55:131 Introduction to VLSI Design
Project #1 -- Fall 2009
Counter built from NAND gates, timing
Due Date: Friday October 9, 2009

Introduction

In this project we will create a transistor-level model of a 2-input NAND gate in SPICE. After measuring the NAND gate's propagation delay, we will build a D flip-flop with clock enable with 2-input NAND gates with the timing measured for the transistor model. All models will be created in SPICE.

Parasitic resistances and capacitances will be included in the NAND gate model. These parasitic elements, along with characteristics of the transistors, will cause the gates and flip flops to have non-zero propagation delays and establish setup and hold time constraints necessary for the flip-flop to work reliably. Delays and constraints (setup and hold times of the flip-flops) will be measured using SPICE.

Goals:

- Create a transistor-level model of a 2-input NAND gate (with parasitic elements)
- Test the transistor model of the NAND gate
- Measure its propagation delay
- Create a 2-input MUX using 2-input NAND gates with timing derived from the transistor model
- Test the MUX
- Design and create a master-slave flip-flop with clock enable from 2-input MUXes and NAND gates
- Test the flip-flop with clock enable
- Measure the setup, hold, and clock-to-output times for the flip-flop with clock enable
- All work is done in SPICE

Description:

The first circuit is a 2-input NAND gate constructed from nchannel MOSFETS, pchannel MOSFETS, Resistors and Capacitors as shown in figure 1. The propagation delay for this circuit should be found using SPICE. Use the default parameters for the transistors. The load should be another 2-input NAND gate.

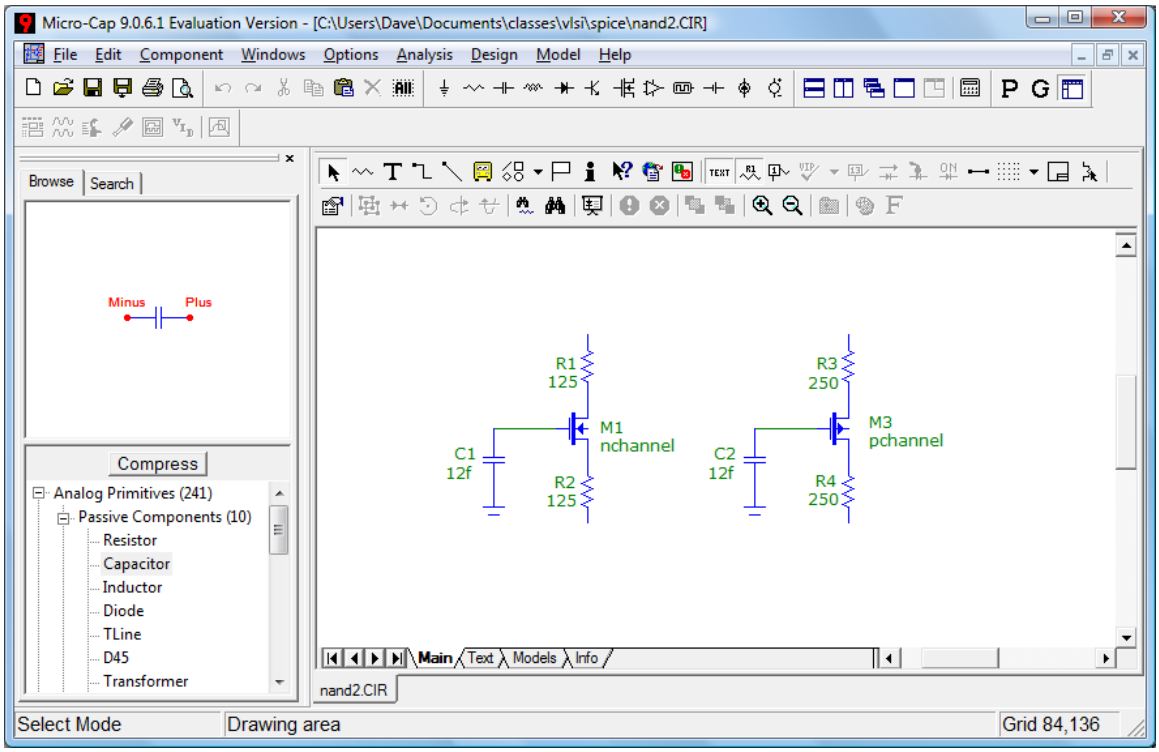

Figure 1: Transistors and parasitic elements

The second circuit to be built is a 2-input multiplexer (MUX), using 2-input NAND gates as shown in figure 2. Include the propagation delays found for the transistor model in these NAND gates.

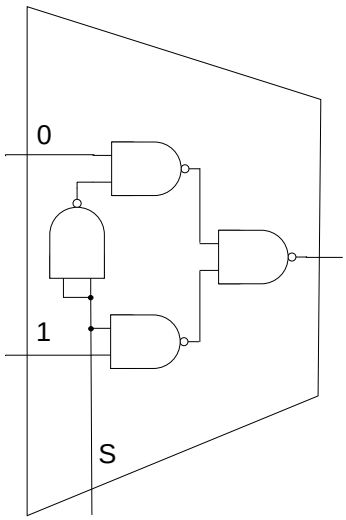

Figure 2: 2-input MUX

The third circuit is a D flip-flop with clock enable, see figure 3. The flop itself is constructed by cascading 2-input MUXes and one 2-input NAND gate (as an inverter). Small RC filters may be added to the MUX feedback paths to keep the circuit from oscillating, see figure 4.

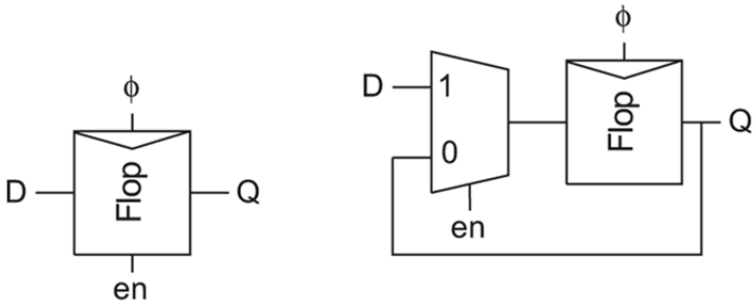

Figure 3: D Flip-flop with clock enable

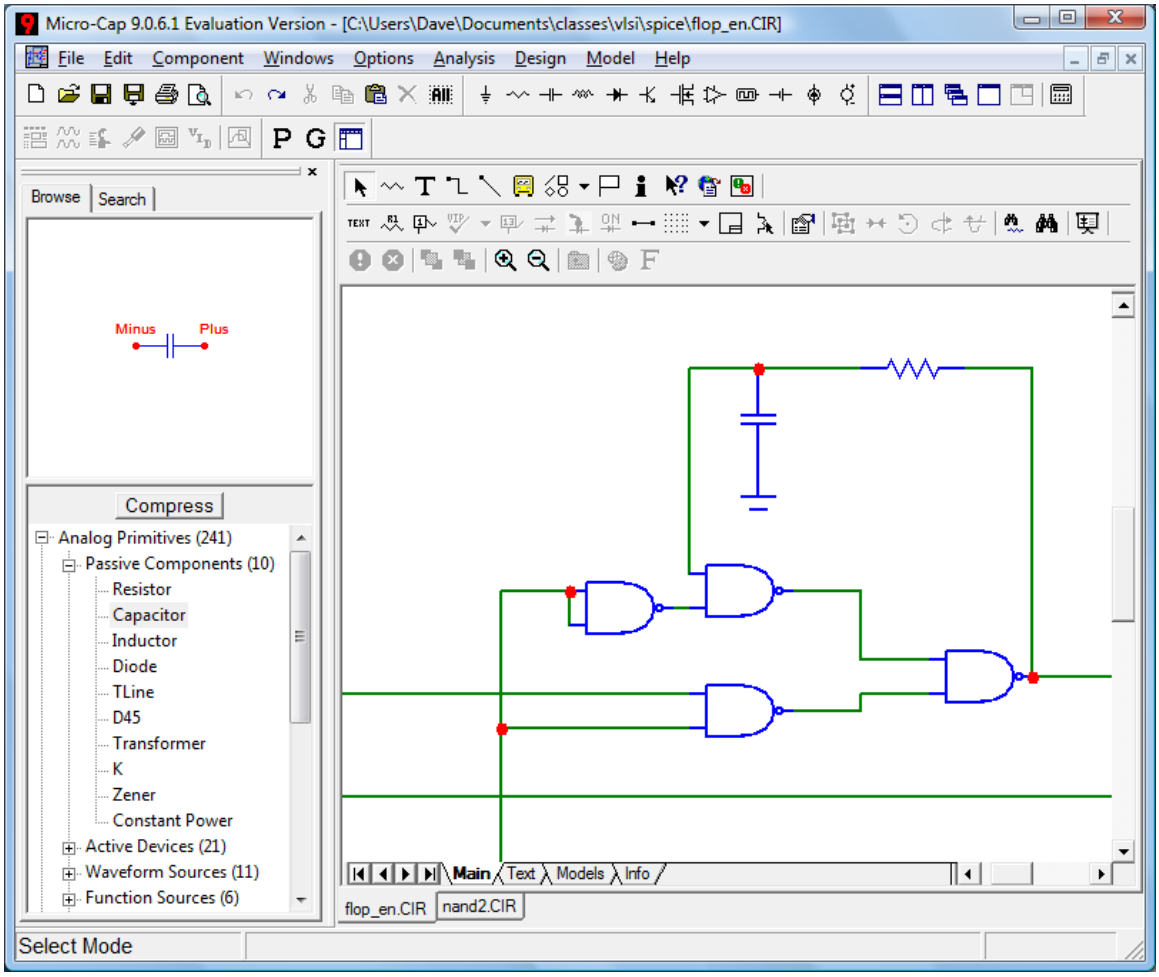

Figure 4: Feedback filter

Simulator:

Spectrum Software's Micro-Cap Evaluation version will be used for SPICE circuit simulations.

Discussion:

D flip-flops are usually a primitive design element in a VLSI device. They are inferred from behavioral code, as we'll see in future projects. Timing for primitive elements like NAND gates and flip-flops is given in device libraries. The library information comes from SPICE simulations and measurements on real silicon. We'll measure timing using SPICE in this project.

Synchronous designs are very common in VLSI devices. The flip-flops (or latches) in synchronous circuits are all driven by the same clock signal, and the clear or preset pin to each flop is driven by a common "power-on-clear" signal. Almost any circuit can be made synchronous! Logic is generally easier to design, build, and debug when it is synchronous. EDA tools are also well suited to build and analyze synchronous logic.

Questions:

1) Are the parasitic values chosen reasonable (locations and values)? Why or why not?
2) Compare the transistor count of the D flip-flop with clock enable to figure 7.19b (with the MUX of figure 7.26) from the textbook. Name at least one advantage each flip-flop has over the other.
3) What are the setup, hold, and clock-to-output times for the D flip-flop with clock enable? Are there any problems with this flip-flop?
4) What is the area of the D flip-flop with enable (using lambda rules)?

What to turn in:

Turn in your circuit files and the answers to the questions above. All designs should run in Micro-Cap SPICE. If you have any special instructions for your simulation, turn them in too. All files should be .cir or ascii text files that are placed in the ICON dropbox for this project.

There is a 10% penalty for submissions that are up to 1 week late and 50% for submissions that are up to 2 weeks late. No submissions are allowed after 2 weeks beyond the project due date.