

Lab 4: Timing Analysis of Logic Gates

Due: Friday February 22nd, 2008

Summary:

Lab 4 introduces post-layout simulation using extracted parameters from the cell layout and allows students to observe the timing and DC characteristics of some simple CMOS gates.

Learning Objectives:

1. Learn how to perform post-layout simulations for timing analysis of CMOS logic gates.
2. Observe the effects of parasitic capacitances on timing characteristics.
3. Learn how to determine the critical path of combinational circuits.

Resources: *Tutorial C*, *Guides to Writing Stimulus Files* and *Automating PWL Source Generation*

Notes:

- For all timing analysis in this lab, use an input voltage pulse rise and fall time of 50ps (rise=0.05n fall=0.05n). This is important in order to obtain consistent results.
- Keep a record of all timing measurements obtained for the NAND, NOR, and inverter. You will need them for your report and to complete Lab 6.

Procedure:

1. Complete **Tutorial C** to measure the timing and DC characteristics of the CMOS inverter you have previously designed. The parasitic capacitances from the extracted view will be used along with a load capacitance to simulate the effect of gates attached to the output of the inverter for timing analysis. Accurately measure (a) the propagation delays (high-to-low and low-to-high), (b) rise and fall times, (c) gate switching threshold (midpoint voltage), and (d) output high and low voltages for the inverter. Print/save a copy of a transient analysis waveform and a DC voltage transfer curve for your report. Include the measured results within the *Discussion Topics* of your report.
2. Using **Tutorial C** as a guide, measure the timing characteristics for the two-input NAND gate you have previously designed.
 - a. Note: In Lab 2 you should have passed LVS for the NAND (and NOR) with “Allow FET Series Permutations” turned off. This forces the order of series transistors to be the same in both schematic and layout. If you did not do this, you need to or you may get some odd results here. If you are unsure, see the end of **Tutorial B** for instructions and run LVS again.
 - b. Create a NAND stimulus file (or multiple stimulus files if you prefer) using piecewise linear (PWL) sources to generate two input signals such that ALL possible input transitions that cause an output transition are tested, e.g., 11 → 00, 10 → 11, etc. Notice there are many possible input transitions that do not cause output transitions and therefore do not need to be tested. Be sure to include the 3fF load capacitor in your stimulus file.

IMPORTANT: To obtain consistent results for delay measurements, each tested input transition MUST be preceded by setting the inputs to a common value. Here we will use ‘11’. For instance, if you want to test a 10 → 11 transition, you must generate the sequence 11 → 10 → 11.

IMPORTANT: For cases where both inputs change, these transitions MUST be setup to occur simultaneously.

- c. Simulate all output transition cases and measure the propagation delays and rise and fall times for the each output transition of the NAND. Record your results in a table similar to the following example.

Table 2c. Example of data required for Step 2.

	00 → 11	??	??		11 → 00	??	??
fall time (ns)				rise time (ns)			
propagation delay, t_{HL} (ns)				propagation delay, t_{LH} (ns)			

- d. Notice that propagation delays and rise/fall times will vary depending on the input logic state transition. Observe your results and determine which input transitions caused the worst-case (slowest) delay for each timing characteristic.
 - e. Save or print a sample output waveform that shows markers set to measure the rise and/or fall time. Save or print a sample output waveform that shows markers set to measure the high-to-low and/or low-to-high propagation delay.
3. After simulations of the NAND gate, you should be familiar with writing stimulus files for multi-input gates. To reduce some of the complexity of writing stimulus files, see the [Guide to Automating PWL Source Generation](#). This guide describes how to use a script to simplify generating piecewise linear (PWL) voltage sources in *spectre*-syntax based on specified input vectors. Identify the input transitions that cause output transitions in a NOR gate and repeat step 2 simulating the two-input CMOS NOR gate you have previously designed. Here you must precede each tested transition with the inputs '00', instead of '11'; otherwise, you may receive inconsistent timing results.
 4. After you have completed the timing analysis for the NAND and NOR gates, make sure you have the information needed to answer **Discussion Topic #2** before continuing.
 5. In steps 2 and 3 you were required to set consistent initial values before testing each transition. To analyze why this is necessary, write three separate NAND stimulus files that test the following input bit patterns: (a) 01 → 00 → 11; (b) 10 → 00 → 11; (c) 11 → 00 → 11. You can do these manually or use the automated script. Simulate your NAND gate and measure the high-to-low propagation delay of the 00 → 11 input transition for each of the three input patterns. Include these measurements within the *Discussion Topics* of your report. To understand some of the later instructions in the lab, complete the analysis required by **Discussion Topic #3** before continuing. Note: results may vary depending on your specific layout and which inputs you assign to which transistors.
 6. Using [Tutorial C](#) as a guide, measure the DC characteristics for the NAND gate.
 - a. Create a stimulus file (or multiple files) that test each possible input transition causing the output to fall from high to low. Refer to previous steps to see which input transitions cause the output to change. For some cases you can set one input high or low with a DC voltage source and do a DC sweep of the second input. For cases where both inputs change value, you can either (a) set both inputs to the same voltage source; or (b) sweep one of the input voltages and short it to the second input through a 0V DC voltage source.
 - b. Measure the gate switching threshold for all possible (output falling) transitions and record their values. Determine which transition has a unique gate switching threshold and include the plot of this case in your report. Your figure caption should identify the logic gate being simulated and input transition shown in the plot.
 - c. Consider **Discussion Topic #4** before continuing. If you wish to confirm your prediction, repeat step 6 for the NOR gate. However, this is not a required step for this lab.
 7. Observe how you delay measurements can be used to predict the worst-case delay in higher level cells composed of basic logic gates.

- a. Create the schematic for the function $F = \overline{\overline{A} \cdot \overline{B}} + C$ by instantiating basic logic gates you have designed in previous labs. This should require one INV, one NAND, and one NOR.
 - b. Create a truth table for this function and determine how many input transitions will cause output transitions (high or low). Note, there may be several transitions to consider in this three-input function.
 - c. Use the propagation delay information you measured above to calculate an estimated delay for each high-to-low output transition. You may find it useful to construct a table to do this; if you do, you can append your table to your report, even if it is handwritten.
 - d. Repeat calculations for the low-to-high output transitions.
 - e. Determine which input transitions should cause the worst-case delay for each output transition (high-to-low and low-to-high). Note that these transitions determine the “critical paths” for function F. Explain your decisions in your report under **Discussion Topic #5**.
 - f. Read **Appendix A** of this assignment and then perform transient simulations of the gate-level schematic for function F to measure propagation delays for the expected worst-case input transitions.
 - g. Test a few additional cases to verify (or disprove) your calculations have correctly identified the worst-case high-to-low and low-to-high output transitions. Measure the worst-case rise/fall times and propagation delays for these critical path cases. (Note, you will measure falling time and low-to-high propagation delay for one (output falling) case, and rise time and high-to-low propagation delay for the other (output rising) case).
 - h. Insure you have all the information you need to answer **Discussion Topic #6**.
8. Print the Lab 5 Grading Sheet and check off your lab with a TA by 5pm on Wednesday.
 9. Construct a *Brief Report* of this assignment using the **Guide to Writing Lab Reports**. Be sure to include responses to the Discussion Topics below.

Deliverables:

		<i>check off</i>	<i>lab report</i>
1	Post layout transient simulation graph and DC transfer curve for the inverter	X	X
2	Table/list of timing and DC characteristics for the inverter		X
3	Post layout transient simulation graph for NAND and NOR gates (worst-case transition only).	X	X
4	Table/list of all measured timing characteristics for NAND and NOR, and DC characteristics and Step 5 delays for NAND gate.		X
5	DC transfer curves for NAND gate (unique case only)	X	X
6	Schematic, truth table, and timing simulation results for function F	X	X
7	Calculations, discussion/explanation of the critical path delays in function F		X

Discussion Questions

Include type-written responses to the following discussion topics in your brief report.

1. Report the recorded timing and DC characteristics for the INV, NAND, and NOR gates. Create tables or other formatted lists to present your data in an organized manner.
 - Step 1: INV rise time, fall time, t_{HL} , t_{LH} , V_{OH} , V_{OL} , V_M .
 - Step 2: Information shown in example Table 2.c for NAND gate.
 - Step 3: Information shown in example Table 2.c for NOR gate.

- *Step 5: High-to-low propagation delays for 3 required cases.*
 - *Step 6: Gate switching thresholds for each falling output case of the NAND gate.*
 - *Step 7: Truth table for function F. Expected worst-case transitions and associated expected delay. Function F simulated worst case rise time, fall time, t_{HL} , t_{LH} .*
2. *Consider the worst case rise and fall times for the NAND and NOR cells. Study the relevant input transitions and try to explain why one transition is worse than all others. Give a thorough analysis of the worst-case transitions for each of the two logic gates, i.e., explain why these are the worst cases in terms of physical characteristics of the circuits.*
 3. *Refer to the three 00 → 11 high-to-low propagation delays you measured for the different input bit patterns for the NAND gate in step 2. Compare each of the three cases by explaining why (in terms of device/circuit characteristics) the observed delay is nearly the same as or significantly different from the other two cases.*
 4. *For the DC analysis of the NAND gate, determine which transition has a unique gate switching threshold and explain why this occurs (in terms of physical characteristics of the circuit). Based on this analysis, predict which input transition will generate the unique gate switching threshold for the NOR gate and briefly explain your decision.*
 5. *Define the term “critical path”. For function F, trace the critical paths in your schematic (a mental exercise) and confirm this makes sense intuitively. Specify through which logic gates the critical paths run for a rising output and a falling output.*
 6. *How well does the critical path delay of circuit F this match the calculated delays? Why is there a difference? Think in terms of any simplifications or approximations that may have been made in your calculations.*

Appendix A: How to simulate from the “analog extracted” view

For the transient simulation of function F, we want to use the extracted view of each gate that is instantiated into the higher level schematic. When you attempt this, Cadence mistranslates the extracted netlists and causes simulation problems. To get around this, you need to create an “analog extracted” view and then simulate using these views (which are different than the “extracted” views).

Complete the following procedure for the INV, NAND and NOR cells before simulating function F.

1. Open the extracted cell view and perform LVS. Make sure the netlists match.
2. In the LVS dialog box, press “Build Analog”.
3. In the box called “Build Analog Extracted View” that pops up, make sure “Include All” is checked for the Extracted Parasitics option. Click OK. This will create a new view for the cell called “analog_extracted”, which you can now see in (and open from) the Library manager.

Once this is completed for the three cells, launch the Analog Environment from the schematic for F. Follow the normal simulation procedure as described in Tutorial C, except when you get to the Switch View box under Environment options you must type “analog_extracted” before “schematic” INSTEAD OF “extracted”. Everything else is the same.

Keep in mind, this procedure is only necessary when you want to simulate a *schematic* that (a) instantiates other cells and (b) you want to include each instantiated cells’ extracted layout parasitics in a timing simulation. In other words, you don’t need to do this if you are simulating directly from an extracted view, as in Tutorial C, or you don’t wish to include extracted parasitic capacitances at all.