Concepts in VLSI Design Laboratory

IT126IU

Instructor: Nguyen Toan Van

Lab 5
Schematic Driven Layout of a 2-input NAND gate using Virtuoso Layout Suite Editor XL

Objectives:

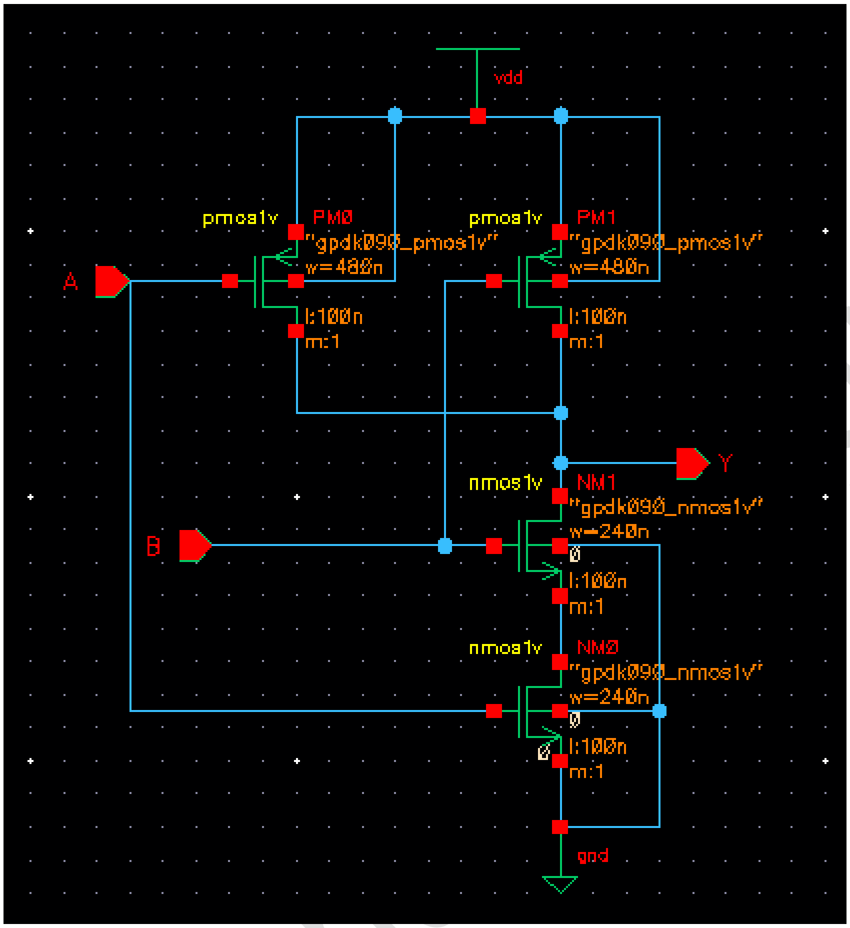
- To be familiar with schematic-driven layout with the example of a 2-input NAND gate.
- To perform Schematic Level Verification, Layout Design, DRC and LVS check and perform post-layout simulation from extracted view.

Creating Layout using Virtuoso Layout Editor XL

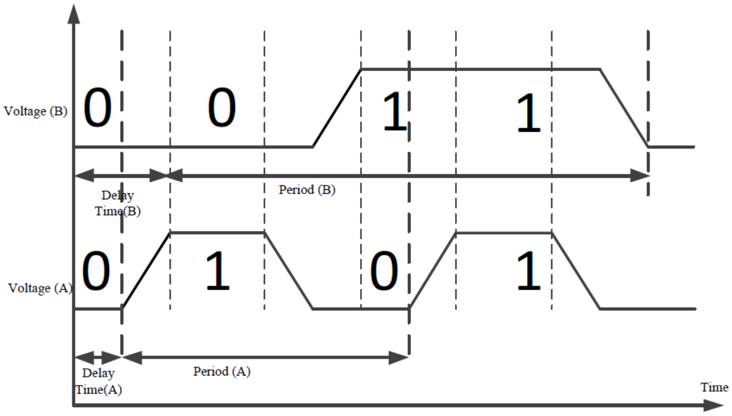
1. Virtuoso Layout Editor XL is a schematic-driven layout generation tool. To learn schematic driven layout we will create the schematic view of a 2-input NAND gate cell which we named ***NAND2X1***.
2. Instantiate the following cells to your schematic.

Library Name	Cell Name	Properties/Comment
gpdk090	*nmos1v*	For NM0 and NM1, Width = 240n
gpdk090	*pmos1v*	For PM0 and PM1, Width = 480n
analogLib	*vdd*	
analogLib	*gnd*	

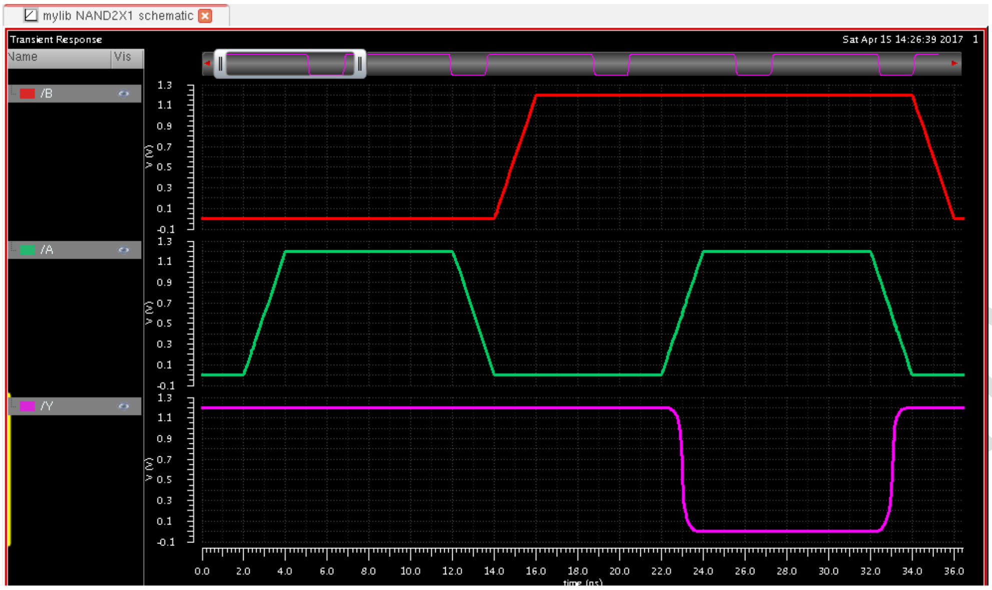
Use your experience from Lab1 to draw the schematic diagram of the nand gate.



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⍰1 and/or 1⍰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:

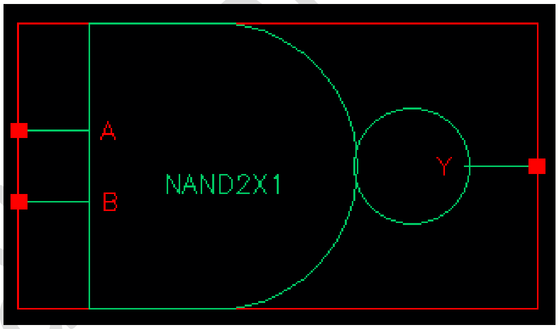


A sample waveform window would look like the following after simulation:

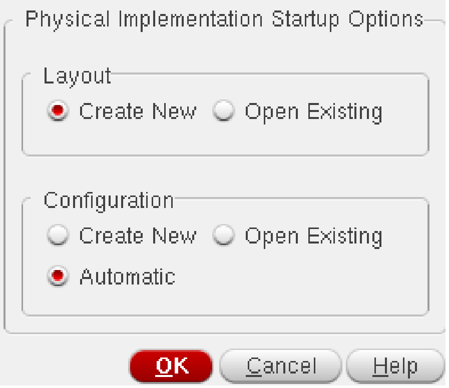


Check the functionality of the schematic (whether it acts like a nand gate).

4. Then create a symbol in the same way you have made symbol of inverter in lab2. Make it look like a nand gate.



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



Physical Implementation Startup Options

Layout

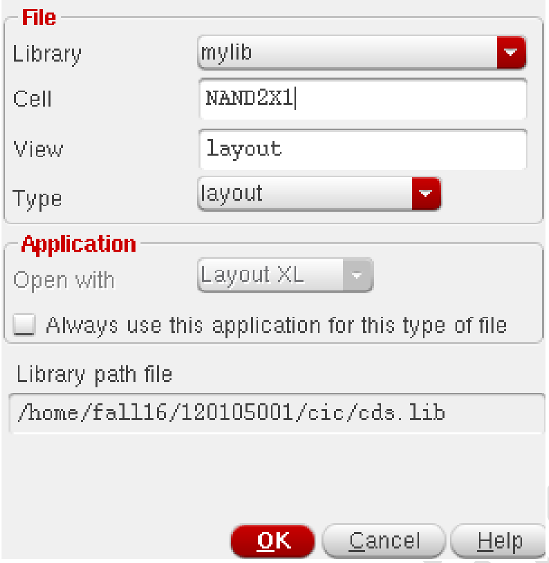
● Create New ○ Open Existing

Configuration

○ Create New ○ Open Existing

● Automatic

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



File

Library: mylib

Cell: NAND2X1

View: layout

Type: layout

Application

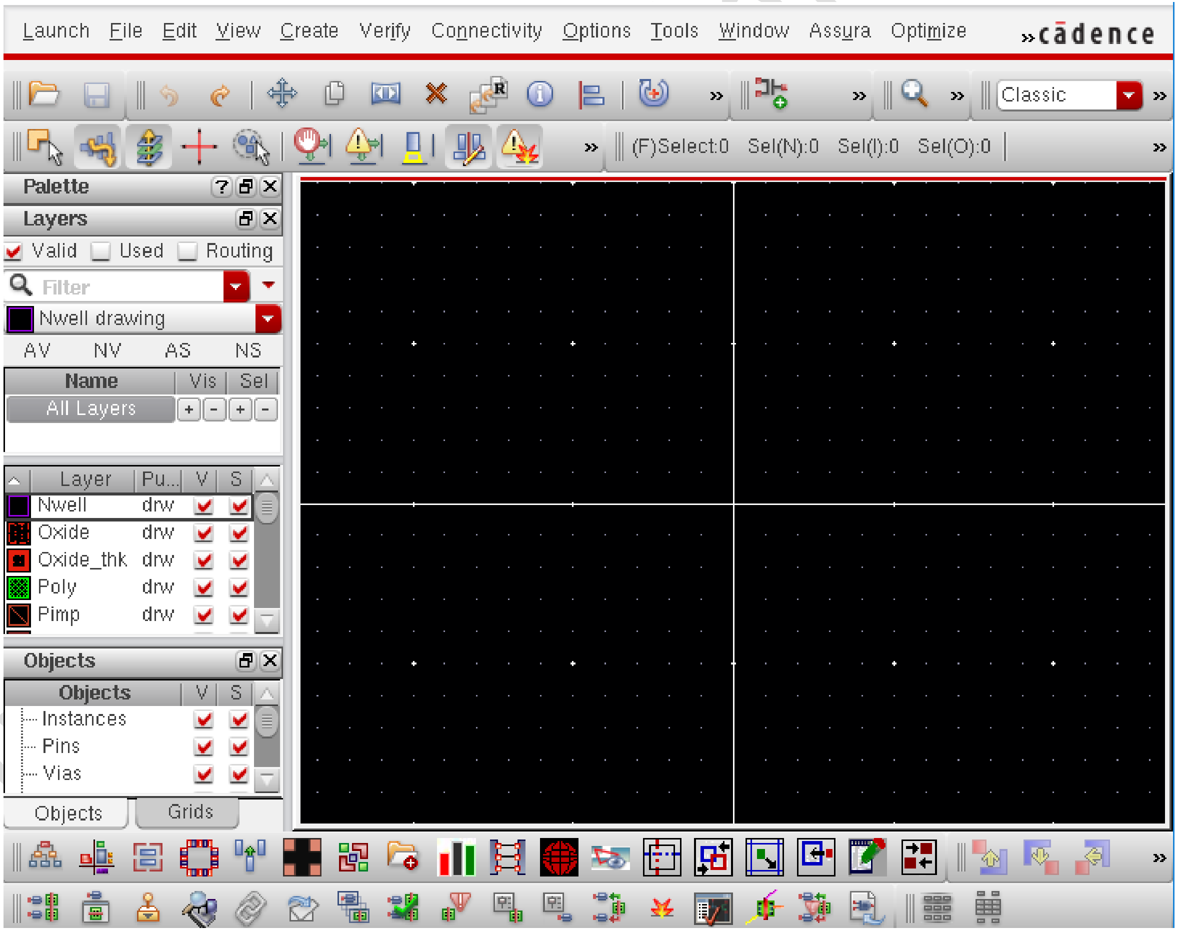
Open with: Layout XL

☐ Always use this application for this type of file

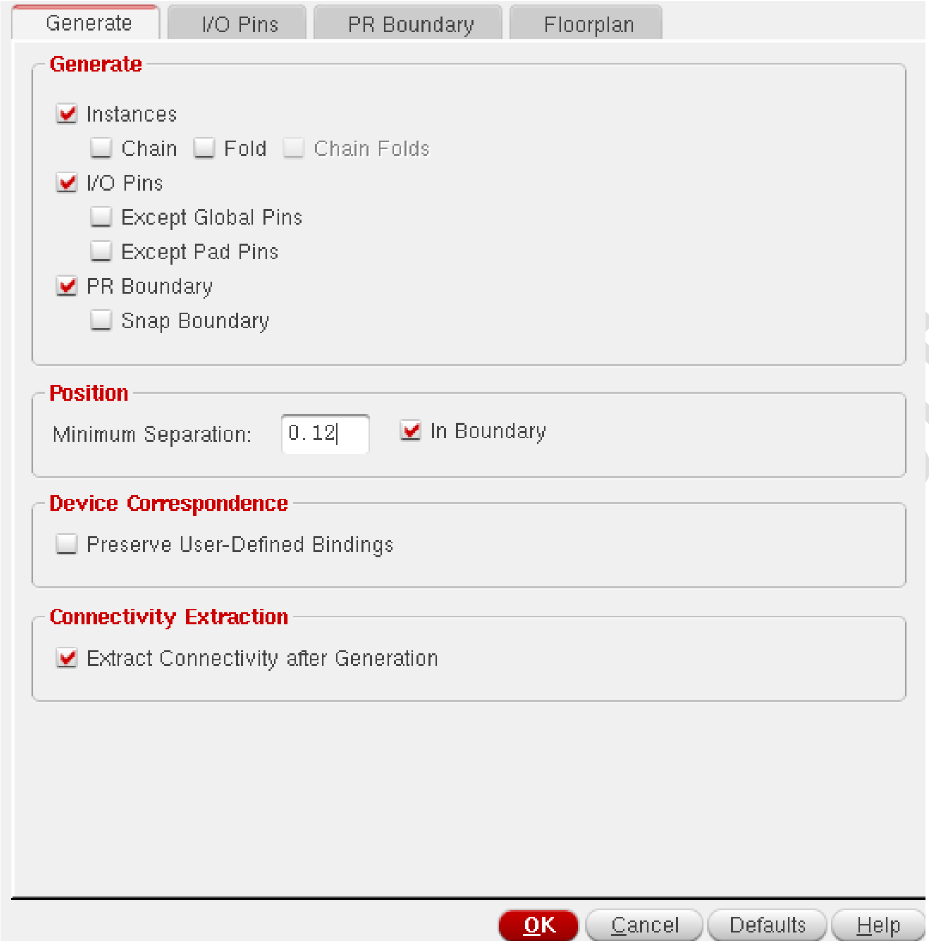
Library path file

/home/fall16/120105001/cic/cds.lib

'Virtuoso Layout Editor XL' window will appear.



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



Generate

☑ Instances

□ Chain □ Fold □ Chain Folds

☑ I/O Pins

□ Except Global Pins

□ Except Pad Pins

☑ PR Boundary

□ Snap Boundary

Position

Minimum Separation: 0.12 ☑ In Boundary

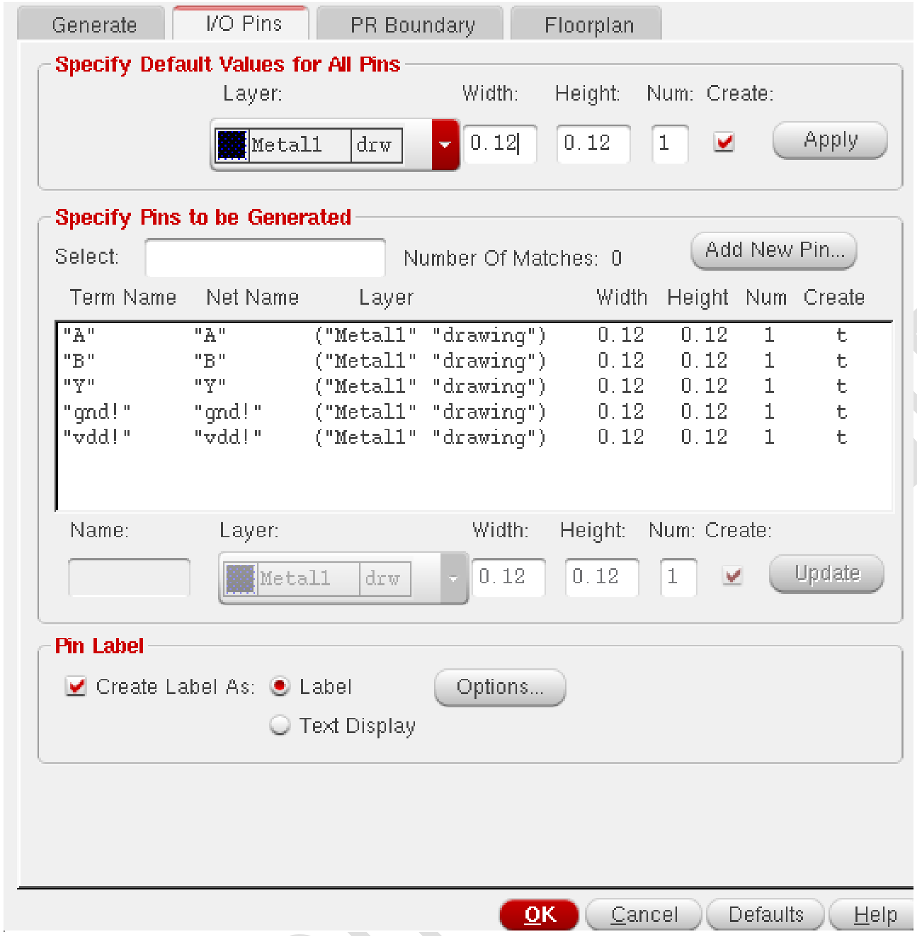
Device Correspondence

□ Preserve User-Defined Bindings

Connectivity Extraction

☑ Extract Connectivity after Generation

8. Go to **I/O pins** tab. The dialog box shows that all I/O pins are in **Metal1** layer (**Metal1 drw**). Also put a tick mark on **Create label as Label**. Click **Options**. Set **Height** to 0.1.



Specify Default Values for All Pins

Layer:	Width:	Height:	Num:	Create:
Metal1 drw	0.12	0.12	1	☑

Specify Pins to be Generated

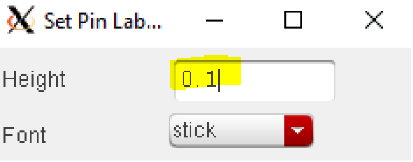
Select: Number Of Matches: 0

```
Term Name   Net Name   Layer                     Width  Height  Num  Create
"A"         "A"        ("Metal1" "drawing")      0.12   0.12    1    t
"B"         "B"        ("Metal1" "drawing")      0.12   0.12    1    t
"Y"         "Y"        ("Metal1" "drawing")      0.12   0.12    1    t
"gnd!"      "gnd!"     ("Metal1" "drawing")      0.12   0.12    1    t
"vdd!"      "vdd!"     ("Metal1" "drawing")      0.12   0.12    1    t
```

Name:	Layer:	Width:	Height:	Num:	Create:
	Metal1 drw	0.12	0.12	1	☑

Pin Label

☑ Create Label As: ● Label ○ Text Display

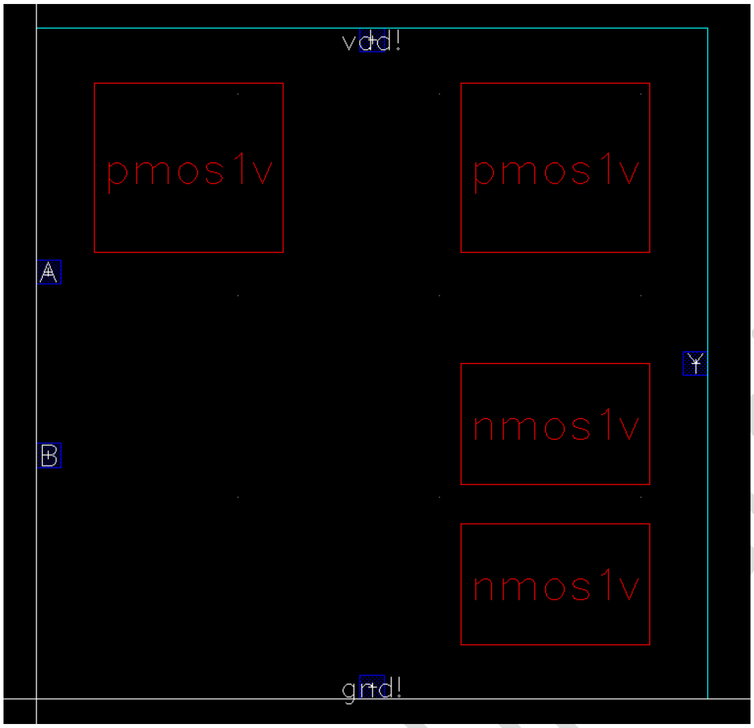


Height 0.1

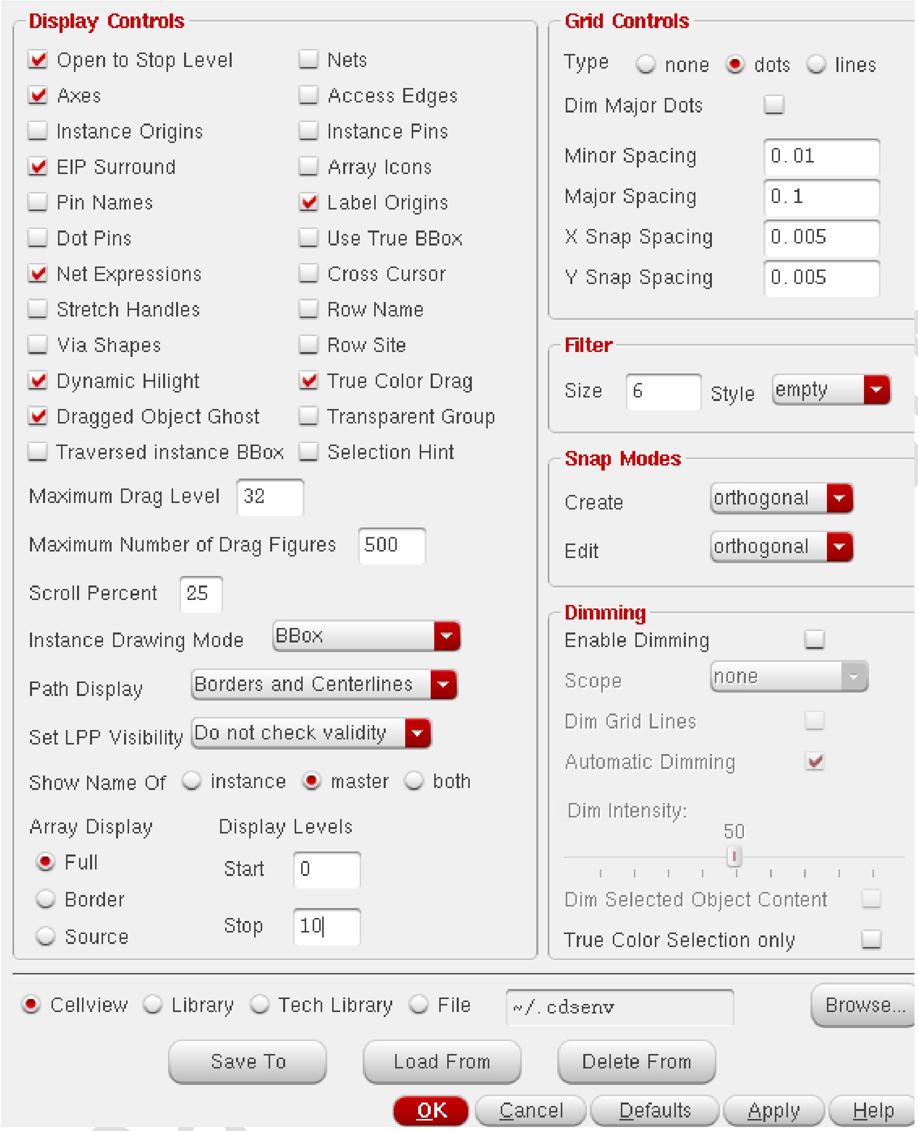
Font stick

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 keyboard to open ‘**Display Options**’. Fill it in as shown:



Display Controls

☑ Open to Stop Level
☑ Axes
☐ Instance Origins
☑ EIP Surround
☐ Pin Names
☐ Dot Pins
☑ Net Expressions
☐ Stretch Handles
☐ Via Shapes
☑ Dynamic Hilight
☑ Dragged Object Ghost
☐ Traversed instance BBox

☐ Nets
☐ Access Edges
☐ Instance Pins
☐ Array Icons
☑ Label Origins
☐ Use True BBox
☐ Cross Cursor
☐ Row Name
☐ Row Site
☑ True Color Drag
☐ Transparent Group
☐ Selection Hint

Maximum Drag Level: 32
Maximum Number of Drag Figures: 500
Scroll Percent: 25
Instance Drawing Mode: BBox
Path Display: Borders and Centerlines
Set LPP Visibility: Do not check validity
Show Name Of: ○ instance ● master ○ both

Array Display: ● Full ○ Border ○ Source
Display Levels: Start 0, Stop 10

Grid Controls

Type: ○ none ● dots ○ lines
Dim Major Dots ☐
Minor Spacing: 0.01
Major Spacing: 0.1
X Snap Spacing: 0.005
Y Snap Spacing: 0.005

Filter

Size: 6, Style: empty

Snap Modes

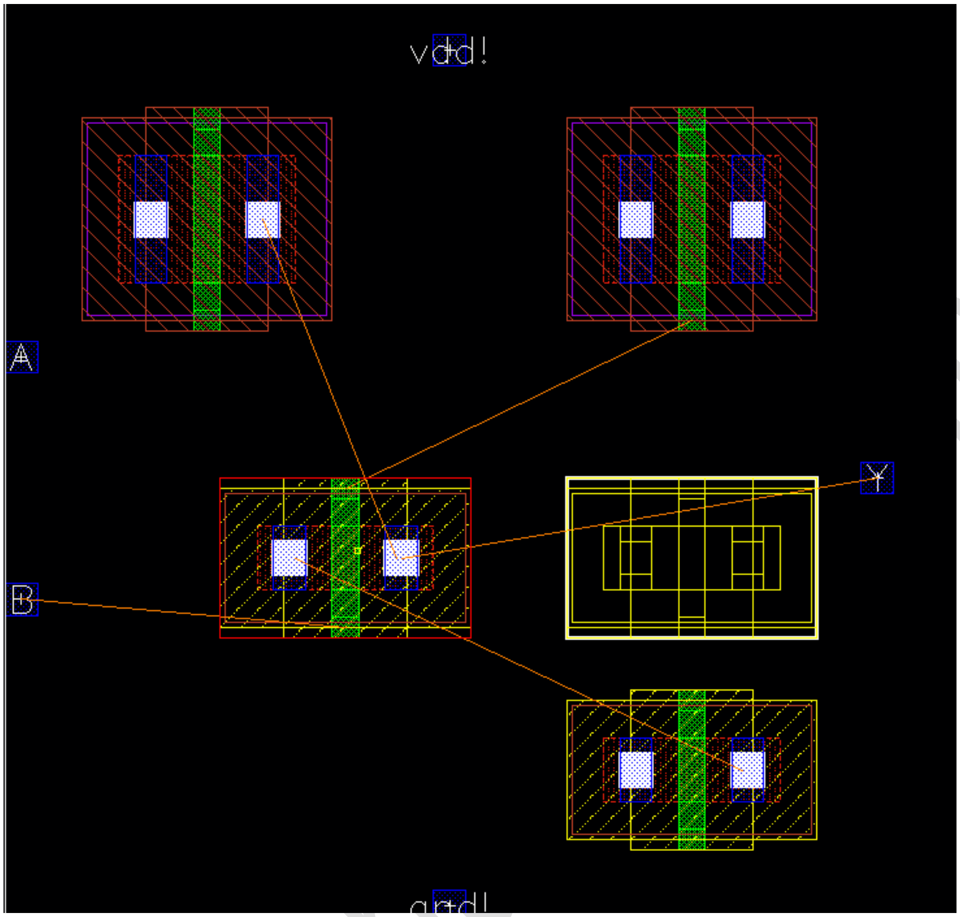
Create: orthogonal
Edit: orthogonal

Dimming

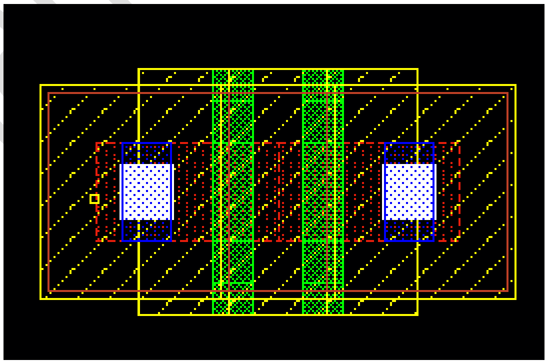
Enable Dimming ☐
Scope: none
Dim Grid Lines ☐
Automatic Dimming ☑
Dim Intensity: 50
Dim Selected Object Content ☐
True Color Selection only ☐

● Cellview ○ Library ○ Tech Library ○ File: ~/.cdsenv

11. The transistors and pins are shown inside 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:

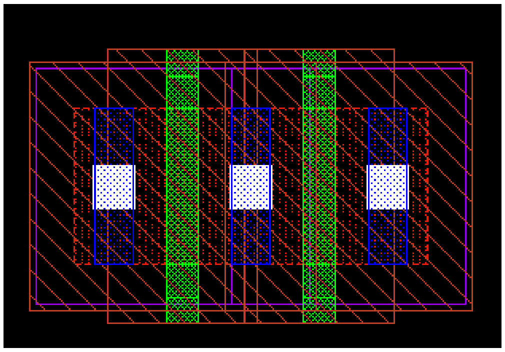


12. When the source/drain areas are overlapped, left-click to fix the position. You should see a transistor stack with shared source/drain areas like this (depending on how far you move, you may need to move left/right a bit):

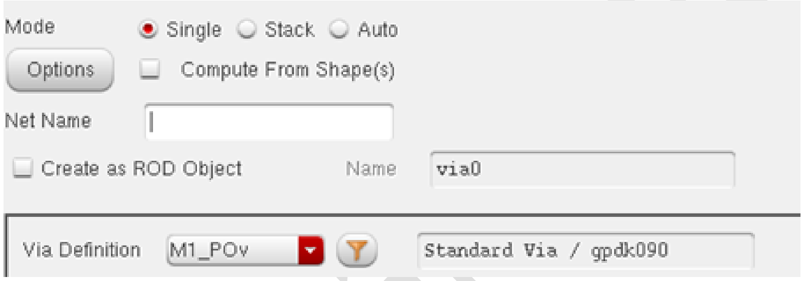


This is a nice NMOS stack for the NAND gate. As you can see, the source/drain contacts have disappeared. Back to the big picture, zoom to fit (press "**f**").

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 Defintion’**.



Mode ● Single ○ Stack ○ Auto

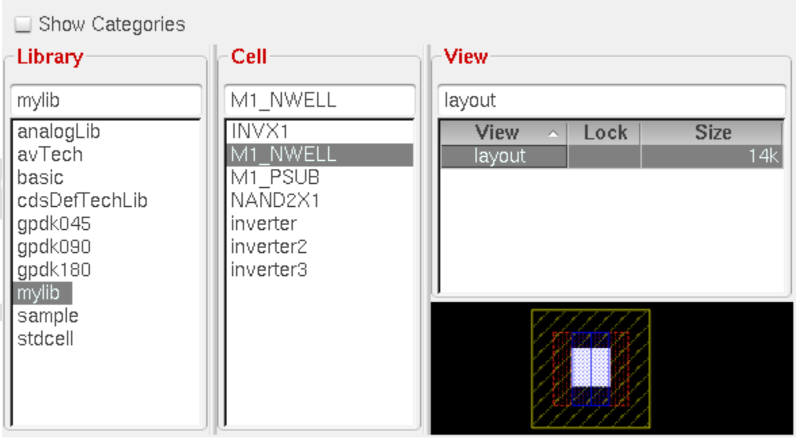
Options □ Compute From Shape(s)

Net Name

□ Create as ROD Object Name via0

Via Definition M1_POv Standard Via / gpdk090

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.



□ Show Categories

Library: mylib

analogLib
avTech
basic
cdsDefTechLib
gpdk045
gpdk090
gpdk180
mylib
sample
stdcell

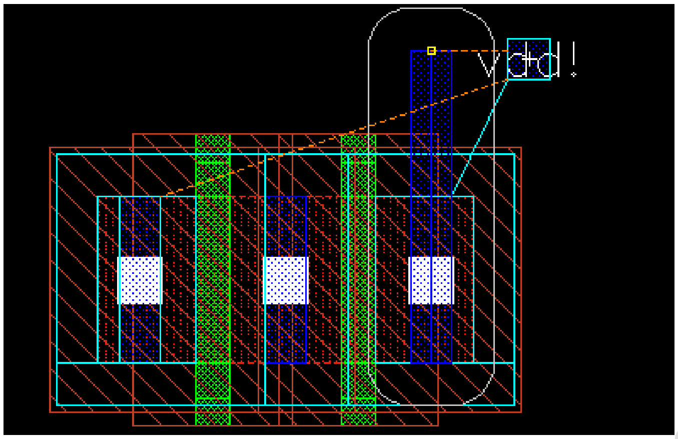
Cell: M1_NWELL

INVX1
M1_NWELL
M1_PSUB
NAND2X1
inverter
inverter2
inverter3

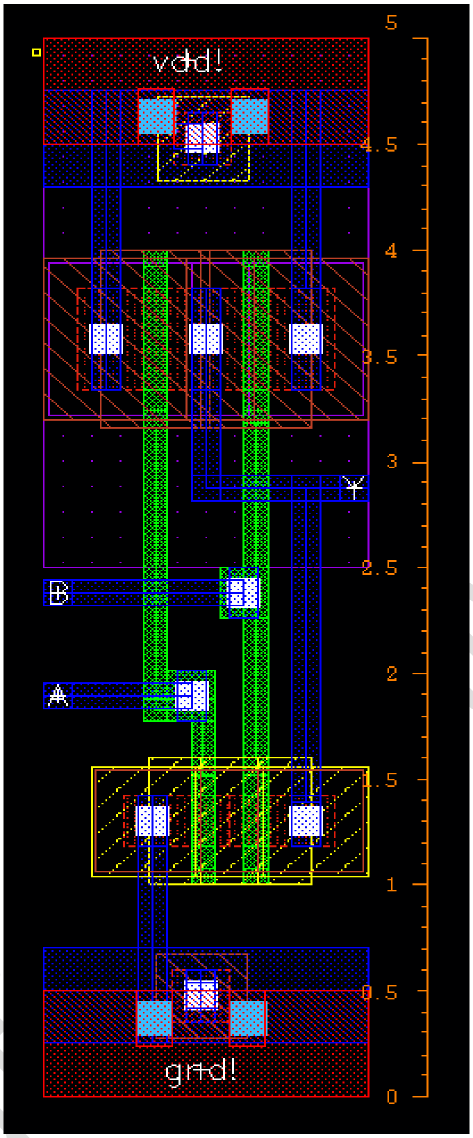
View: layout

View	Lock	Size
layout		14k

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



17. Perform DRC, LVS and QRC for NAND2X1 as you have done in Lab 4. Generate **av_extracted** view and simulate the circuit from that view to verify the functionality.

QUESTIONS:
Why should you stack transistors in a layout which have shared drain/source? What are the differences between stacked transistors and unstacked ones in terms of performance of post-layout simulation?