ECE 558/658 VLSI Design -- Fall 2021
Lab 1: Design and Analysis of CMOS NAND/Multiplexer

I. General Instructions:

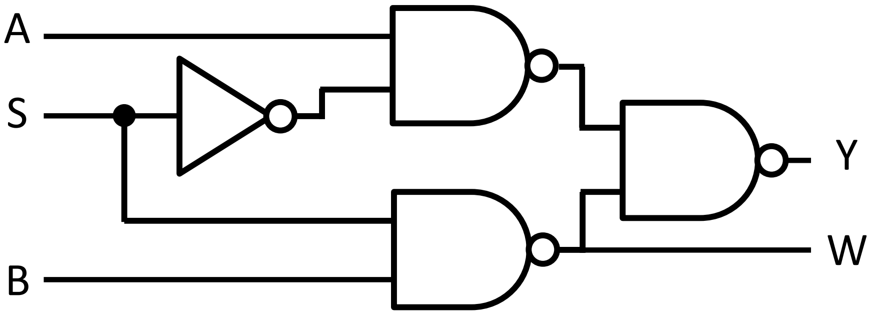
In this lab you will use industrial CAD tools to design a small circuit and run experiments to observe how circuit metrics vary with supply voltage, load capacitance, and transistor sizing. This lab assignment is a lot of work so plan ahead. We recommend that you complete Steps 1-6 soon and try out the layout editor within the first week. Be warned that creating a layout is a very time-consuming task. Your lab report should be submitted on Moodle as a pdf file with your name at the top and an indication of whether you are enrolled in ECE558 or ECE658. Some sections of the lab, as marked, are only required for students enrolled in ECE658.

You will follow a VLSI design flow for complete design, testing, and analysis of the circuit using the following tools:

- Cadence Schematic L for drawing circuit schematics
- Synopsys HSPICE for circuit level simulations
- Cadence Layout XL for creating layouts
- Mentor Graphics Calibre for Design Rule Check (DRC), Layout-versus-Schematic (LVS), and parasitics extraction.

Where to get assistance:

- The following videos go through the functionality needed for the tools. The red box at bottom left of video shows the key presses. There is no sound.
  - demo 1 (schematic and netlist creation): https://www.youtube.com/watch?v=ZWffBaY9pkM
  - demo 2 (layout): https://www.youtube.com/watch?v=tiFXe359jNg
  - demo 3 (DRC, LVS, Parasitics Extraction): https://www.youtube.com/watch?v=0esMO_DJseg
- The references listed at the end of this document include helpful resources.
- If you have any questions, please contact the TAs during office hours or post on Piazza.



II. Technical Background Information:

In this lab with you implement a simple CMOS circuit based on the schematic shown. Students taking ECE558 will implement only the NAND gate with inputs S, B and output W. Students taking ECE658 will implement the entire multiplexer, with inputs A, S, B and output Y.

Layout Requirements:

- The transistors must use minimum channel length. The technology that we use is denoted as 45nm technology node because 45nm is the effective channel length of the transistors. However, the drawn size of the channel length is L=50nm, and therefore λ=25nm.
- The layout must fit into a typical standard cell library as follows: total cell height = 128λ, with 20λ-high rails in Metal 1 for VDD and GND on the top and bottom edges of the cell, respectively. This leaves 88λ (11 tracks) of vertical space between the rails.
- Use only M1 and M2 metal layers. As a rule, M1 must be laid horizontally and M2 vertically.
- Inputs and outputs must be accessible in M1 from the left and right of the cell, respectively. Please look at a multiplexer layout example posted on the CAD website. (http://www.ecs.umass.edu/ece558/TA_Page/layout/index.html)
- You must add well ties (PTAP and NTAP) to make connections between the supply rails and the bodies of the respective MOS devices.
- Poly layer is only used for transistor gates, and there is no routing in poly. You can use a single vertical piece of poly as the gate of a PMOS and nearby NMOS; anything more complex must be routed through metal.

Assumptions:

- The nominal supply voltage is 1.1V. All specs assume this voltage.
- Use β = 2 (the P/N width ratio).
- Upsize appropriately for series transistors.
- Assume inputs that have 30 ps rise time and fall times (t_r and t_f, using 20%-80% metric from pg 141 of book)
- Assume the load of your design is a fanout of 16 minimum sized CMOS inverters (L_n=50nm, W_n=90nm, L_p=50nm, W_p=180nm, β ratio=2) and an additional 7fF lumped capacitance to represent the load from the wiring to the 16 parallel inverters. The load should be applied to either Y or W, depending on whether you are creating the multiplexer or NAND gate.

Measurement specifications:

- Spec 1: The rising and falling propagation delays must be measured using 50%-50% metric
- Spec 2: The rise time (t_r) and fall time (t_f) of the output must be measured using 20%-80% metric

III. Procedure:

Be sure to include each required item (indicated by **SUBMIT**) in your report. You must also explain what you did and why; images alone are not sufficient. The report should be organized, and all plots and annotations should be clearly readable. Incorrect or ambiguous results will result in point deductions.

Step 1: Truth Table [5 pts]

Write the complete truth table for your design, which is either a NAND or a multiplexer, depending on whether you are in ECE558 or ECE658.

SUBMIT: Truth table.

Step 2: Create Schematic [5 pts]

Step 2a: Implement a NAND gate for your design as a transistor-level schematic using the Cadence schematic editor tool (Schematic L). Use minimum sized transistors, except for the stack sizing and β ratio rules that require some transistors to be upsized. Create a symbol for your NAND gate. Capture screen images of your NAND gate symbol and its transistor-level schematic.

SUBMIT: Image of your transistor-level NAND schematic and its corresponding gate symbol

Step 2b (ECE658 only): Repeat this process to also create a NOT gate, and then create the multiplexer by assembling the NOT gate and NAND gates.

SUBMIT: Image of your transistor-level NOT schematic and its corresponding gate symbol.
SUBMIT: Gate level multiplexer schematic showing NOT and NAND gates appropriately connected.

Step 3: Check Functionality of Schematic [10 pts]

Verify that your design works correctly by extracting a SPICE netlist and using HSPICE to perform transient simulation. The SPICE netlist that you extract will not include the transistor models, so be sure to include the models as shown in the example netlist included with the lab 1 material. Your simulation should demonstrate that the circuit implements the function of the truth table. You will apply in sequence each row of the truth table to the circuit inputs, and observe the output. During this step, you might want to use the digital vector generation capability of HSPICE (**".vec"** statement) as shown in the example file we've provided with the lab 1 material. You are required to apply the inputs to the circuit in the same sequence order as used in your truth table. Use Cscope to view the result and check that it matches your truth table.

SUBMIT: Image of simulation output from Cscope, with explanation of how it demonstrates that your design implements the truth table. Annotate the graph to indicate the value (0 or 1) of each input and output signal in each step of your test sequence.

Step 4: Worst-Case Transitions [10 pts]

Simulate your design in HSPICE to verify its performance, without adding the inverters and capacitance load on the output. The propagation delays (t_{pdf} and t_{pdr}) and rise/fall times (t_t and t_f) will vary according to what input patterns are applied. Later in the course we will see how to predict which transitions are likely to be faster or slower, but for now, since you don't know what pattern will be the worst-case for delay, you must try many different patterns.

Specifically, you must test all patterns in which a transition on a single input causes a transition on the output. For example, in the NAND gate, one such case has the inputs (SB) transition from 11 → 01 (meaning that S has a falling transition while B stays high), which causes the output (W) to rise from 0 to 1, so you would need to measure t_{pdr}

and t_r from that case. When making repeated measurements, it may be easier to use SPICE measure statements, instead of relying on cscope.

SUBMIT: Create a table showing the results of all your test cases. Each row of the table should give: (1) the input combination applied (the input/output values before and after the transition), (2) the t_{pdf} or t_{pdr} that was measured, and (3) the t_f or t_r that was measured. Annotate the table to highlight the worst case for t_{pdf} , t_{pdr} , t_f , and t_r .

Step 5 Performance Analysis with Loading [10pts]

Now you will measure performance when using the load described in the performance specs. In your SPICE netlist, add the 16 parallel inverters and a 7fF lumped wire capacitance as load on the output of your design. Determine delays - $\mathbf{t_{pdf}}$**,** $\mathbf{t_{pdr}}$**,** $\mathbf{t_r}$**,** $\mathbf{t_f}$, static power (**P_{stat}**) and peak dynamic power (**P_{dyn}**) during switching. Because you are adding the loads to the SPICE netlist schematic but not to the layout, you will have to remove them from the SPICE netlist for step 7 to pass the LVS (Layout-versus-Schematic) check.

- Measure the propagation delay and rise/fall times for the following patterns:
  - ECE558: use SB = 11 → 01 to induce a rising transition on W, and 01 → 11 to induce a falling W.
  - ECE658: use ABS = 000 → 100 to induce a rising transition on Y, and 100 → 000 to induce a falling Y.
  - Remember to use the 20-80% metric for rise/fall time as required for timing spec.
- Measure the peak dynamic power when the circuit input is switching from all 0 values to all 1 values; ECE558 uses 00 → 11, and ECE658 uses 000 → 111. Explain how you've measured peak dynamic power, and where the power is consumed in the circuit. Support your explanation with plots of node voltages or currents.
- Static power results from leakage current when transistors are not switching. Measure the total static power consumed when all of the inputs are held constant at 0. Explain how you've made this measurement.

SUBMIT: Create a table with a row showing your values of t_{pdr}, t_{pdf}, t_r, t_f, P_{stat} and P_{dyn}. You will add rows to this table in Steps 6 and 8.
SUBMIT: Simulation waveforms from Cscope with t_{pdr}, t_{pdf}, t_r, t_f metrics annotated.
SUBMIT: Explain how the time measurements in step 5 compare with the measurements of the same transitions in step 4. What explains the differences?
SUBMIT: Explain how you measured static and peak dynamic power through simulation, and analysis of where dynamic power is being consumed.

Step 6: Resizing [10 pts]

Adjusting the width of transistors is a way to change the drive strength of gates, which therefore changes the performance. In this step, resize your transistors and repeat the measurements from step 5. Specifically, you must resize the width of all transistors in your design by a multiplicative factor of 2+X where X is the final digit of your student ID number. For example, if your student ID numbers ends in 7, then you will make every transistor 2+7=9 times wider than it was in step 5. Be sure to include the load on the output as in step 5.

SUBMIT: Your modified transistor sizes.
SUBMIT: Extend the table from Step 5 with a new row for measured values of t_{pdr}, t_{pdf}, t_r, t_f, P_{stat} and P_{dyn} after resizing
SUBMIT: Explain the differences between the step 6 and step 5 results, and explain how wider transistors could cause those differences. Be specific about the circuit mechanisms involved. If unsure, you may want to perform additional experiments or measurements in this step to determine how width is impacting performance.

Step 7: Layout [20 pts]

Create a layout for your design using Cadence Layout XL. Use the transistor widths from step 5. The cost of a circuit is proportional to its area, so you should try to minimize the total width of your design layout. Show the height and width of your cell(s) using the ruler tool (Tools>Create Ruler). Make sure your layout passes design rule checks and layout-vs-schematic checking.

SUBMIT: Image of your layout, with total height and width annotated. All layout requirements from page 2 of this assignment must be satisfied.
SUBMIT: Screen capture of DRC showing that your design passes. Must be clearly readable.
SUBMIT: Screen capture of LVS showing that your design passes. Must be clearly readable.

Step 8: Parasitics Extraction [10 pts]

Extract a netlist from your layout with parasitic information (choose "R+C+CC" option from Calibre). Repeat again the simulation and analysis from step 5, using the same input patterns and load added on the output, but now using the extracted netlist. Compare the new results against the results from step 5 and explain any differences that are due to the extracted parasitics. Consider how parasitics could be reduced.

SUBMIT: Extend the table created in Step 5 by adding a new row with the values of t_{pdr}, t_{pdf}, t_r, t_f, P_{stat} and P_{dyn} from simulating the netlist with extracted parasitics.
SUBMIT: Simulation waveforms with t_{pdr}, t_{pdf}, t_r, t_f metrics annotated. Make sure the simulation results look correct.
SUBMIT: Explain the differences between these results and step 5. How significant are the parasitics?
SUBMIT: Explain one possible change to your layout that would impact the parasitics. Describe whether it would be a good change or a bad change in your opinion. You don't need to implement the change, it is hypothetical.

Step 9: Characterizing your Design [20 pts]

Library cells that are available to a designer come with precharacterized timing models. These characterizations are simpler but less detailed than SPICE models, and are suitable for quickly determining the overall delay of paths through a large circuit. In this step, you use HSPICE to characterize how $\mathbf{t_{pdr}}$ of your design varies with voltage, load capacitance and input rise/fall time. Since you are performing repeated simulations while changing a few parameters, you should investigate using the **"sweep"** and **".alter"** commands in HSPICE to automate the analyses. The input transitions that you use to measure t_{pdr} should be the same patterns given in step 5. The netlist that you are characterizing should be from step 8, including the extracted parasitics.

Step 9a: Vary the supply voltage and observe how t_{pdr} changes. You can select three supply voltages: Nominal voltage, 10% higher and 10% lower than the nominal supply voltage. Make sure that the input stimulus you apply also has the appropriate voltage level for its logical high value. Make sure that you are using the same load on the output as before.

SUBMIT: A graph showing how t_{pdr} changes with supply voltage.

Step 9b: Study at nominal voltage how the delay varies with load capacitance. Remove the 16 inverters from the load and vary the value of the 7fF lumped capacitance to observe how t_{pdr} changes with load capacitance. Perform experiments to obtain at least 4 data points. Compare these results to the t_{pdr} at nominal voltage from the prior step in order to estimate the total amount of load capacitance that was resulting from the inverters and 7fF lumped cap. From this, estimate the capacitance of a single inverter load, and calculate $C_{permicron}$.

SUBMIT: A graph showing how t_{pdr} changes with load capacitance.
SUBMIT: Estimate of inverter cap and $C_{permicron}$. Show your derivation.

IV. Useful Links:

Be aware that there may be minor changes in the procedure shown in the links below due to different tool versions

- Schematic design:
  http://www.eda.ncsu.edu/wiki/Tutorial:Analog_Artist_with_HSPICE#Create_the_myInverter_Schematic

- Schematic and Symbol editor demo:
  http://www.ecs.umass.edu/ece558/TA_Page/07.html

- Netlist extraction(after schematic design) / HSPICE simulation:
  http://www.eda.ncsu.edu/wiki/Tutorial:Analog_Artist_with_HSPICE#Simulate_the_Schematic_with_HSPICE_within_Analog_Artist

- HSPICE tutorials :
  http://www.ecs.umass.edu/ece558/TA_Page/hspice/index.html

- Layout design (Layout editor) :
  http://www.eda.ncsu.edu/wiki/Tutorial:Layout_Tutorial#Create_Layout_View_of_an_Inverter

- Sample Layouts :
  http://www.ecs.umass.edu/ece558/TA_Page/layout/index.html

- Layout vs. Schematic Verification(LVS) :
  http://www.eda.ncsu.edu/wiki/Tutorial:Layout_Tutorial2

- Netlist extraction (after layout design) / HSPICE simulation :
  http://www.eda.ncsu.edu/wiki/Tutorial:Layout_Tutorial2#Extract_Parasitics

- Simulation of the netlist extracted from the PEX tool :
  http://www.ecs.umass.edu/ece558/TA_Page/hspice/hspice_sim_pex_netlist.html