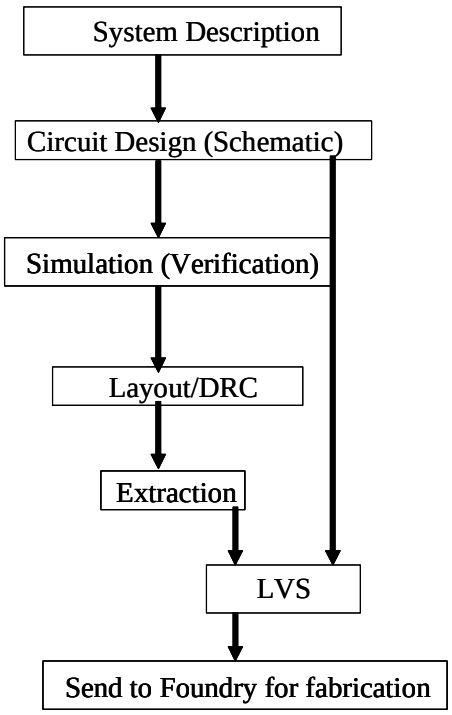
Laboratory 4: From Boolean equation to Silicon

Objective: The objective of this experiment is to implement a Boolean function description in silicon, given area and pin constraints. The concept of parameterized cells (pcells) is introduced for quick and less error-prone layout.

Introduction

A simplified Custom IC design flow is shown in the flow chart. From the flow chart, we can observe that we now have the basic skills to do a complete design of a simple circuit. In this lab, we will take the system description in the form of a simple Boolean expression and convert it to a layout for fabrication through MOSIS. Before we start the design, let us first learn a layout technique that can save us considerable time. Once you have finished learning the technique in Part 1, you can start implementing the Boolean expression assigned to you.



Part 1: Layout of an inverter using pcells:

From the previous lab on layout basics, we can see that it takes considerable time to layout even a simple circuit such as an inverter and the process is prone to errors. One way to expedite the layout process is to use the parameterized cells (pcells). Pcells are "macros" that take numeric arguments and generate a layout accordingly. For example, you can specify the length and the width for an nmos pcell to create the layout of an NMOS transistor (you still need to create the bulk connection!). To look at all the available pcells for our technology, look in the category "layout macros" of the **NCSU_TechLib_ami06 library**.

To practice, backup your **inverter** cell by copying it to a new cell called **inverter_old**. Delete the **layout** view of the **inverter** and re-create it. In the new layout window, click on **Create→Instance** and select **NCSU_TechLib_ami06**, in the Library entry, and then choose **nmos** in the Cell entry. If you now click in the layout window you can place an instance of the **nmos** pcell. Hit Shift-F to view the pcell in its layered form. If you need to modify the properties of this instance, select the instance, bring up the "properties" form, and edit the width and length to desired values. Note that using pcells will only produce the transistors or contacts; you still have to draw the required interconnects.

Complete the inverter layout using pcells and make sure that you pass LVS.

Experiment with the fields "**Fingers**" and "**Multiplier**" in the properties of the transistor pcells. Can you use the resulting layouts in making your gate designs more compact?

Part 2.1: Layout of a logic function:

Ask the lab instructor for the Boolean expression that you will be implementing. The expression is of the form $Y = f(A,B,C,D)$. You are allowed to use NAND, NOR, and Inverter gates only. The NAND and the NOR gates can have at most 3 inputs. Rewrite the expression so that you know which gates you will be using. Each gate must go through the complete design flow, i.e., schematic, layout, and LVS. Refer to your textbook if you are unsure about the schematic of a gate. All transistors in the gate design can be minimum-sized.

Exchanging gates

You are allowed to visit with other students around you and exchange the gates to speed up the process. Share a gate with a colleague only after you have simulated, performed layout without DRC violations, and have passed LVS. You may exchange gates using the following steps. These steps are used to "pack" an **inverter** cell in **lablib** library in a single file that can then be given to a colleague..

- From command line, cd into the **lablib** directory
  ```
  cd ~/ee434/lablib
  ```
- Create an archive of the inverter cell
  ```
  tar cvf ~/inverter.tar inverter
  ```

The file **inverter.tar** in the home directory contains all the views required to use the inverter cell. If you receive this file from a colleague, use the following steps to enable the use of **inverter** in your design.

- To keep the external designs separate, create a new library called **exchange** (or some other name) and attach it to the AMI06 process.
- If you have the inverter.tar file in your home directory, move it to the **exchange** directory
  ```
  mv ~/inverter.tar ~/ee434/exchange
  ```
- Extract the design
  ```
  cd ~/ee434/exchange
  tar xvf inverter.tar
  ```

The inverter cell will now appear in the **exchange** library. Instantiate the symbol and the **layout** views from this cell to use the inverter in your own design.

Part 2.2: Chip layout planning:

After we have our design in layout view, we need to place it in a "pad frame" and send it to be fabricated. The complete chip has a usable area of $900\mu m \times 900\mu m$. We will share the available area among all the students in the class and combine the designs onto one chip. The complete layout of the expression from each student should be as small and compact as possible. You have to fit your design in an area no more than $150\mu m \times 150\mu m$.

To make your design routable, layout of your logic function should follow the scheme of Fig. 1. Use VDD on top, VSS at the bottom, the inputs to the left, and the output to the right. This will make the final routing of all the designs easy since we will have signal and power paths running vertically and horizontally between the designs.

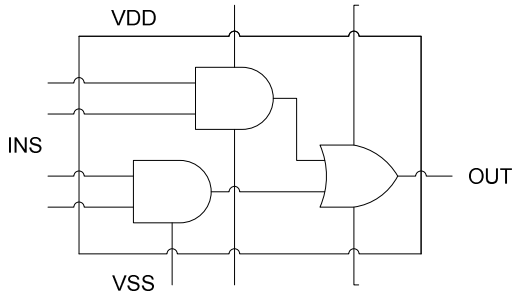


Figure 1: Cell input, output and power directions

It is a good idea to fix the height of every gate to be the same. Similarly, fix the widths of the power rails of an individual gate before starting layout. If you and your colleague agree on a set of specifications before starting layout, you could just abut the gates and avoid routing power and ground rails.

Design Constraints

Following is a summary of the constraints you must abide by when doing the layout.

- You may only use NAND, NOR or inverters.
- A gate can have a maximum of three inputs.
- The completed cell should have dimensions of $150\mu m \times 150\mu m$.
- The inputs (A, B, C, D) for the complete cell must be available at the left boundary of the cell.
- The output Y of the completed cell must be available at the right boundary of the cell.
- Use a 6µ wide metal1 bus at the top of the complete cell for VDD and a 6µ wide metal1 bus at the bottom of the complete cell for ground. Use these rails to supply power to your gates. You may use different widths for the power supplies inside individual gates.
- Avoid using metal2 inside layout of individual gates.
- You may not use metal3 anywhere.

♦ This document is available online at http://class.ece.iastate.edu/ee434/labs/lab4.pdf