EE103 Intro to VLSI Design

Fall 2024

Project: CMOS Combinational Design and Optimization

Please read the assignment carefully!

Objective:

- You will design a CMOS combinational circuit, implement by using Synopsys Custom Compiler, and use HSPICE simulation tool to analyze and optimize it.
- The DUE DATE for the project report is Dec. 6, 2024.

Lab Tasks:

- Design a CMOS combinational circuit given the functionality and optimize the delay.

Assignment:

- You are required to design a binary counter and **minimize the number of transistors**. The circuit detail is given below:
    1. 3 inputs (A, B, C), each is a 1-bit signal
    2. 2 outputs (N1, N0) – used to count the number of signal 1 for inputs.
    3. Functionality:
        - If no 1 in ABC: N1=0; N0=0;
        - If one 1 in ABC: N1=0; N0=1;
        - If two 1's in ABC: N1=1; N0=0;
        - If three 1's in ABC: N1=1; N0=1;
- Implement your circuit by Synopsys Custom Compiler. Set the length of all transistors to 100nm and assign some arbitrary numbers to the widths. Generate the SPICE netlist.
- Use HSPICE to verify the functionality is correct. Similar to previous lab where you instantiate an inverter, now you need to instantiate your cell in an HSPICE script.
- For HSPICE optimization, you need to use what you have learned from the lectures and previous labs and **minimize the worst-case delay** at your best by assigning **appropriate size (adjusting the width)** for each transistor. We only consider the case that **only one input is switching.**
- **Undergraduate section: It is not required to consider inserting additional buffers/inverters for optimizing the delay.**
- **Graduate section: Graduate students should add buffers/inverters if needed to optimize the delay.**
- The student in each section who achieves the fastest delay will get 2/100 extra credits, and the second will get 1/100 extra credit.

Additional requirements:

1. You design should be a CMOS circuit.
2. Width for all transistors should be in the range of [300nm, 10000nm].
3. You first need to add a buffer to each primary input, which should consist of two inverters in series. The length and width of the first inverter in the buffer should be the same as the **non-skewed or balanced inverter** (tphl = tplh), which we have obtained from HSPICE optimization in the previous lab. In this case, the delay of your circuit should be the delay from the output node of the input buffer to the outputs of your circuit.
4. Add a load which is equivalent to 128 times of the gate capacitance of the **non-skewed inverter** (the first inverter of the buffer you added to your primary inputs) for each primary output (i.e., H = 128 for each path).
5. The goal of the first stage is to minimize the number of transistors. You need to count the transistors in the input buffers. While the first inverter in the input buffer should be fixed, it is possible to integrate the second inverter with the required logic to minimize the number of transistors. You do not need to count the buffers/inverters you will later insert for reducing the delay (for grad students). Then, you will fix the design and optimize for the delay. Feel free to add additional buffers/inverters in this stage.
6. Each gate in your circuit should be non-skewed, i.e., the fall delay and rise delay should be approximately the same. Note the gate is not defined by the name or the functionality. If one signal propagates from the drain of one transistor to the gate of another transistor, then you have the output of ONE gate to connect to one input of ANOTHER gate. For example, AND gate is considered as two gates: NAND + Inverter.

Submission:

- The HSPICE code files.
- A lab report include:
  1. Screenshot of your schematic form SYSNOPSYS Custom Compiler. Include the names for wires and MOSFETs that corresponds to your HSPCIE netlist.
  2. A table with all the transistors' names and their widths.
  3. HSPICE netlist of your cell (subckt).
  4. Screenshots for verifying the functionality of your circuit.
  5. Steps and techniques used in improving your worst-case delay. Show effort from both manual analysis and optimization tool. You need to clearly state which path and which input transition correspond to the worst-case delay.
  6. Screenshots showing the performance of your circuit (worst-case delay).
  7. Conclusion and insights from the project.

Grading:

The grading will be mainly based on three aspects:

1. The quality of the lab report.
2. Number of transistors used (additional buffers for optimizing the delay will not be counted here). Grading will be heavily dependent on the circuit performance. For example, assuming the optimal number of transistors is K, if your design has 2K transistors, you might only get 50% of the grades.
3. Worst case delay performance. If your delay is twice as large as the minimum delay, you might only get 50% of the grades.
4. In order to receive extra credits, you will have to measure the correct worst-case input transition, and the number of transistors should be (close to) optimal.