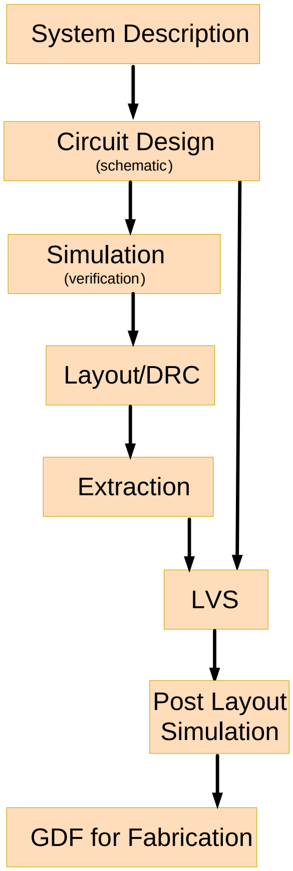
EE 330 Laboratory 5
From Boolean Equation to Silicon

Fall 2024

Contents

Background Information:

The objective of this experiment is to experience the design flow of a digital system from the system description through layout where more than one designer is a key contributor to the design. This will be approached by using a simple Boolean system and two designers.



In this experiment, one designer will create a 3-input NAND gate and a second designer will create a 3-input NOR gate. Both should create an inverter. By sharing the gate you create with a partner, you will have all three gates. As part of the prelab for this experiment you will select a Boolean function that can be implemented (without Boolean simplification) with 3-input NAND gates, 3-input NOR gates, and Inverters. You will then implement the Boolean function you selected in silicon (the layout) given area and pin constraints. You can re-use an inverter designed previously or design an inverter for use in this experiment.

Of particular importance when sharing these gates is that they are compatible with the overall floorplan you have for your layout. Horizontal rails for power (VDD) and ground are widely used in the layout of digital circuits with the logic blocks themselves placed between these rails. At a minimum, the floorplan of both designers should have the same distance between the VDD and ground rails.

A simplified Custom IC design flow for an analog circuit and small digital circuits is shown in the flow chart. From the flow chart, it can be observed that we have now studied most of the basic skills needed to do a complete design of a simple circuit. In this experiment, we will take the system description of a system in the form of a simple Boolean expression and convert it to a layout for fabrication through MOSIS. Before you start the design, consider your layout technique, specifically the floorplan. This will allow you to save considerable time in later parts of the experiment.

When considering the floorplan you will be using to implement the Boolean function in Part 2.3, you will find it useful to **layout your gates so that it is easy to interconnect them** in the layout of the Boolean function. For example, in addition to having a **VDD rail (BUS) on the top** of your layout and a **VSS (Ground) rail on the bottom**, it might be convenient to place all **inputs on the left** and the **output to the right** in your cells. With this approach, layout of the Boolean function could involve running signal paths horizontally in the layout. If the layout becomes to long horizontally, pairs of VDD and VSS rails (still equally spaced) can be stacked vertically.

By having you and your partner agree on the floorplan, most importantly the spacing between VDD and VSS busses, the interconnection of the NAND gates, NOR gates, and inverters will be simplified in Part 3 of this experiment.

Checkpoints

The checkpoints for this lab are as follows:

1. NAND/NOR DRC and LVS from Part 1
2. Exchanged NAND/NOR gate
3. Boolean Function Gate-Level Schematic
4. Boolean Function schematic TB
5. Boolean Function DRC and LVS

Remember, all checkpoints must be shown to a lab TA before the report is submitted. You should include these checkpoints in your lab report.

Part 1: 3 - Input NAND or NOR Gate

As discussed in the prelab, create an Inverter along with a NAND gate or a NOR gate as agreed upon between you and your partner. **Use pcells** (the instances from Lab 3) for the transistors when creating the layout. In a previous experiment you created a schematic and test bench, so following the same approach, create a **layout** of the gates you designed. Be sure your layout passes both DRC and LVS.

When creating the gates for this experiment, use minimum allowable widths and lengths for the NMOS and PMOS devices. Though not of emphasis in this experiment, sizing of devices that are larger than minimum are often preferred in many applications. It will be easiest to achieve such sizing requirements if pcells with multiplicity and fingers are used.

Part 2: Exchanging gates

Exchange gates using the following steps. Though these steps are described for exchanging the cell named **“inverter”**, they can be used for exchanging other cells as well. You will need to

```
# Navigate to your lablib folder:
cd ~/ee330/lablib
# Modify the permissions of the files you want to transfer
chmod 777 -R ~/ee330/lablib
# Create an archive of the cell
tar cvf ~/inverter.tar inverter
```

“pack” the **inverter** cell in the **lablib** library in a single file. Close Virtuoso, open a new terminal window, and from command line run the following:

The file **inverter.tar** in the home directory now contains all the views required to use the **inverter** cell along with read, write and execute (rwx) access for others to use **inverter** and all subdirectories.

To send this file to another person, simply send it as an attachment to an email message, or a shared object in Google Drive. The recipient should then download it into their home directory. The following steps describe how the recipient can then use of **inverter** in their design. To keep the imported designs separate, create a new library called **exchange** (or some other name) and attach it to the tsmc18rf process. Then run the following:

```
# Move the inverter.tar from home to exchange:
mv ~/inverter.tar ~/ee330/exchange
# Move to the exchange directory
cd ~/ee330/exchange
# Extract the design
tar xvf inverter.tar
```

The inverter cell will now appear in the **exchange** library. Instantiate (create an instance of **inverter** in your schematic using the “i” hot key) the symbol and the **layout** views from this cell to use **inverter** in your own design.

Part 3: Boolean Function

Implement the Boolean function from the prelab by going through the complete design flow (schematic, symbol, test bench, layout, LVS). In this implementation, adhere to the constraints given below. Assign an appropriate name for the cell you create that implements the Boolean function. In this implementation, use the Inverter you created in the previous lab, your NAND or NOR gate from Part 1, and your partners NAND or NOR gate from their Part 1.

By re-using cells that were created, considerable time and effort should be saved since you should not need to place any individual transistors when implementing the Boolean function. The test bench should verify proper output values for all 8 input combinations of A, B, C.

Do not **simplify the Boolean expression** you have chosen. If you follow the floor planning suggestion of placing VDD on the top, VSS on the bottom, inputs on the left, and outputs on the right, the layout of your logic circuit comprising the Inverter, NAND, and NOR gates, your layout may have gate placements that look something like that shown in Fig. 1 (depicted for 2-input rather than 3-input gates). Of course, the actual layout will be with transistors, not symbols for the gates.

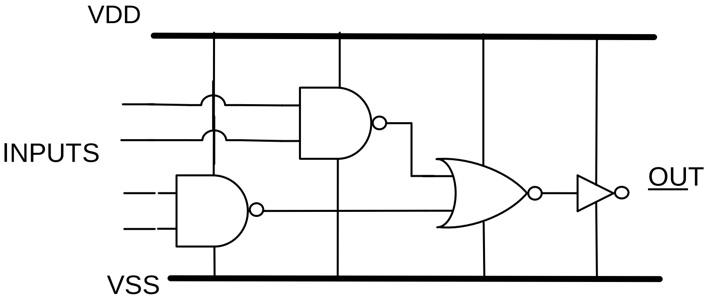


Figure 1: Cell input, output, signal flow, and power directions

Design Constraints

- You may only use the NAND, NOR and inverters you previously created.
- The Boolean function you selected should not be simplified
- The completed cell should fit inside a square of dimensions 50µm×50µm. (Use a ruler)
    - 50µm×50µm (Base Requirements)
    - 45µm×45µm (5% Extra Credit)
    - 35µm×35µm (10% Extra Credit)
- The inputs (A, B, C,) for the complete cell must be available at the left boundary of the cell and enter the cell with Metal1.
- The output Y of the completed cell must be available at the right boundary of the cell and emerge with Metal1.
- Use a 3µ wide Metal1 bus at the top of the complete cell for VDD and a 3µ wide Metal1 bus at the bottom of the complete cell for VSS. Use these rails to supply power to your gates. You may use different widths for the power busses internal to your cell or internal to individual gates.
- You may not use Metal4 or higher-level metals anywhere.