DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION ENGINEERING

CHITTAGONG UNIVERSITY OF ENGINEERING AND TECHNOLOGY

CHATTOGRAM – 4349, BANGLADESH

Experiment No. 7

Schematic Driven Layout Design of a NAND Gate Using Virtuoso Layout Suite Editor XL

PRECAUTIONS:

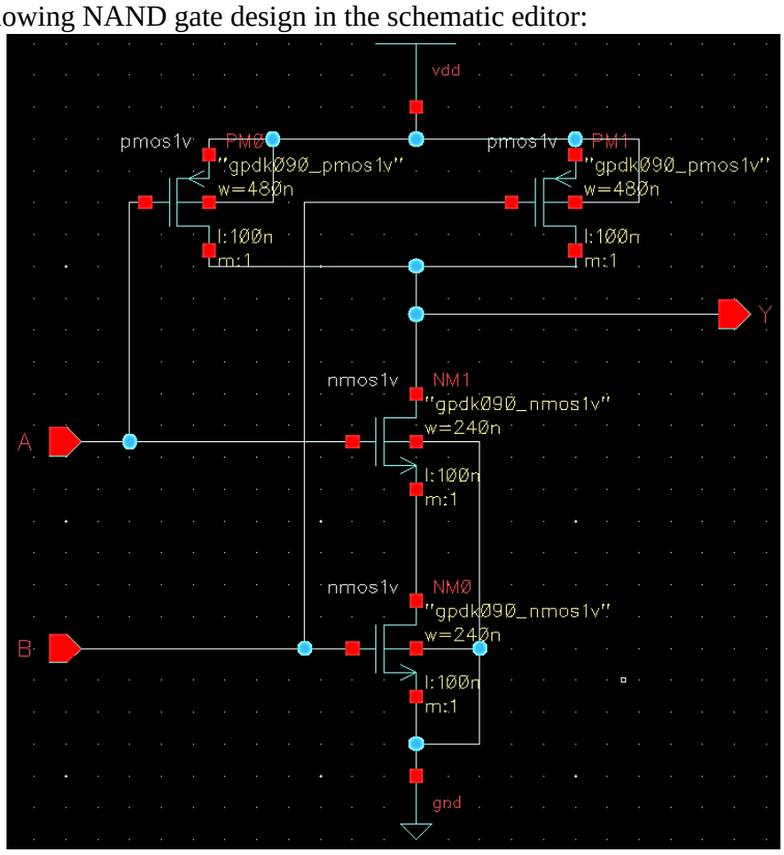
- Students must carefully read the lab manual before coming to lab to avoid any inconveniences.
- Students must carry a flash drive to transfer files and lab manuals.
- Use of mobile phone in lab is strictly prohibited and punishable offence.
- Experiment files must be uploaded to GitHub including home tasks.

OBJECTIVES:

- To familiarize with schematic-driven layout design.
- To perform schematic-level verification, DRC and LVS.
- To perform post-layout simulation of NAND Gate.

LAYOUT DESIGN USING LAYOUT EDITOR XL:

1. Virtuoso Layout Editor XL is a schematic-driven layout generation tool. To learn schematic driven layout, we will create the schematic view of a 2-input NAND gate cell which we named **nand2x1**. Execute **File > New > Cellview** from **CIW** to start the process.
2. Make the following NAND gate design in the schematic editor:

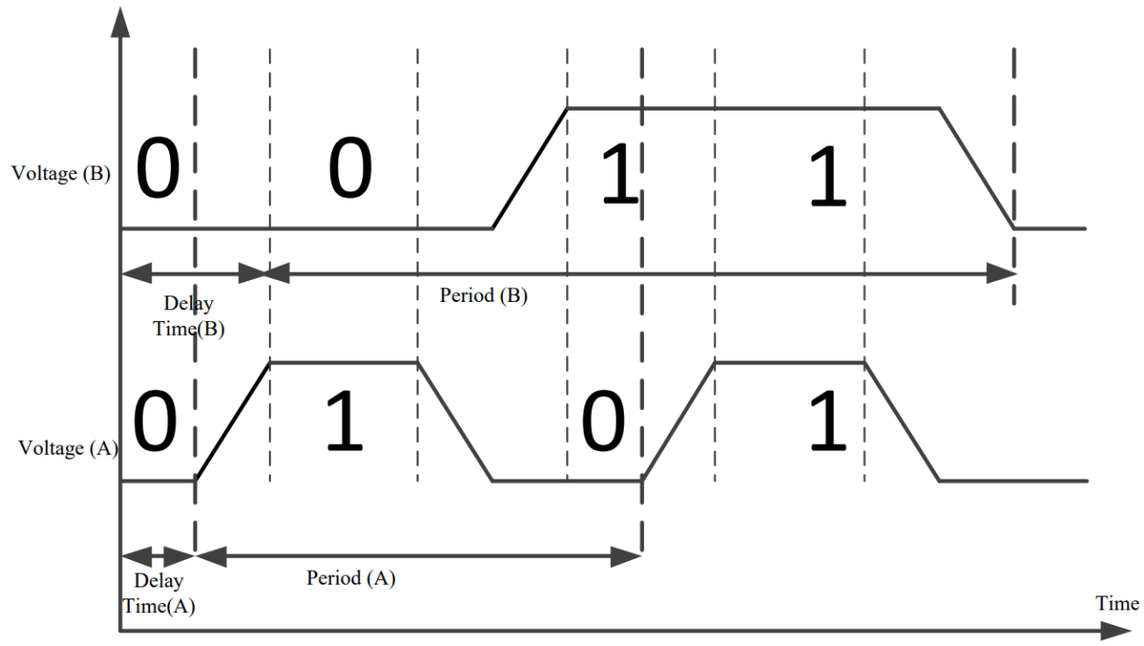


Instantiate the following cells in your schematic:

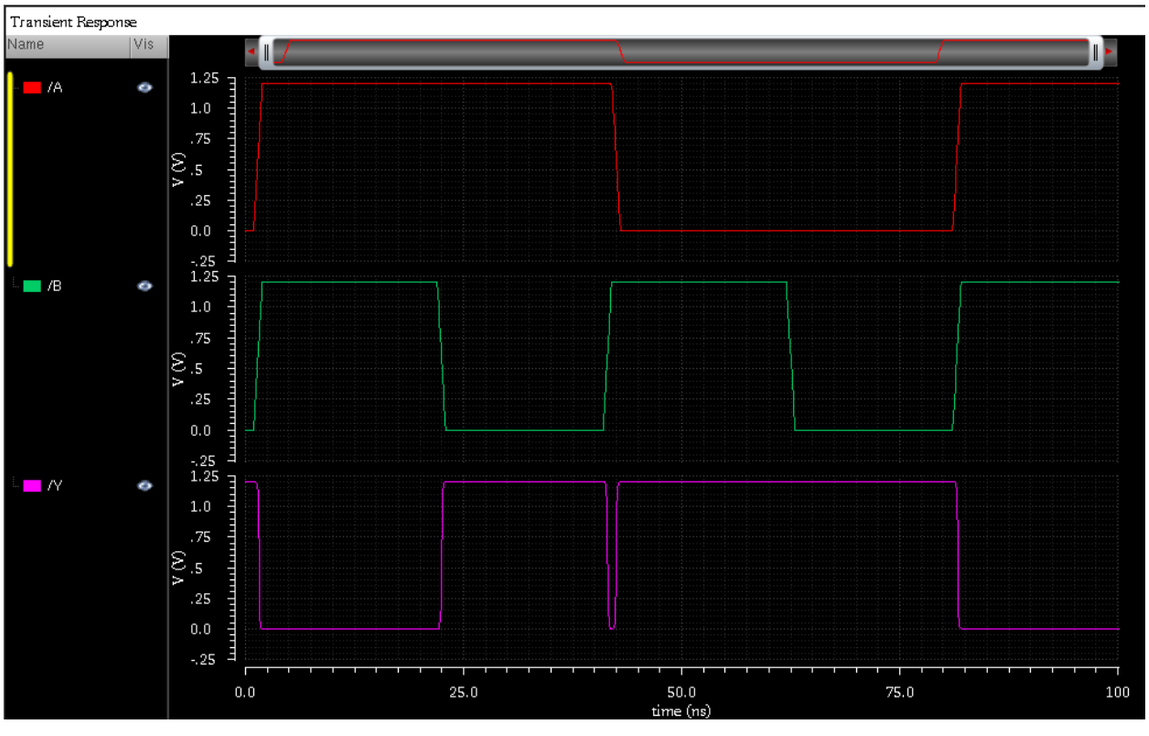
Library Name	Cell Name	Properties to Change
gpdk090	nmos1v	Width = 240n
gpdk090	pmos1v	Width = 480n
analogLib	vdd	
analogLib	gnd	

Use your experience from previous labs to properly design the NAND gate.

3. Launch ADE L and simulate the design to verify its functionality. Setup Model library, Analysis type and Outputs to be plotted as you have done in Lab1. While setting inputs for signals A and B, you have to use different periods and delays for the two signals, so that you can observe all four cases (00, 01, 10, 11) of input signals. Also make sure 0 to 1 and/or 1 to 0 transitions for both input signals do not occur at the same time. The following figure shows a sample of two signals meeting these criteria:

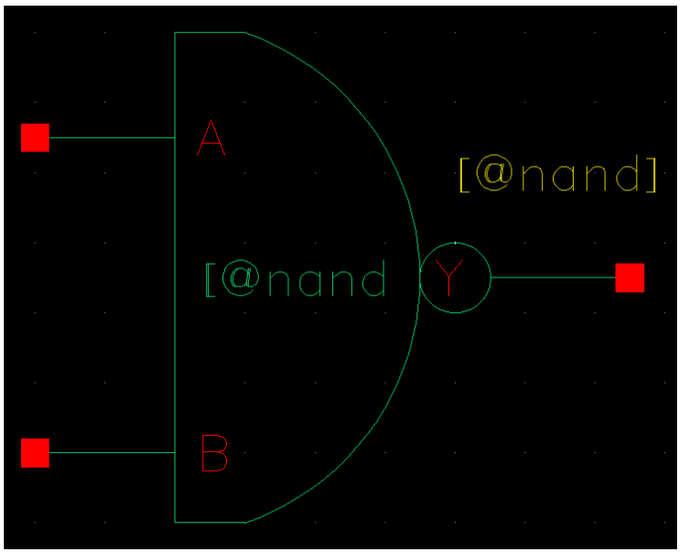


The ADE results should look like this:

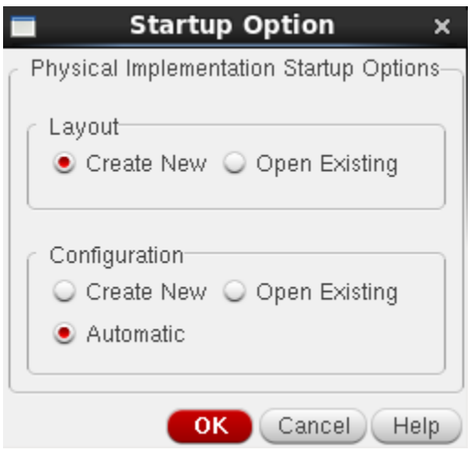


Check the functionality of the schematic (whether it acts like a NAND gate).

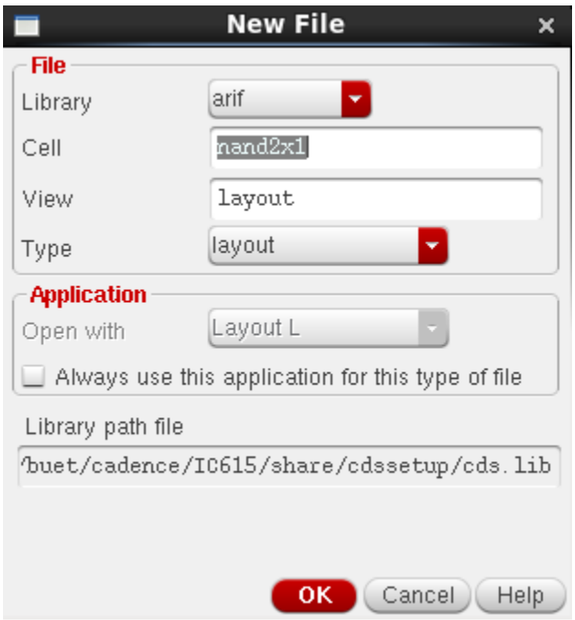
4. Then create a symbol in the same way you did in your previous labs. Make sure it looks like a NAND gate.



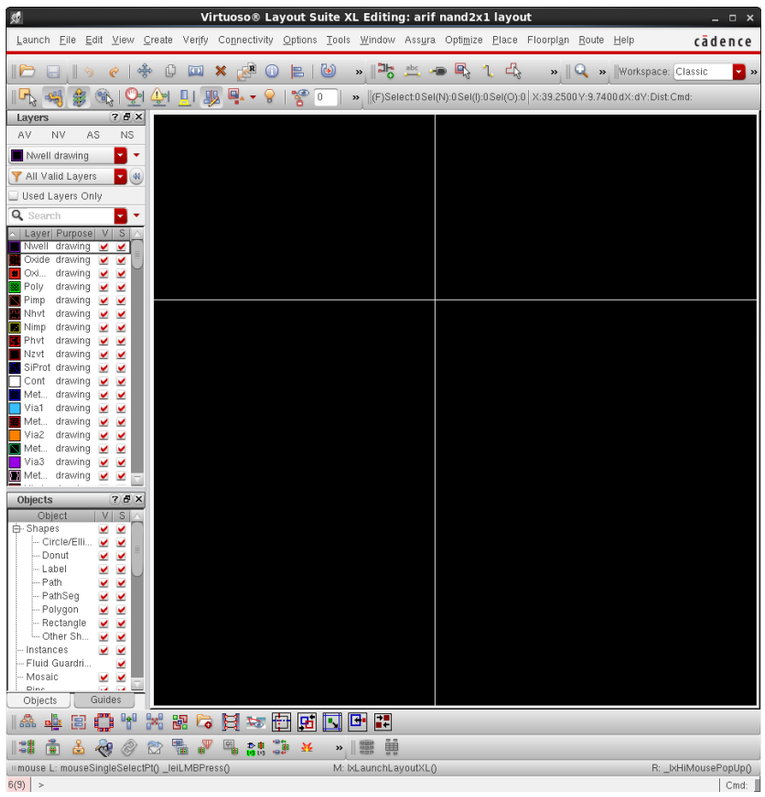
5. In schematic editor window, execute: **Launch > Layout XL**. The following window will appear:



6. Click **OK**. **'New File'** window for layout will appear. Click **OK**.

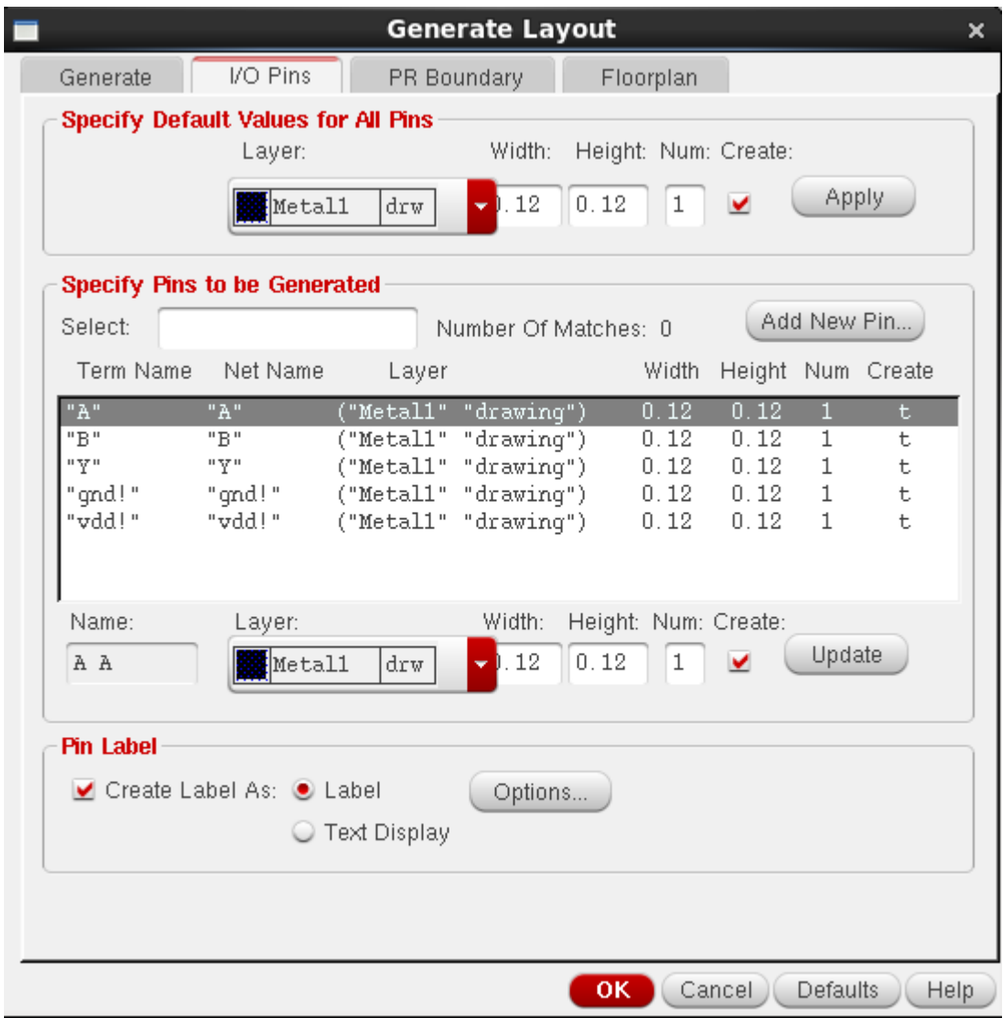


'Virtuoso Layout Editor XL' window will appear.



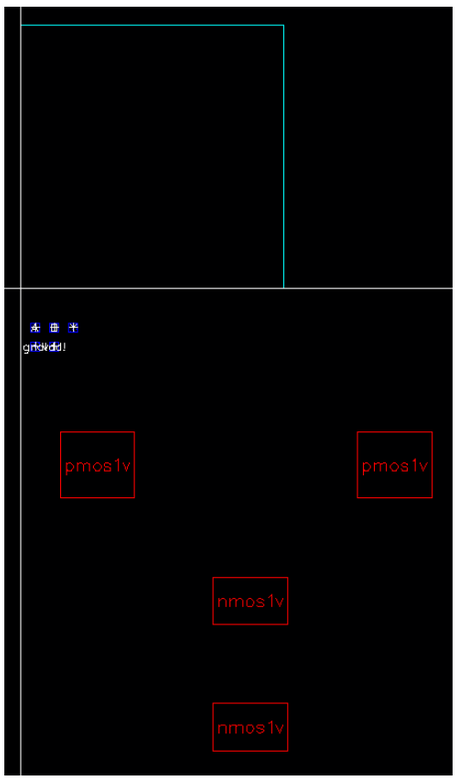
7. Execute **Connectivity > Generate > All From Source**. The following pop-up window will appear:

8. Go to the **'I/O Pins'** tab. In **'Specify Default Values of All Pins'** section, set layer to **Metal1** layer (**Metal1 drw**) and click **Apply**. Also put a tick mark on **Create Label as Label**. Click **Options**. Set Height to 0.1.

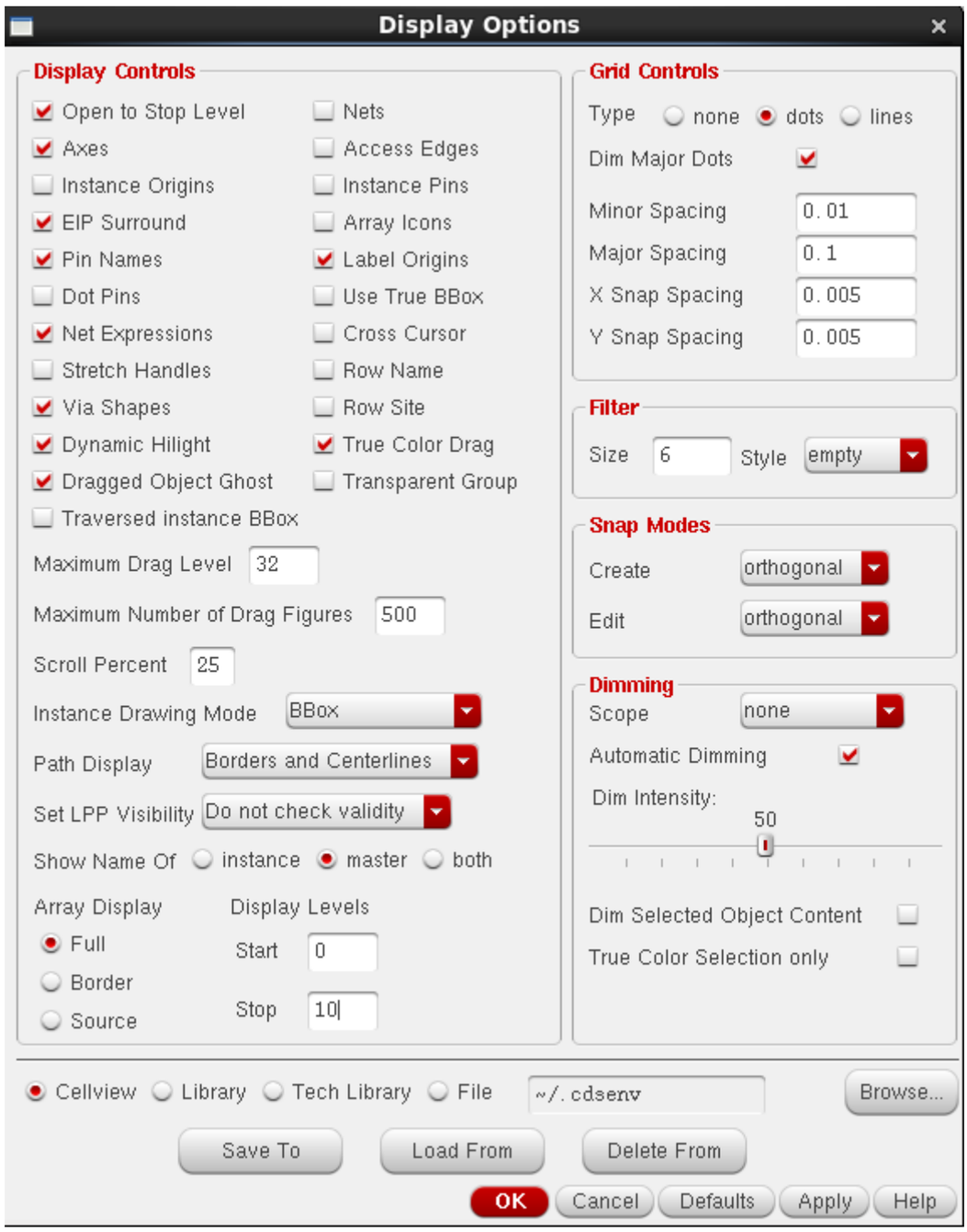


Click **OK**, and **OK**.

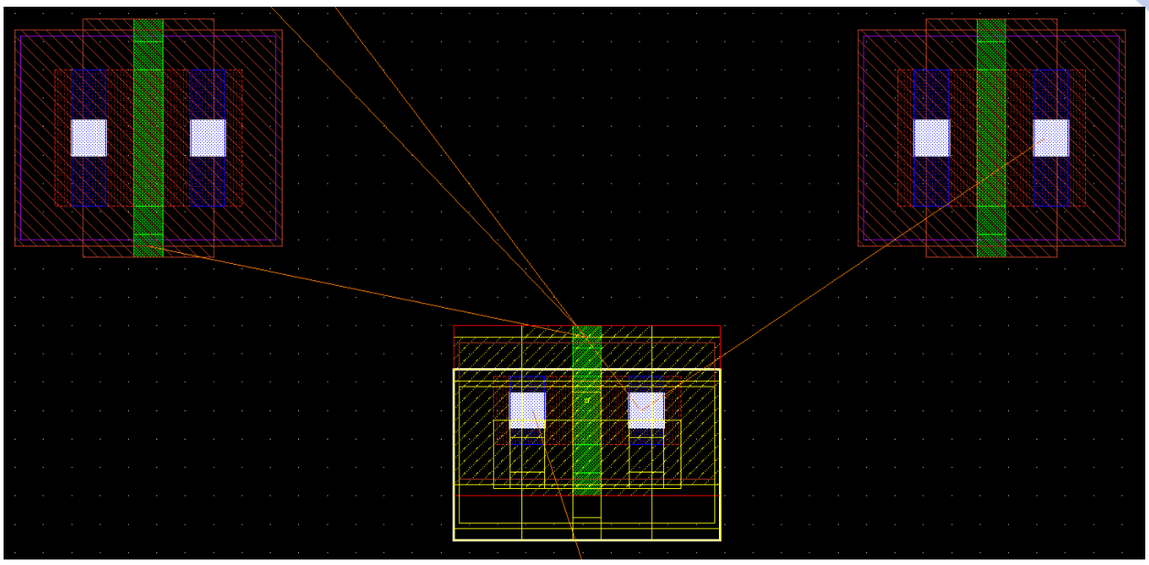
9. The initial pin and transistor placement in layout will look like the following:



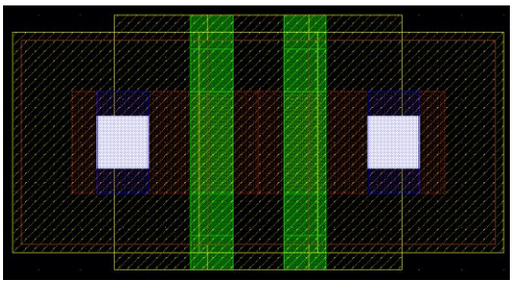
10. Execute **Options > Display** or press **'e'** on the keyboard to open **'Display Options'**. Fill it in as shown:



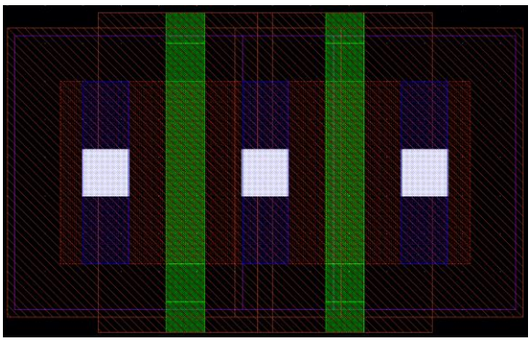
11. The transistors and pins are shown along with a bounding box, which is an estimate of the optimum size of the final layout. Automatic router will use the bounding box to constrain all routing to occur within the box. The bounding box may need to be re-sized to accommodate all components. An important concept to keep in mind during resizing is that standard cells typically have fixed height (so that power/ground rails line up correctly for routing purposes). **Delete** the **PR Boundary** for now.
    Virtuoso Layout Editor XL (VXL) and gpdk090 allow us to create stacked transistors with shared source/drain areas. Zoom in to two transistors at the bottom (to zoom in, type **"z"** and draw a box around the transistors). Click on the transistor on the right and type **"m"** to move the object. As you start dragging the object to the left, fly-lines indicating connectivity will appear as shown below:



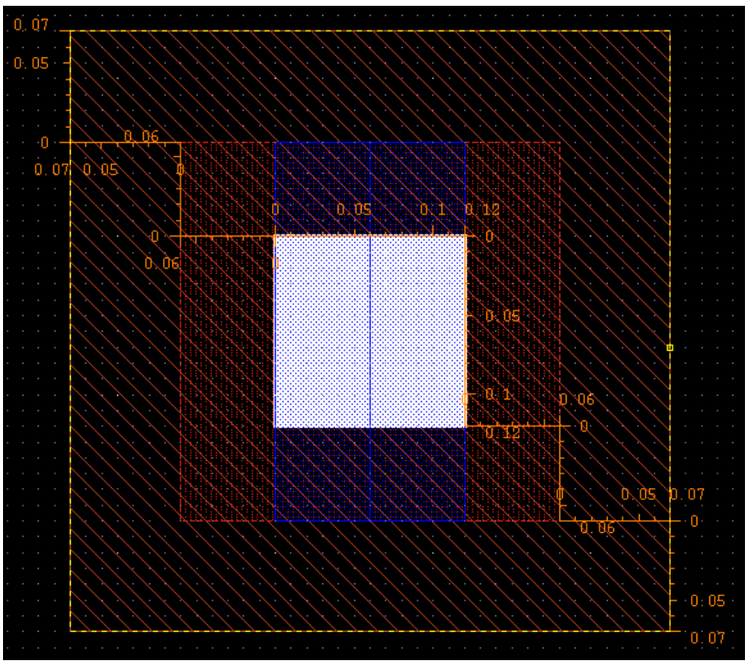
12. When the source/drain areas are overlapped, left-click to fix the position. You should see a transistor stack with shared source/drain areas like this (depending on how far you move, you may need to move left/right a bit):



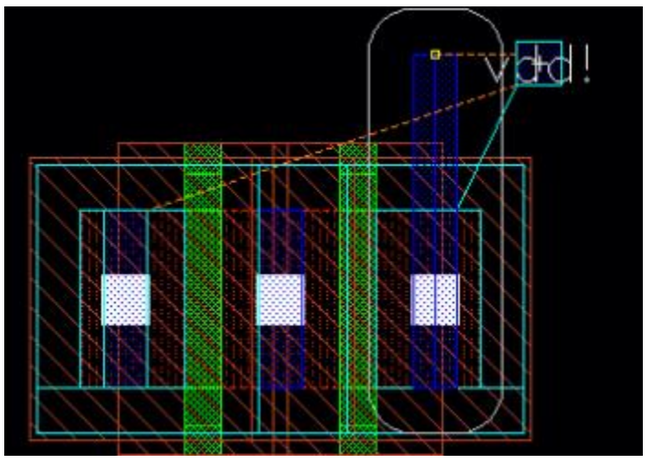
This is a nice NMOS stack for the NAND gate. As you can see, the source/drain contacts have disappeared. Back to the big picture, zoom to fit (press "f"). Let's do the same exercise for the PMOS transistors. The PMOS transistors in NAND gate do have shared drain contacts because they work in parallel. Connectivity information is extracted from schematic by VXL. The pull-up network looks like the following:



13. Now, connect different layers using path tool (press **'p'** on keyboard), and fill areas by drawing rectangles where necessary (press **'r'** on keyboard). To connect one layer to another (e.g., **Poly to Metal1** or **Metal1** to **Metal2**), create via by pressing **'o'** on keyboard and selecting proper 'Via Definition'.
14. Instantiate **M1_PSUB** and **M1_NWELL** cells (that you have created earlier) by pressing 'i' on keyboard and selecting the layout view from library browser. If you don't have the instance, create a new one.



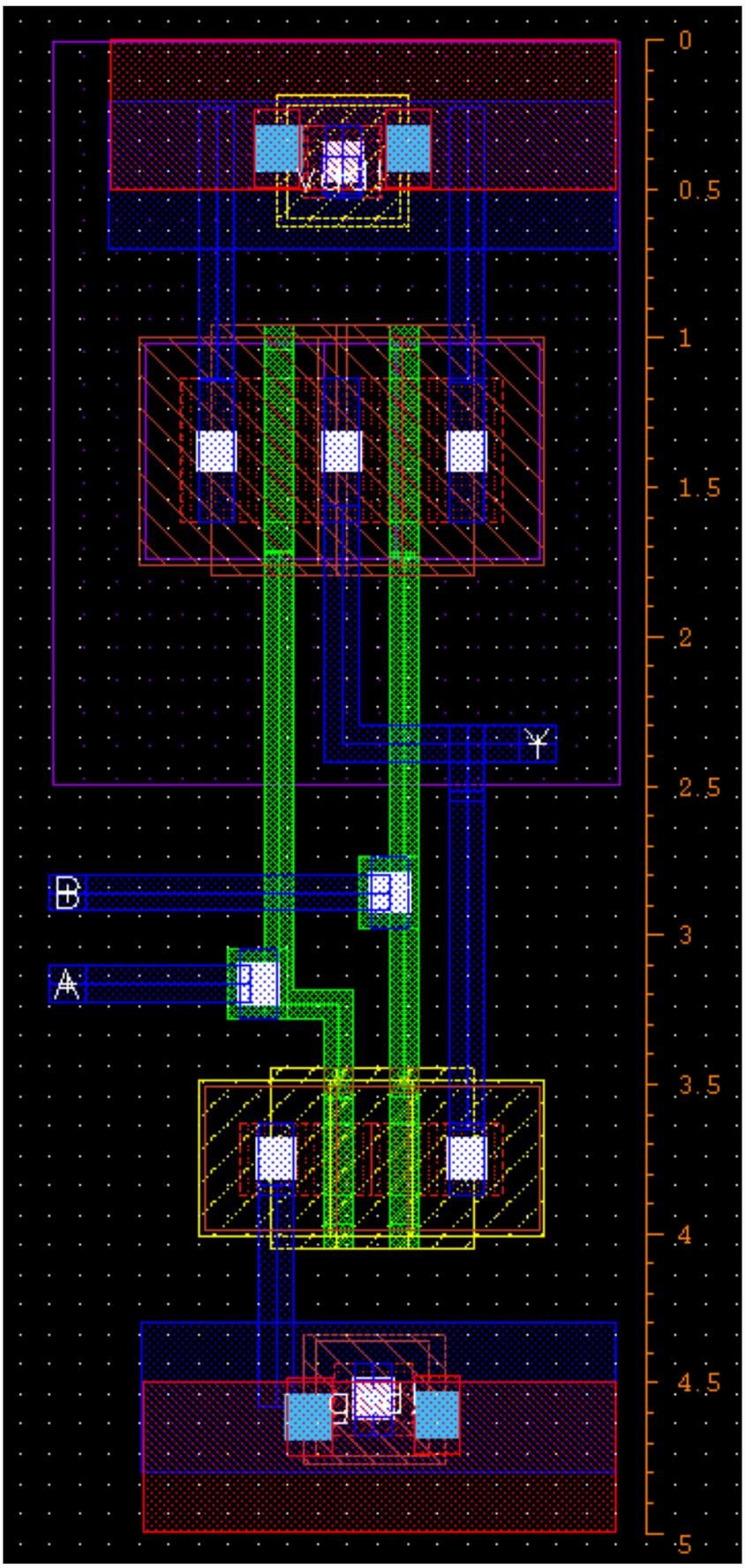
15. Wire up the layout. When you do so, you may encounter multiple options for certain pins. For example, when you select the PMOS to connect its source to VDD, there are multiple Metal1 wires in the PMOS. The desired path will be highlighted and you'll see the fly-line. Continue until you finish routing all the signals. Move **vdd**! and **gnd**! pins to the power rails. As you are moving the pins around, notice the fly-lines that indicate the connections.



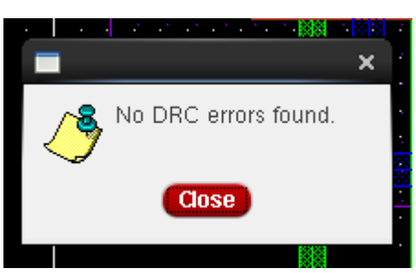
A practice that you can follow while wiring is to use Metal1 for all vertical wiring and Metal2 for all horizontal wiring inside the cell.

Also make the cell height **5 µm**.

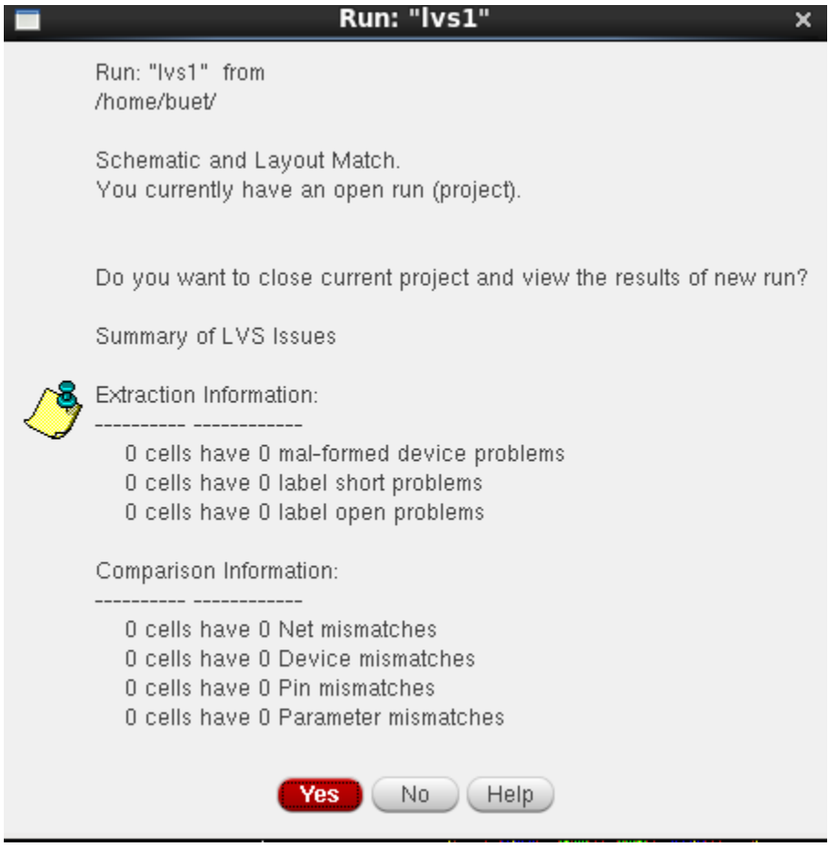
16. Your final layout will look something like the following:



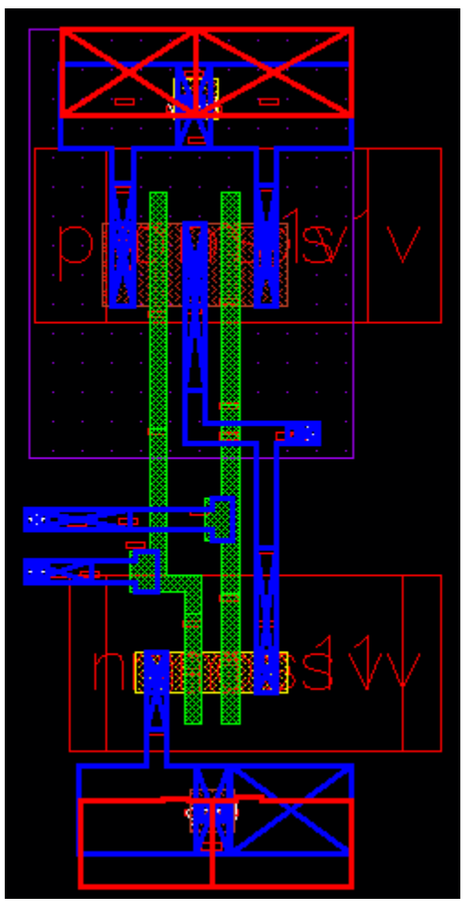
17. Now perform DRC. You should get the following:



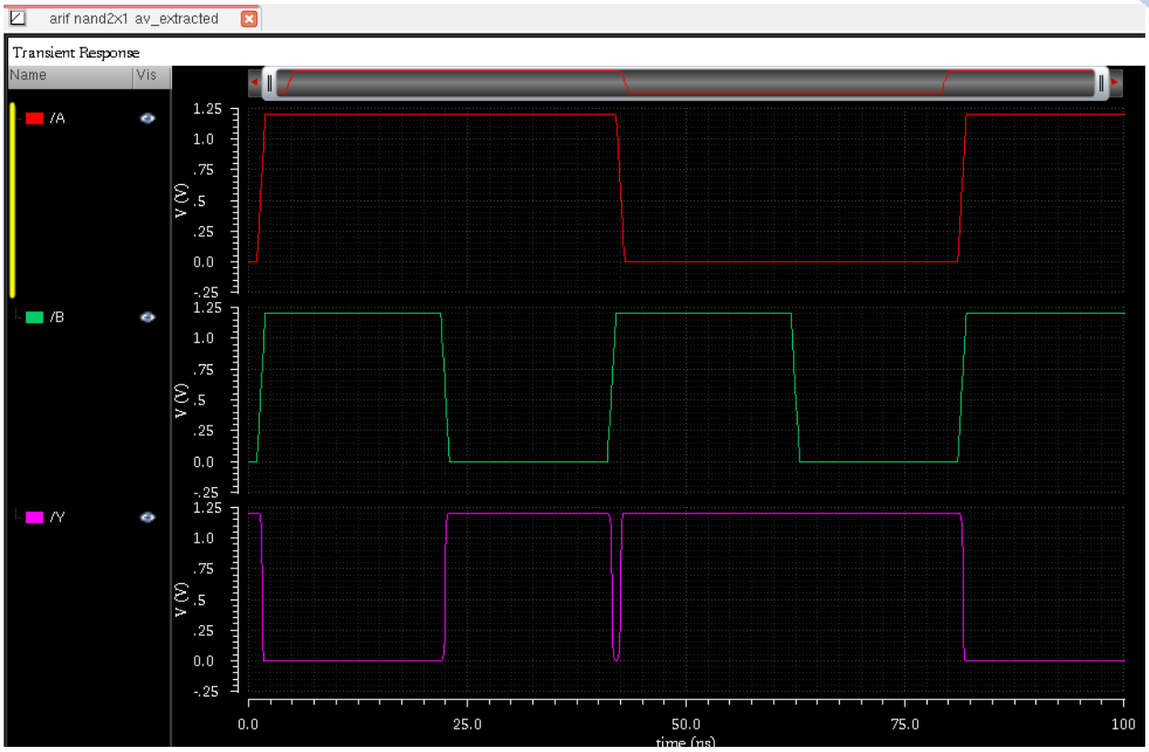
18. Perform LVS next. You should get the following:



19. Now perform RCX. You should get the **av_extracted** view.



20. Now simulate the circuit from this view to check its functionality. You should get the following output:



HOME TASK:

1. Analyze the difference between stacked and unstacked transistors in post-layout simulation and explain why we should stack transistors with common drain/source terminals.

Rubrics:

Criteria	Below Average (1)	Average (2)	Good (3)
Formatting	Report is not properly formatted, missing objectives, discussions and references.	Report is somewhat formatted but missing proper references.	Report is properly formatted.
Design	Design steps are somewhat followed and results have errors.	Design steps are properly followed and the results are flawless.	
Writing	Writing is poor and not informative. It does not address the design decisions and contains high plagiarism.	Writing is average and original. Design decisions are somewhat explored.	Writing is excellent with every design decision adequately explored.
Diagram	Diagrams are of bad quality and unreadable.	Diagrams are clear and of high quality.	

It gets easier.

Every day, it gets a little easier.

But you gotta do it every day.

That’s the hard part. But it does get easier.