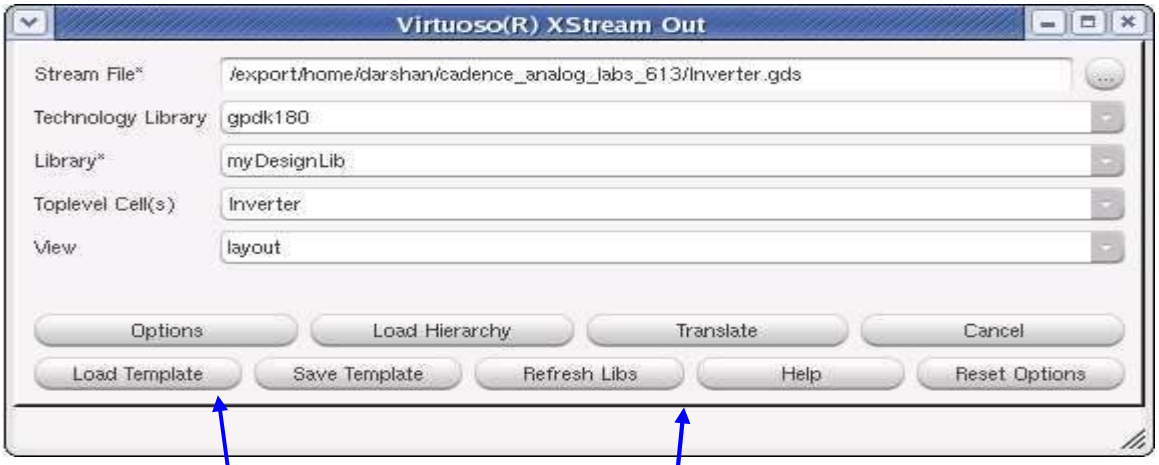
- Click on the **Recompute the hierarchy** icon,  the configuration is now updated from schematic to av_extracted view.

- From the **Analog Design Environment** window click **Netlist and Run** to start the simulation again.
- When simulation completes, note the **Circuit inventory conditions**, this time the list shows all nets, designed devices, sources and parasitic devices as well.
- Calculate the delay again and match with the previous one. Now you can conclude how much delay is introduced by these parasites, now our main aim should to minimize the delay due to these parasites so number of iteration takes place for making an optimize layout.

Generating Stream Data

Streaming Out the Design

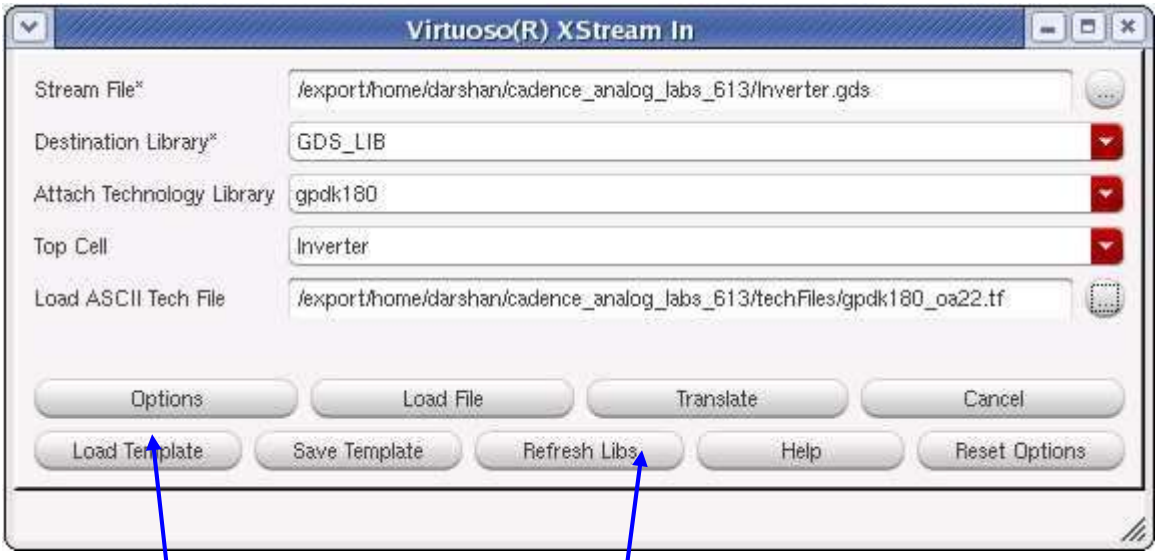
- Select **File – Export – Stream** from the CIW menu and **Virtuoso Xstream out** form appears change the following in the form.



- Click on the **Options** button.
- In the **StreamOut-Options** form select Use Automatic Mapping under **Layers** tab and click **OK.**
- In the **Virtuoso XStream Out** form, click **Translate** button to start the stream translator.
- The stream file Inverter.gds is stored in the specified location.

Streaming In the Design

- Select **File – Import – Stream** from the CIW menu and change the following in the form.



You need to specify the **gpdk180_oa22.tf** file. This is the entire technology file that has been dumped from the design library.

- Click on the **Options** button.
- In the **StreamOut-Options** form select Use Automatic Mapping under **Layers** tab and click **OK.**
- In the **Virtuoso XStream Out** form, click **Translate** button to start the stream translator.
- From the Library Manager open the **Inverter** cellview from the **GDS_LIB** library and notice the design.

- Close all the windows except CIW window, which is needed for the next lab.

9.5 RESULT: Designed layout of inverter and verifications has been done on the layout.

9.6 PRE LAB VIVA QUESTIONS:

1. What do you observe from inverter previous experiment?

2. What are the calculation did using simulation of inverter?

9.7 POST LAB VIVA QUESTIONS:

1. What is DRC?

2 What is LVS?

EXPERIMENT-10

10.1 AIM: To Design the layout of 2-input NAND, NOR gates.

10.2 LEARNING OBJECTIVE: To Design the layout of 2-input NAND, NOR gates.

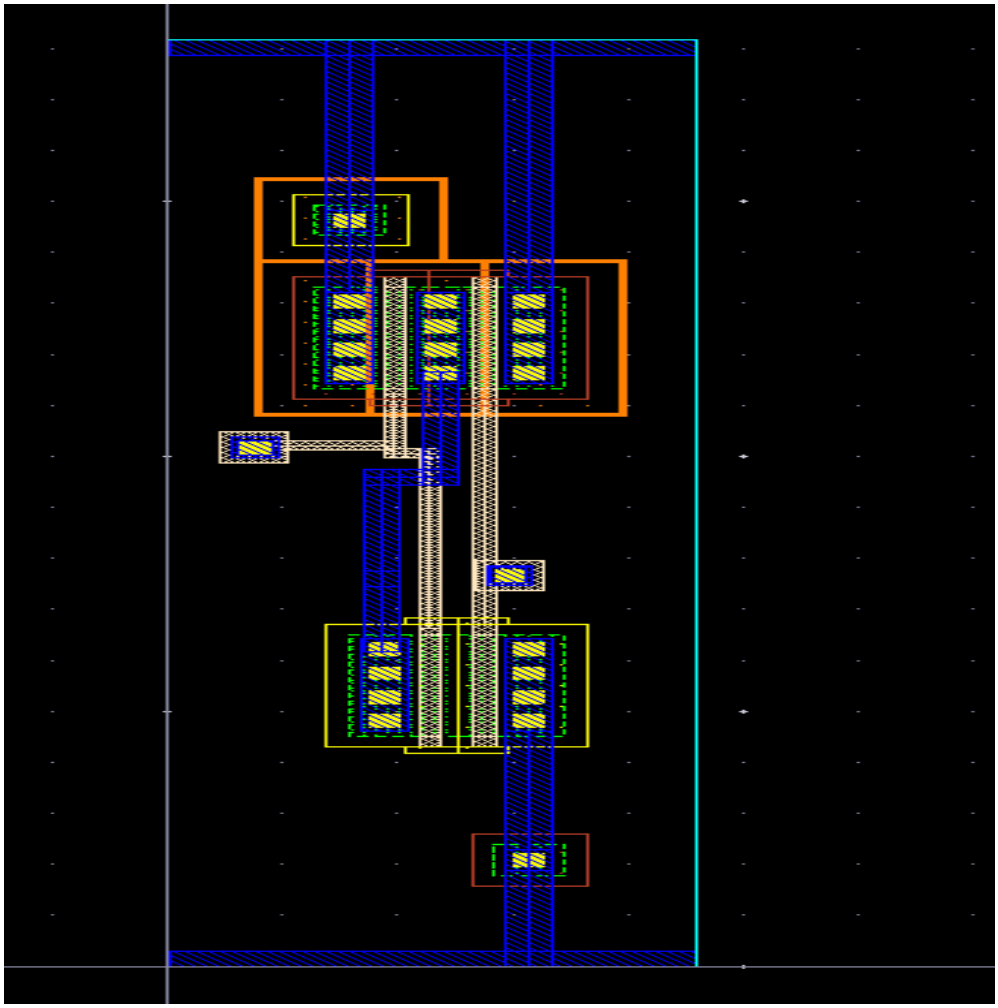
10.3 TOOLS REQUIRED: PC

CADENCE TOOLS

10.4 PROCEDURE:

Use the techniques learned in the Lab2.1 to complete the layout of NAND gate. Complete the DRC, LVS check using the assura tool.

Extract RC parasites for back annotation and Re-simulation.



10.4 RESULT: Designed layout of nand gate and verifications has been done on the layout.

10.5 PRE LAB VIVA QUESTIONS:

1. What is the application of NAND gate?

2. What is difference between nand and nor function?

3. Why NAND and NOR are universal gates?

10.6 POST LAB VIVA QUESTIONS:

1. Define LVS?

2. What do you observe from characteristics of NAND gate?

EXPERIMENT-11

11.1 AIM: To draw the schematic, perform simulation for common source amplifier.

11.2 TOOLS REQUIRED:

1.PC

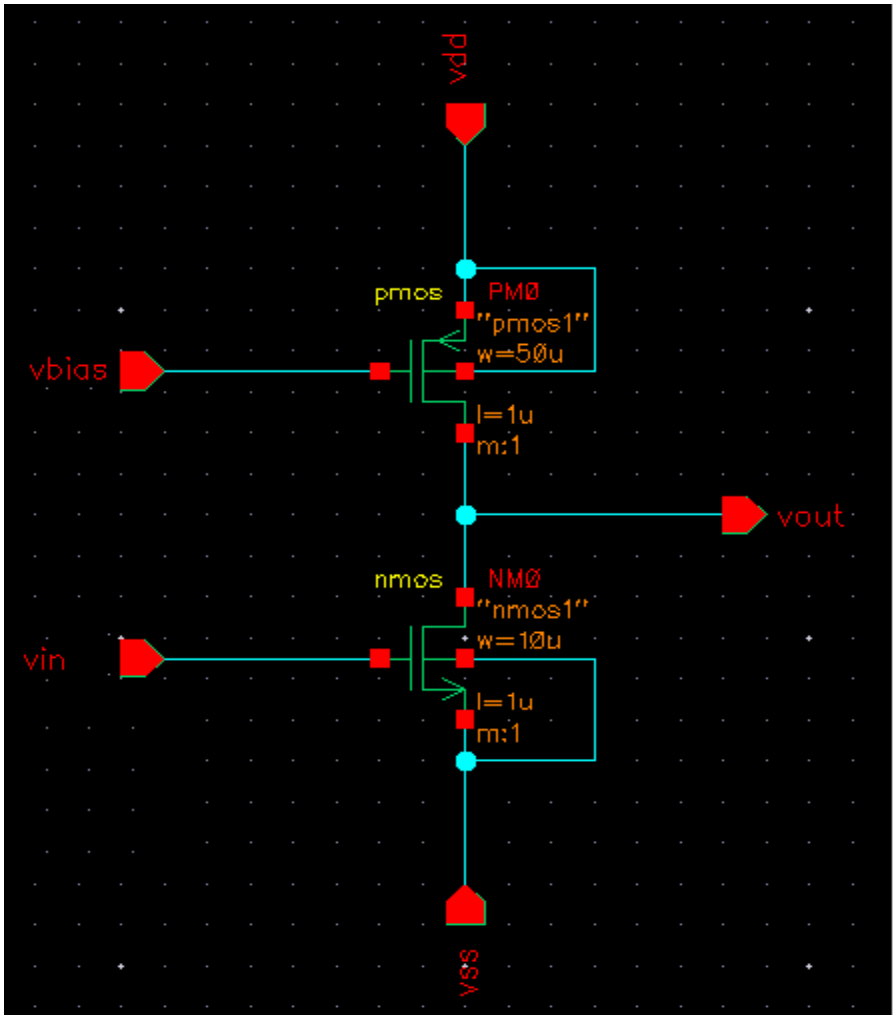
2. CADENCE IC 6.1.4

11.3 PROCEDURE:

Schematic Entry

Objective: To create a new cell view and build Common Source Amplifier

---



Schematic Capture

Use the techniques learned in the inverter to complete the schematic of Common Source Amplifier.

This is a table of components for building the Common Source Amplifier schematic.

Library name	**Cell Name**	**Properties/Comments**
gpdk180	Pmos	Model Name = pmos1; W= 50u ; L= 1u; Body Type : Integrated
gpdk180	Nmos	Model Name = nmos1; W= 10u ; L= 1u; Body Type : Integrated

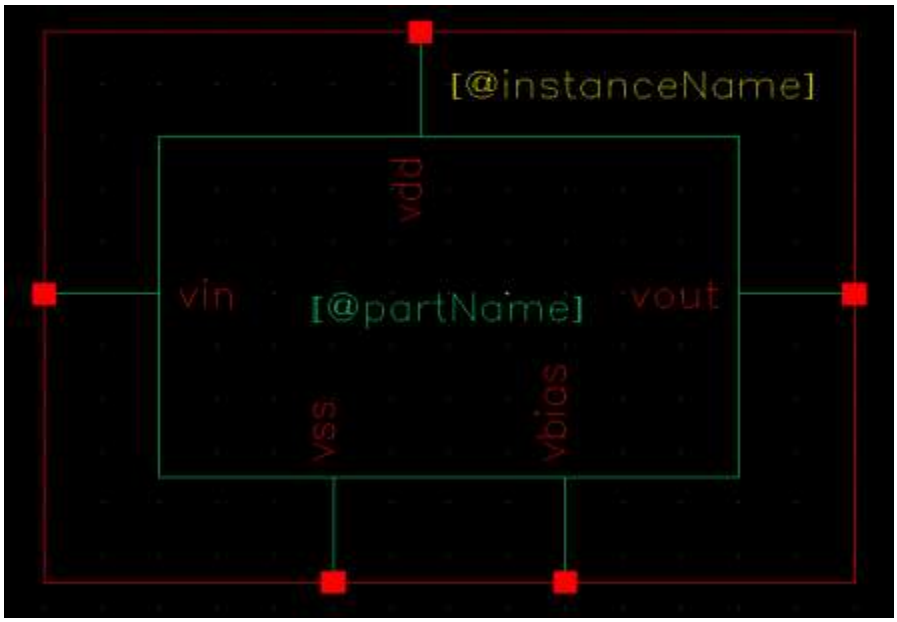
Type the following in the ADD pin form in the exact order leaving space between the pin names.

Pin Names	**Direction**
vin vbias	Input
vout	Output
vdd vss	Input

Symbol Creation

Objective: To create a symbol for the Common Source Amplifier

Use the techniques learned in the inverter to complete the symbol of cs-amplifier



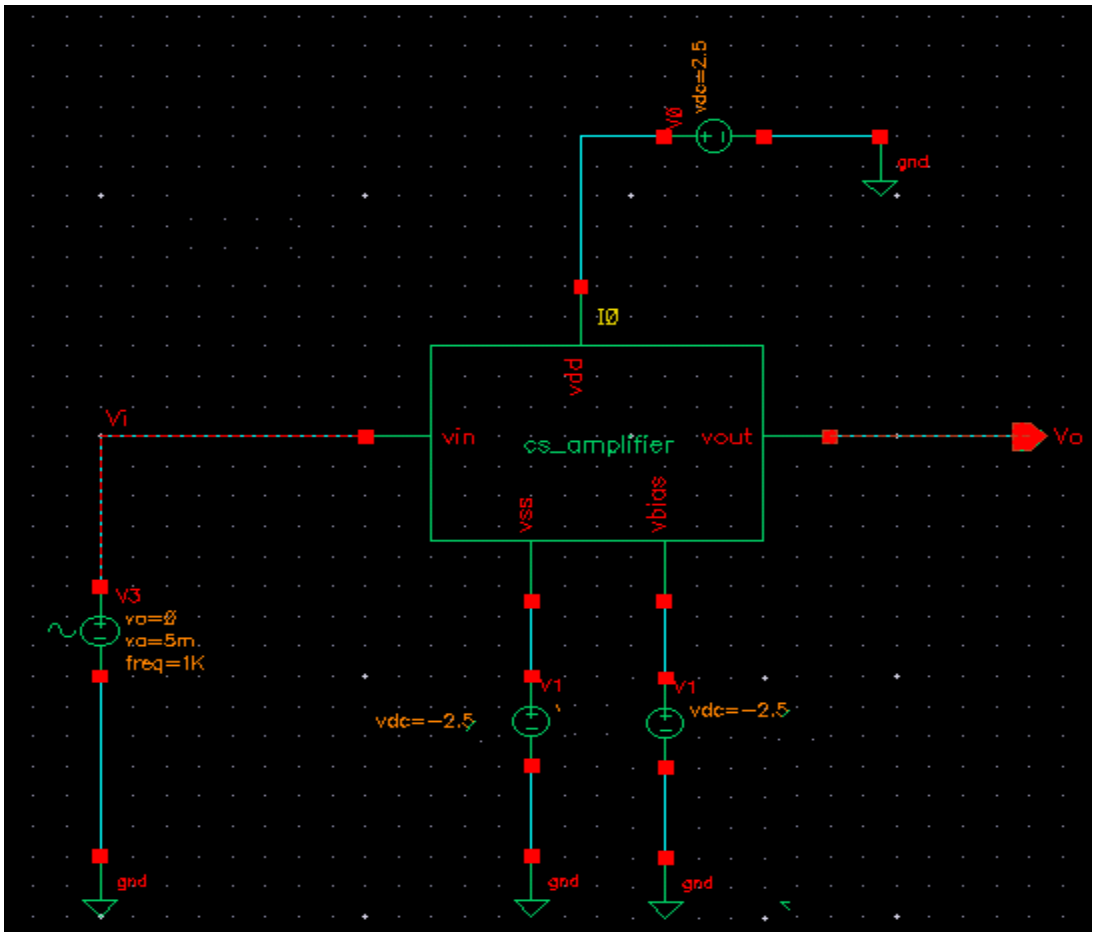
Building the Common Source Amplifier Test Design

Objective: To build cs_amplifier_test circuit using your cs_amplifier

Using the component list and Properties/Comments in the table,

build the cs-amplifier_test schematic as shown below.

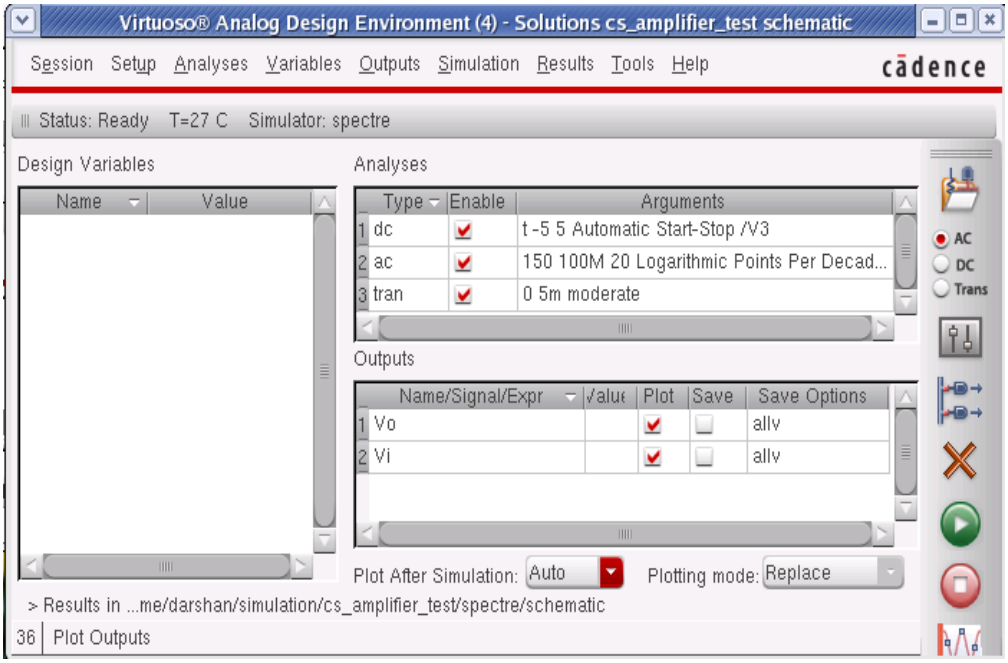
Library name	**Cellview name**	**Properties/Comments**
myDesignLib	cs_amplifier	Symbol
analogLib	vsin	Define pulse specification as AC Magnitude= 1; DC Voltage= 0; Offset Voltage= 0; Amplitude= 5m; Frequency= 1K
analogLib	vdd,vss,gnd	vdd=2.5 ; vss= -2.5 vbias=-2.5

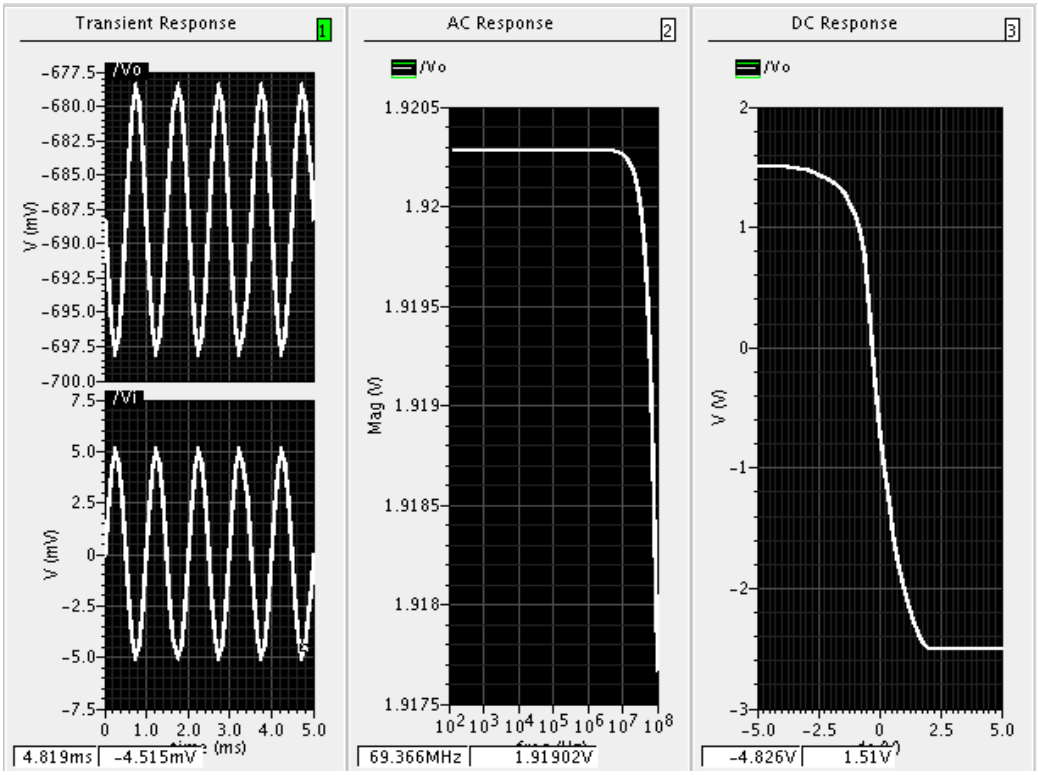


Analog Simulation with Spectre

Objective: To set up and run simulations on the cs_amplifier_test design.

- Use the techniques learned in the inverter to complete the simulation of cs_amplifier, ADE window and waveform should look like below.





11.4 RESULT: Analyzed of frequency response of common source amplifier.

11.5 PRE LAB VIVA QUESTIONS:

1. Define common source amplifier?
2. What are the characterstics of common source amplifier?
3. What is the application of common source amplifier?

11.6 POST LAB VIVA QUESTIONS:

1. Design opamp by using differential amplifier?
2. What do you observe from characteristics of common source amplifier?

EXPERIMENT-12

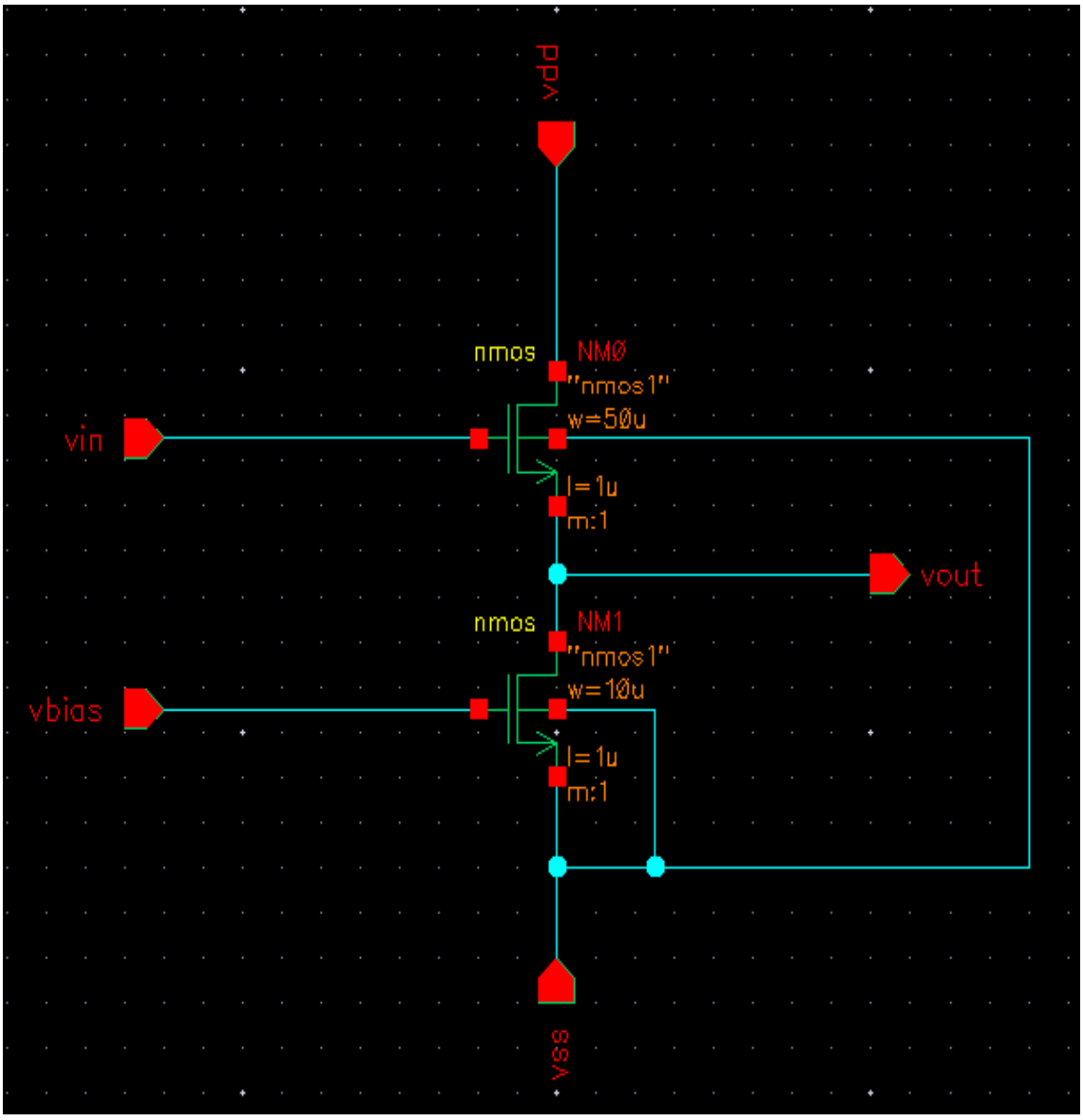
12.1 AIM: To draw the schematic, perform simulation for Common Drain amplifier.

12.2 TOOLS REQUIRED:

1. PC 2. CADENCE IC 6.1.4

12.3 PROCEDURE:

Schematic Entry



Schematic Entry

Objective: To create a new cell view and build Common Drain Amplifier

- Use the techniques learned in the Lab1 and Lab2 to complete the schematic of Common Drain Amplifier.

This is a table of components for building the Common Drain Amplifier schematic.

Library name	**Cell Name**	**Properties/Comments**
gpdk180	nmos	Model Name = nmos1; W= 50u ; L= 1u
gpdk180	nmos	Model Name = nmos1; W= 10u ; L= 1u

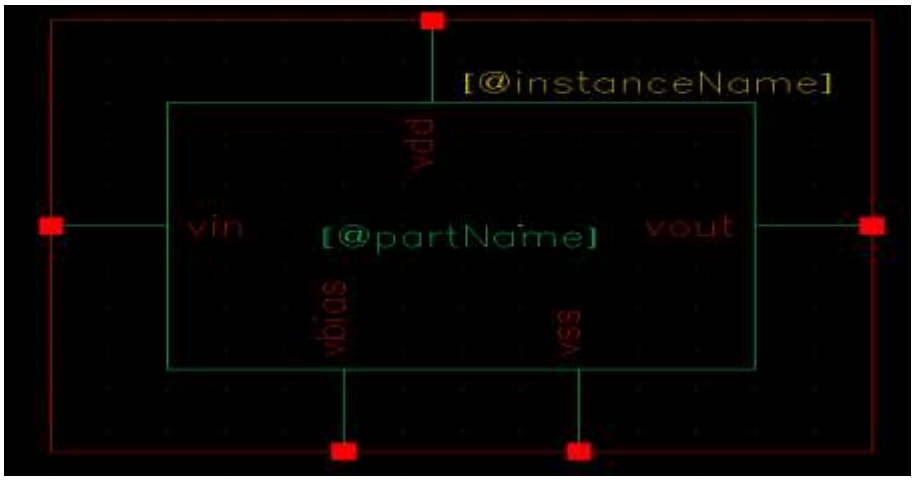
- Type the following in the ADD pin form in the exact order leaving space between the pin names.

Pin Names	**Direction**
vin, vbias	Input
vout	Output
vdd vss	Input

Symbol Creation

Objective: To create a symbol for the Common Drain Amplifier

Use the techniques learned in the Lab1 and Lab2 to complete the symbol of cd-amplifier



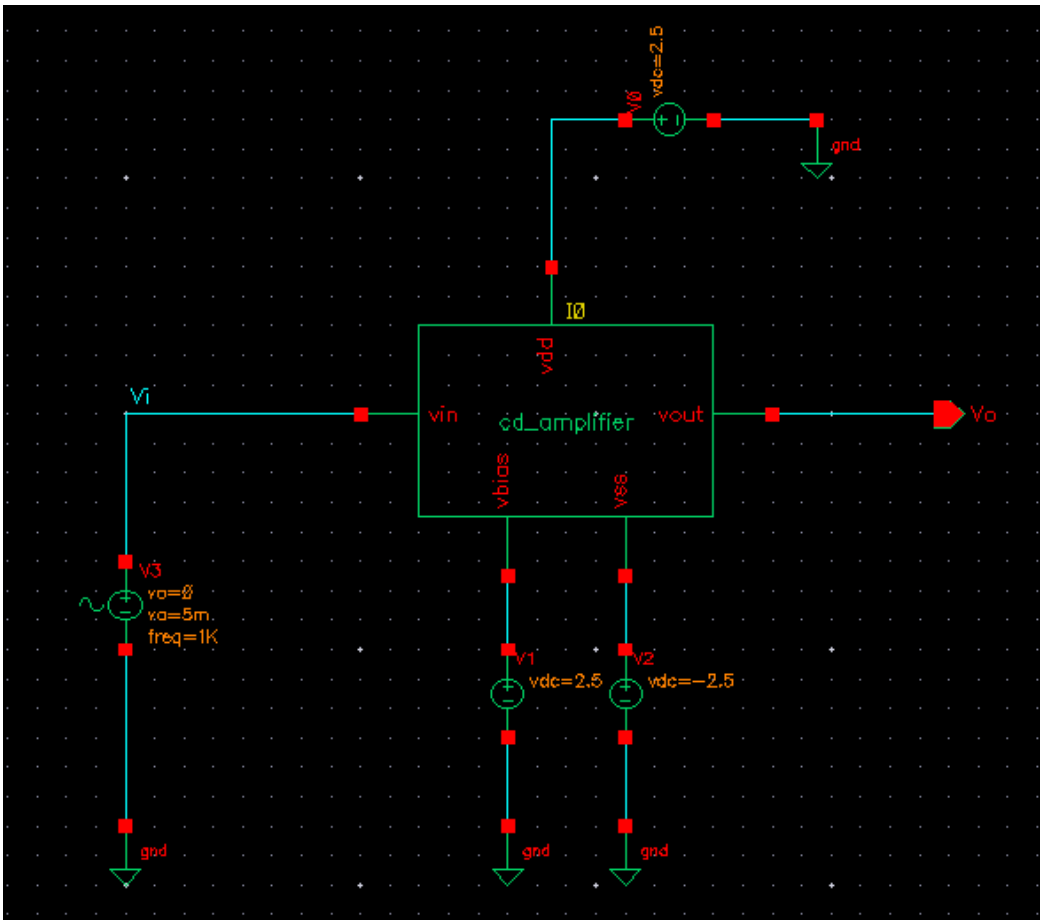
Building the Common Drain Amplifier Test Design

Objective: To build cd_amplifier_test circuit using your cd_amplifier

Using the component list and Properties/Comments in the table,

build the cd-amplifier_test schematic as shown below.

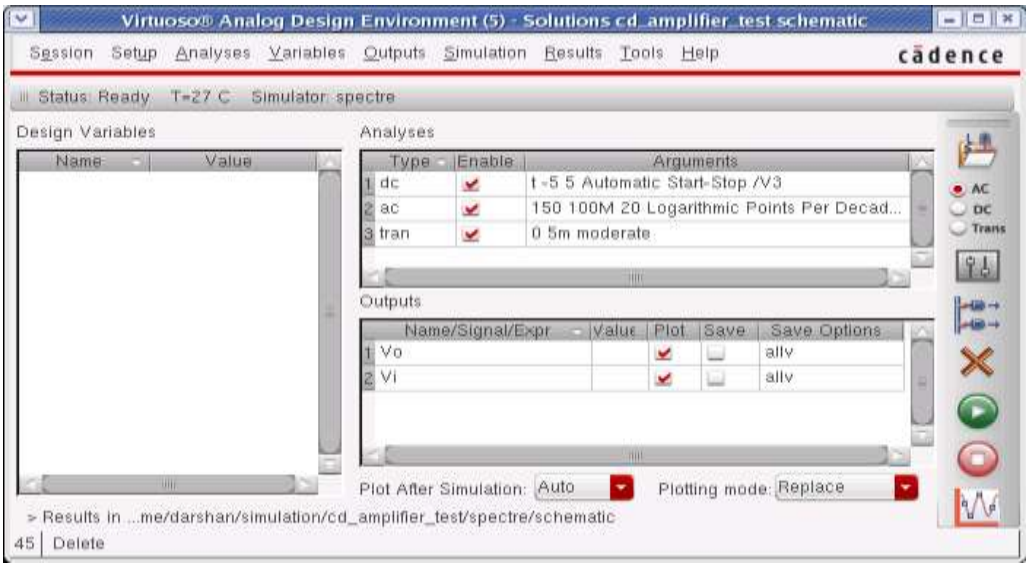
Library name	Cellview name	Properties/Comments
myDesignLib	cd_amplifier	Symbol
analogLib	vsin	Define pulse specification as AC Magnitude= 1; DC Voltage= 0; Offset Voltage= 0; Amplitude= 5m; Frequency= 1K
analogLib	vdd,vss,gnd	vdd=2.5 ; vss= -2.5

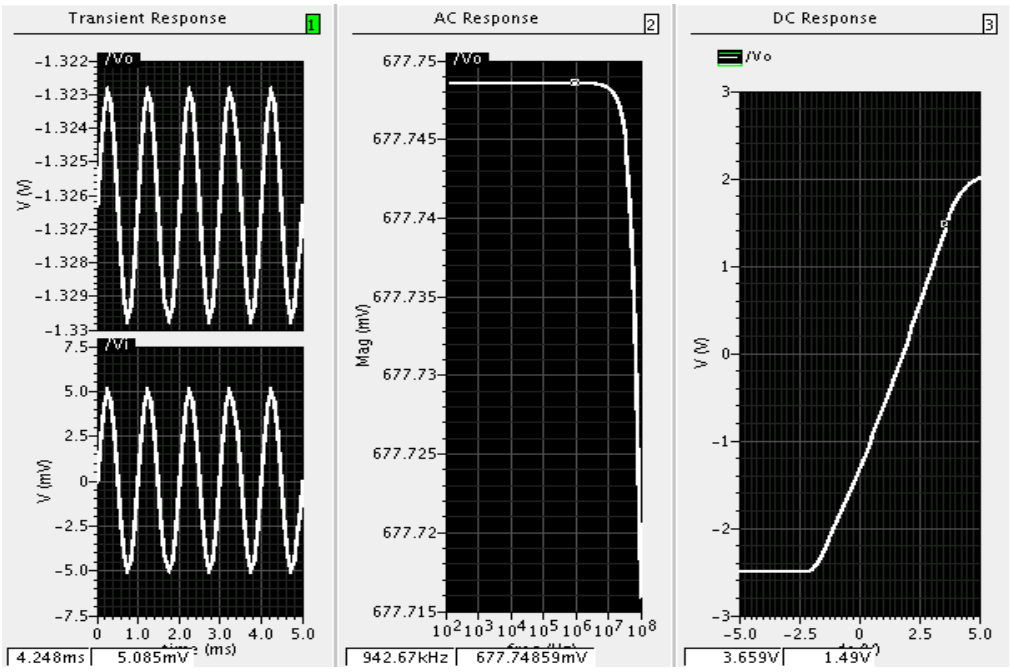


Analog Simulation with Spectre

Objective: To set up and run simulations on the cd_amplifier_test design.

- Use the techniques learned in the Lab1 and Lab2 to complete the simulation of cd_amplifier, ADE window and waveform should look like below.





12.4 Result: **Observed the simulation result of the experiment .**

12.5 PRE LAB VIVA QUESTIONS:

1. Define Common drain amplifier?
2. What are the characterstics of Common drain amplifier?
3. What is the difference between common source and common drain amplifier?

12.6 POST LAB VIVA QUESTIONS:

1. What are the advantages of common drain amplifier?
2. What do you observe from characteristics of common drain amplifier?

EXPERIMENT-13

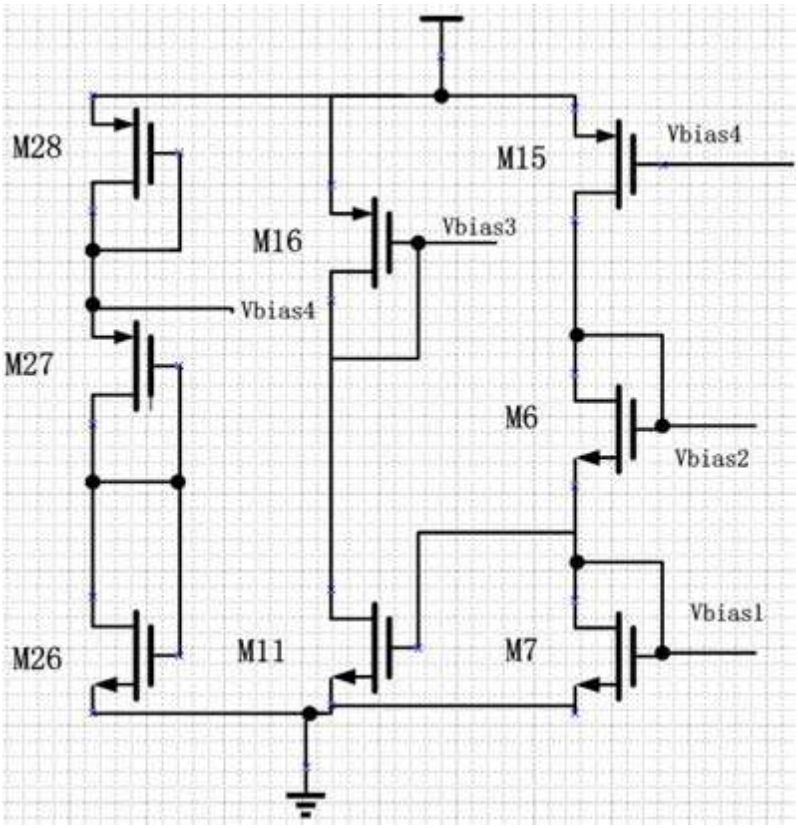
13.1 AIM: To draw the schematic, perform simulation for Cascode amplifier.

13.2 TOOLS REQUIRED:1. PC 2. CADENCE IC 6.1.4

13.3 PROCEDURE: Follow the procedure what you have followed till now for the design ,do the simulation find gain margin and phase margin .

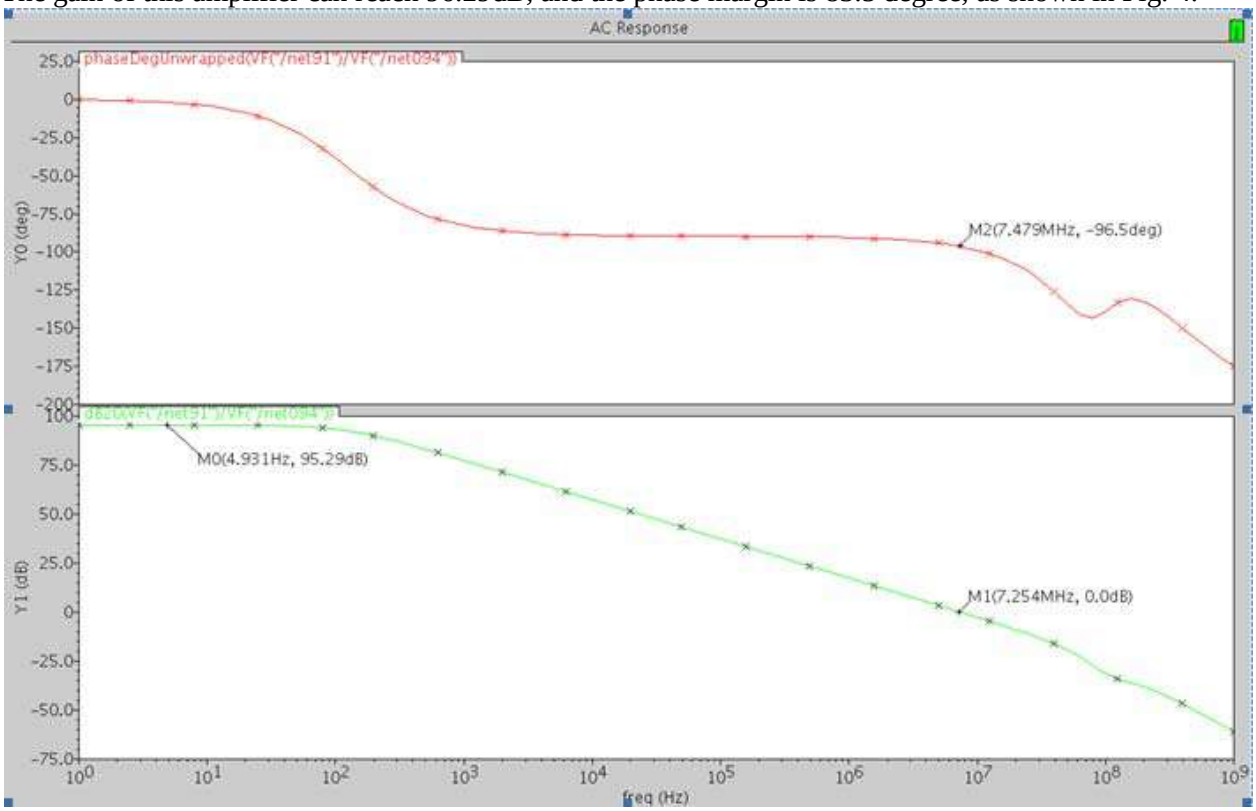
Schematic Entry

This amplifier is designed with Cadence simulation based on the technique library of SMIC 0.18um.



Stimulation result

The gain of this amplifier can reach 96.29dB, and the phase margin is 83.5 degree, as shown in Fig. 4.



13.4 RESULT: Study the characterstics of cascade amplifier .

13.5 PRE LAB VIVA QUESTIONS:

1. Define gain margin?
2. Define phase margin?
3. What is the difference between cascade and cascode?

13.6 POST LAB VIVA QUESTIONS:

1. What are the advantages of cascade amplifier?
2. What do you observe from characteristics of common drain amplifier?

EXPERIMENT-14

14.1 AIM: To draw the schematic, perform simulation, of current mirror.

14.2 LEARNING OBJECTIVE: To Design the layout of 2-input NAND, NOR gates.

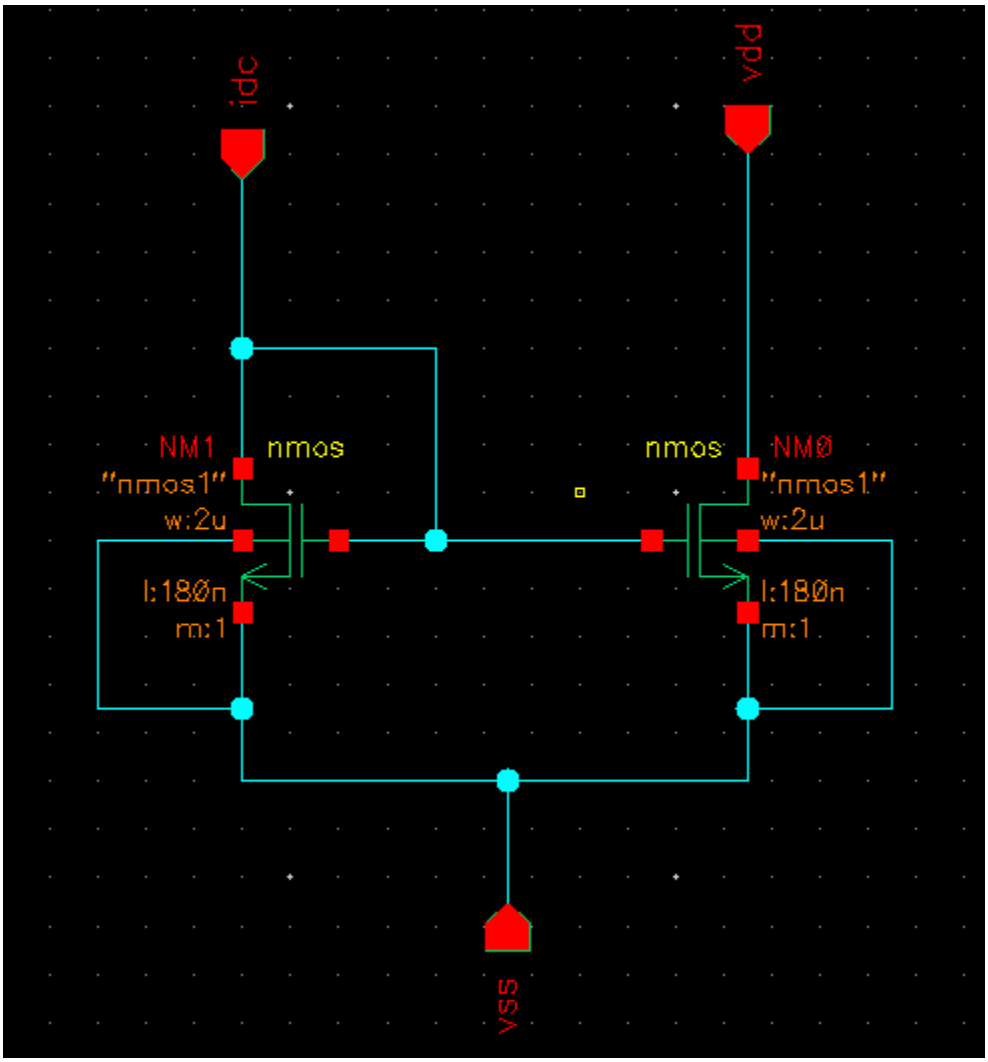
14.3 TOOLS REQUIRED: PC

CADENCE TOOLS

14.4 PROCEDURE:

Schematic Entry

Objective: To create a new cell view and build current mirror



Schematic Capture

- Use the techniques learned in the inverter to complete the schematic of current mirror.
- This is a table of components for building the Common Source Amplifier schematic.

Library name	Cell Name	Properties/Comments
gpdk180	Nmos	Model Name = nmos1; W=2u ; L= 180n; Body Type : Integrated

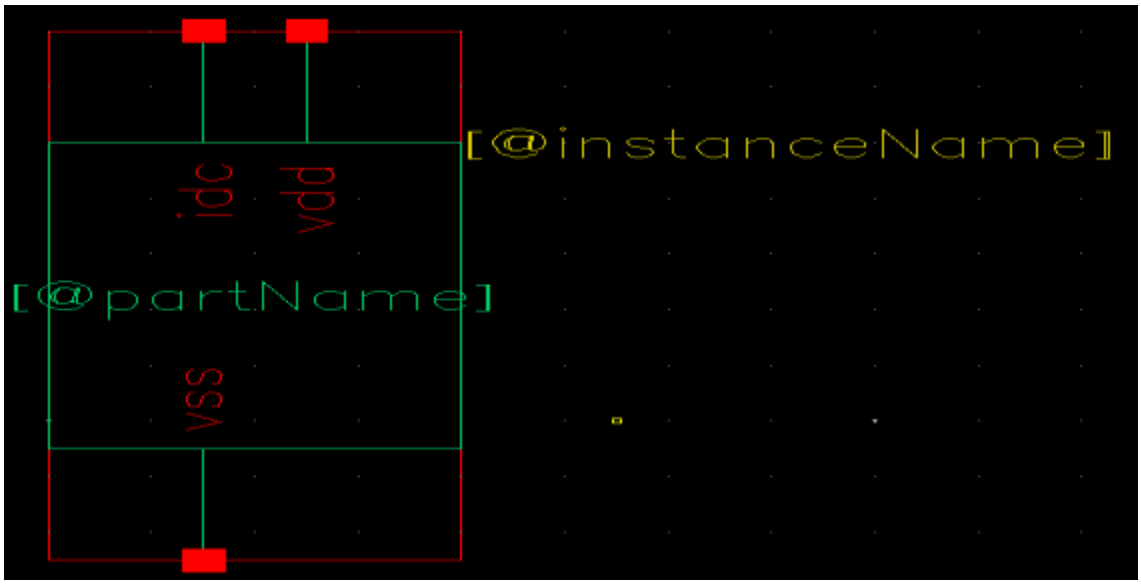
- Type the following in the ADD pin form in the exact order leaving space between the pin names.

Pin Names	Direction
idc	Input
vdd vss	Input

Symbol Creation

Objective: To create a symbol for the current mirror

- Use the techniques learned in the inverter to complete the symbol of currmirror

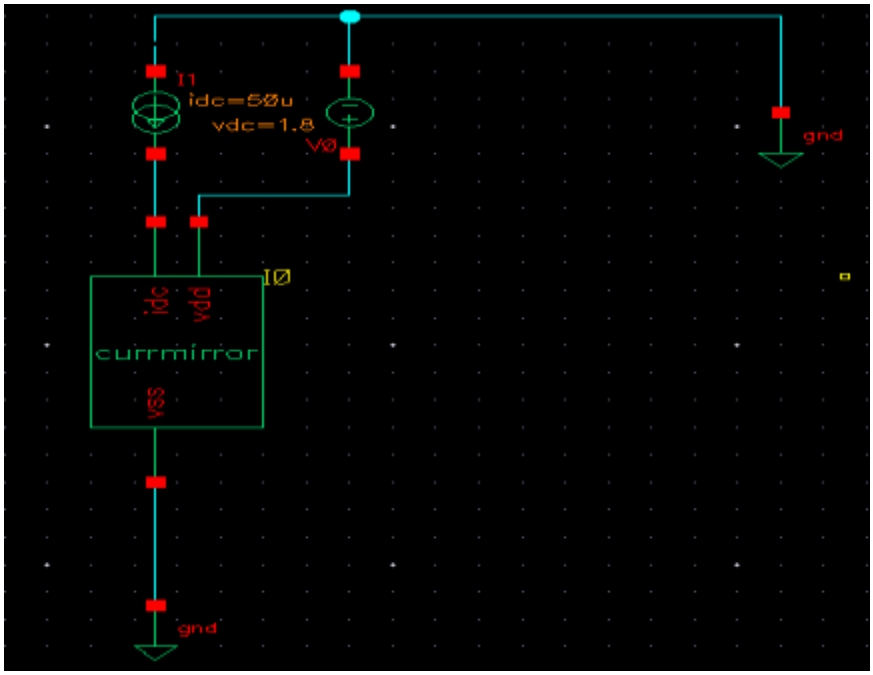


Building the current mirror Test Design

Objective: To build _test circuit using your currmirror

- Using the component list and Properties/Comments in the table,build the currmirror_test schematic as shown below.

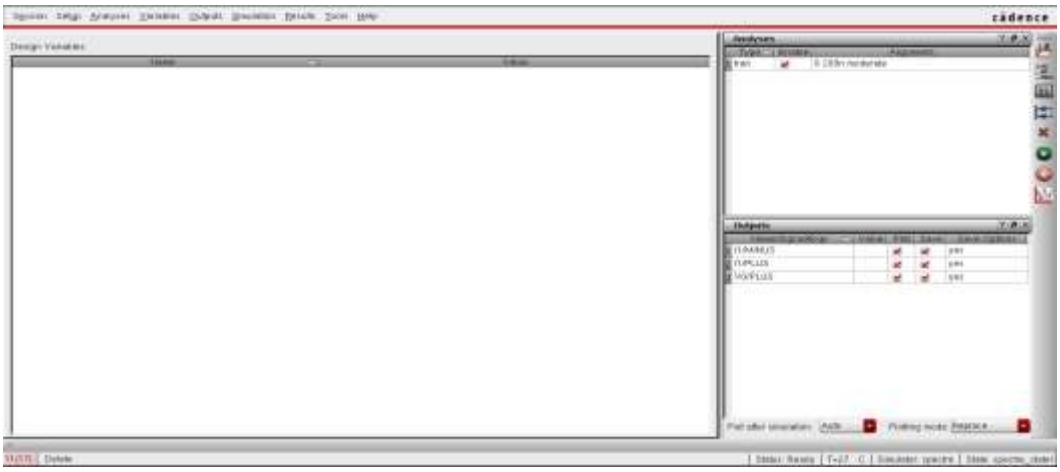
Library name	Cellview name	Properties/Comments
myDesignLib	currmirror	Symbol
analogLib	idc	DC Current = 50u
analogLib	vdc,gnd	vdd=2.5 ; vss= 0;

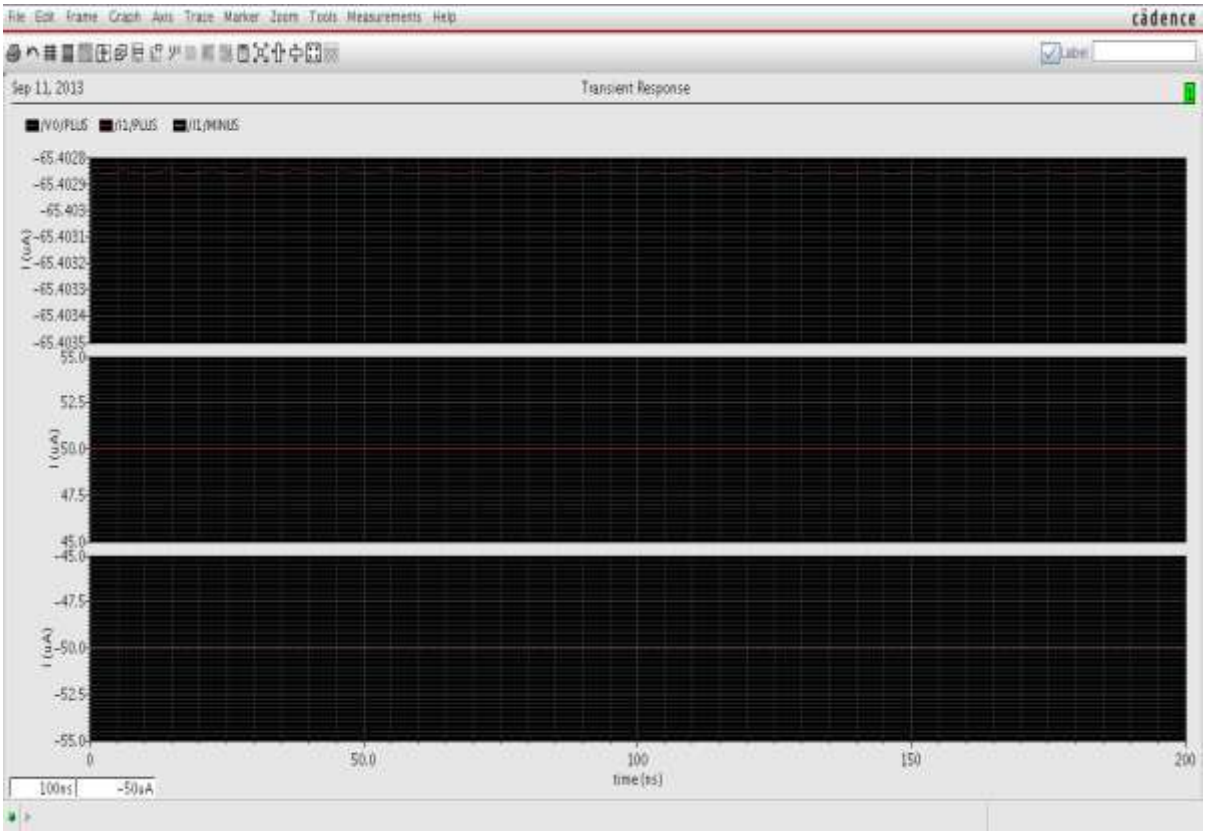


Analog Simulation with Spectre

Objective: To set up and run simulations on the currmirror_test design.

Use the techniques learned in the inverter to complete the simulation of currmirror, ADE window and waveform should look like below.





14.4 RESULT: Analyzed of simulation response of current mirror.

14.5 PRE LAB VIVA QUESTIONS:

1. Define current mirror?
2. What are the characterstics of current mirror?
3. What is the difference between current source and current mirror?

14.6 POST LAB VIVA QUESTIONS:

1. What are the advantages of current mirror?
2. What do you observe from characteristics of common mirror?

.