# EEC 116 Fall 2011 Lab #3:
# Digital Simulation Tutorial

Dept. of Electrical and Computer Engineering
University of California, Davis

Issued: October 10, 2011
Due: October 19, 2011, 4PM

**Reading:** Rabaey Insert G, Chapter 11 [1].
**Reference:** Kang and Leblebici Chapter 7.3-4 [2], Brunvand Chapters 3-5 [3].

## OBJECTIVE

The objective of this lab is to create a multi-bit adder, a test bench schematic, and to verify your adder's functionality using a digital circuit simulator.

## TOOL SETUP

No additional setup should be required for this lab.

## GATE-LEVEL FULL ADDER SCHEMATIC

In this section of the lab, you will construct two versions of a full adder using basic logic gates. You will create a multibit ripple carry adder using each of the full adder cells and verify both of them using the same testbench through Verilog simulation.

**Part 1 Full Adder (First Version)** Create a new schematic cell view for a cell named `FullAdder` in your EEC116 Cadence library. Create a schematic identical to the one shown in Figure 1. Use two-input logic gates from the `UCD_Digital_Parts` library. Note that there are two possible symbol views (`Symbol` and `SymbolA`) corresponding to the cells. It is good practice to create a more readable schematic by using the alternative symbol views (`SymbolA`), for example when you are implementing a Boolean Sum-of-Products expression using only NAND gates. Add pins and meaningful wire labels. Check and save your schematic.

Next, create a symbol view for the `FullAdder` cell. You can do this easily by choosing Create→Cellview→From Cellview... from the schematic editor window menu. A popup should appear with the fields filled in automatically. Make sure the Edit Options button is

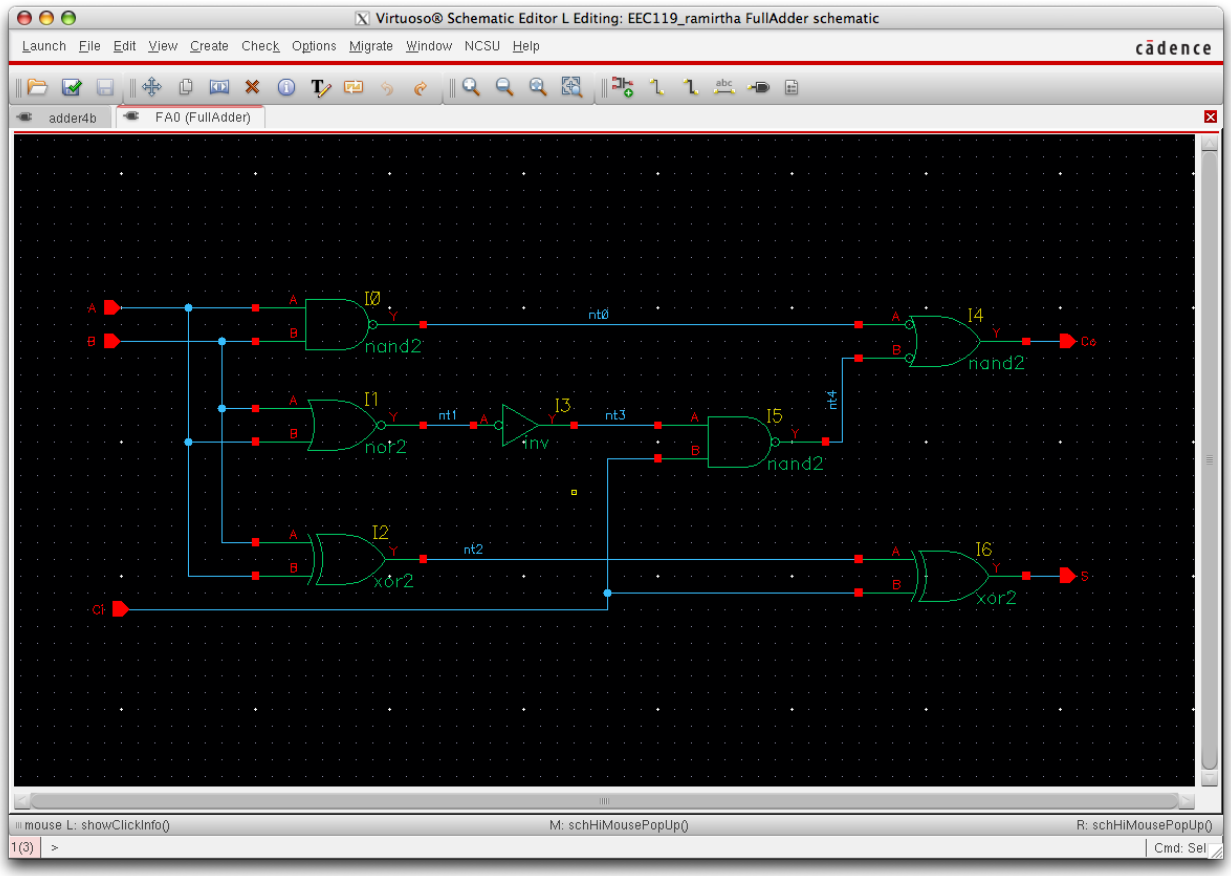


Figure 1: Full adder (first version) schematic.

checked and the Tool / Data Type pulldown menu is set to `symbol`. Click OK and another popup should appear. Edit the Pin Specifications fields so that Carry In (`Ci`) and Carry Out (`Co`) are in the Top Pins and Bottom Pins fields respectively. Click OK twice. If a window appears saying something about CDF parameters, click OK on that as well. A new symbol editor window should come up which shows the new symbol you created with pins in the locations you specified. Check and save your symbol view and close the window.

Proceed by creating a four bit ripple-carry adder schematic named `adder4b` by placing instances of your new `FullAdder` cell such that it looks like Figure 2. This schematic makes use of *vector* names for pins and wire labels so that entire buses can be represented very concisely. Use Create→Wire (wide) to draw the fat lines which represent buses. Be sure to label things consistently to avoid errors down the line. Check and save your adder schematic.

# ADDER VERILOG-XL SIMULATION

**Part 2 Ripple-Carry Adder (First Version) Simulation** At this point, we are nearly ready to simulate the four bit adder. Invoke Launch→Verilog-XL from the schematic editor and fill in the Run Directory field with

`/project/<username>/simulation/adder4b.run1`

to ensure that simulation files are written to the correct directory. Click OK and the **Verilog Environment** window should pop up. `cd` the the simulation run directory and take

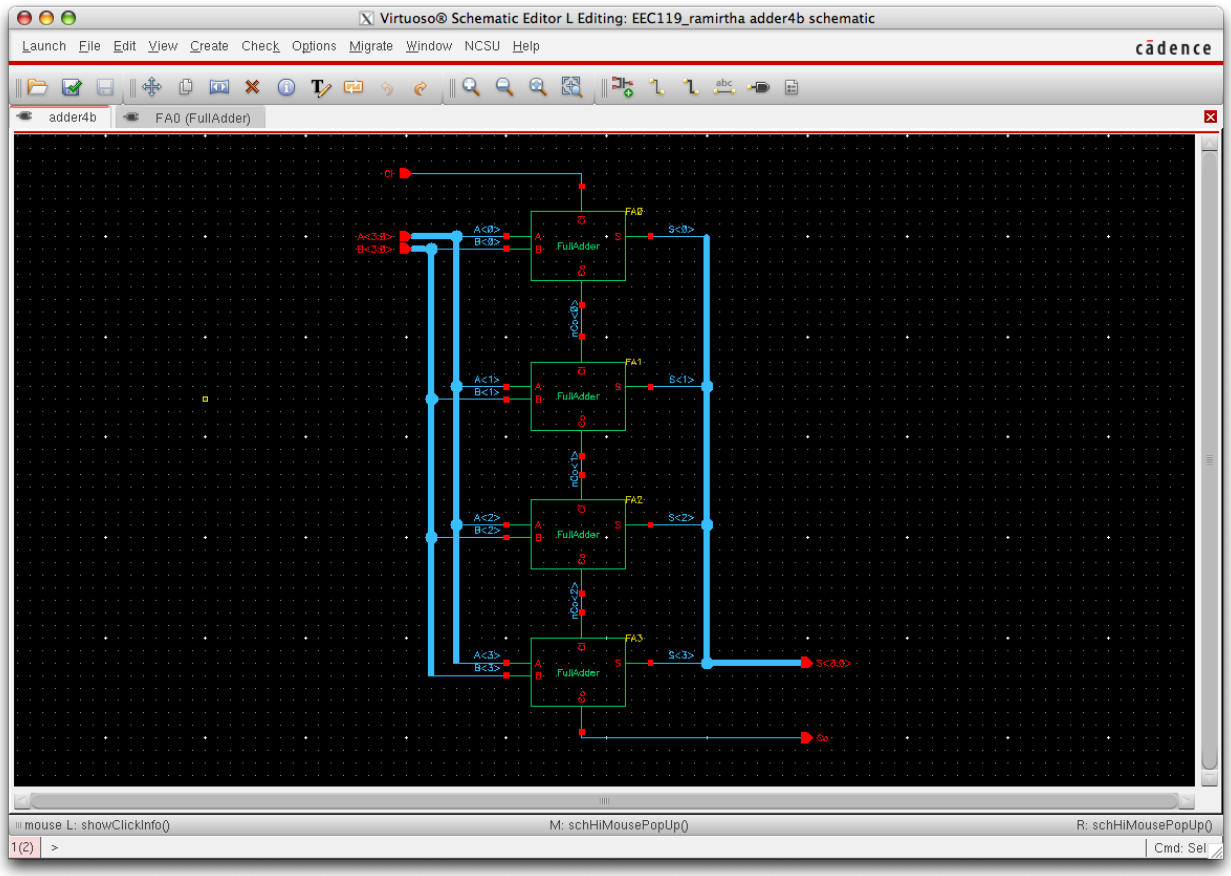


Figure 2: Four bit ripple-carry adder (first version) schematic.

a look around to get a sense of the various files that Verilog-XL will use to simulate your circuit. Next, click Setup→Netlist... and select More. Ensure that the Stop Netlisting at Views field is set to `behavioral functional` only. Then click Setup→Record Signals... and make certain that All Signals is selected and Save is checked. The next step is to Invoke Stimulus→Verilog... to edit a new testfixture. Select `testfixture.verilog` and Copy. In the new window, choose `testfixture.verilog` and copy it to File Name `testfixture.new`. Verify that Design Instance Path is set to `test.top` and check the Make Current Testfixture and Check Verilog Syntax boxes. Click OK. Go to Stimulus→Verilog... again and select `testfixture.new` for editing. The default text editor window should appear with some initialization code already included in an `initial` block. Edit the file until it looks like the nested for loop testfixture code from lecture (or you can download the `testfixtureLoop.verilog` file from the course SmartSite and cut and paste the code). Close the editor window and correct any Verilog syntax errors which may be flagged by the tool. Invoke Simulation→Start Interactive from the menu bar or press the icon in the toolbar that looks like a "play/stop" button for a CD player. A popup may appear warning you about renetlisting the design. Click OK. A number of messages should appear in the CIW and the Verilog Environment window as various source files are compiled. Eventually, you should see a `C1 >` prompt in the Verilog Environment window. Invoke Simulation→Continue or click the "play" toolbar button and you should see results of the `$display` tasks in your

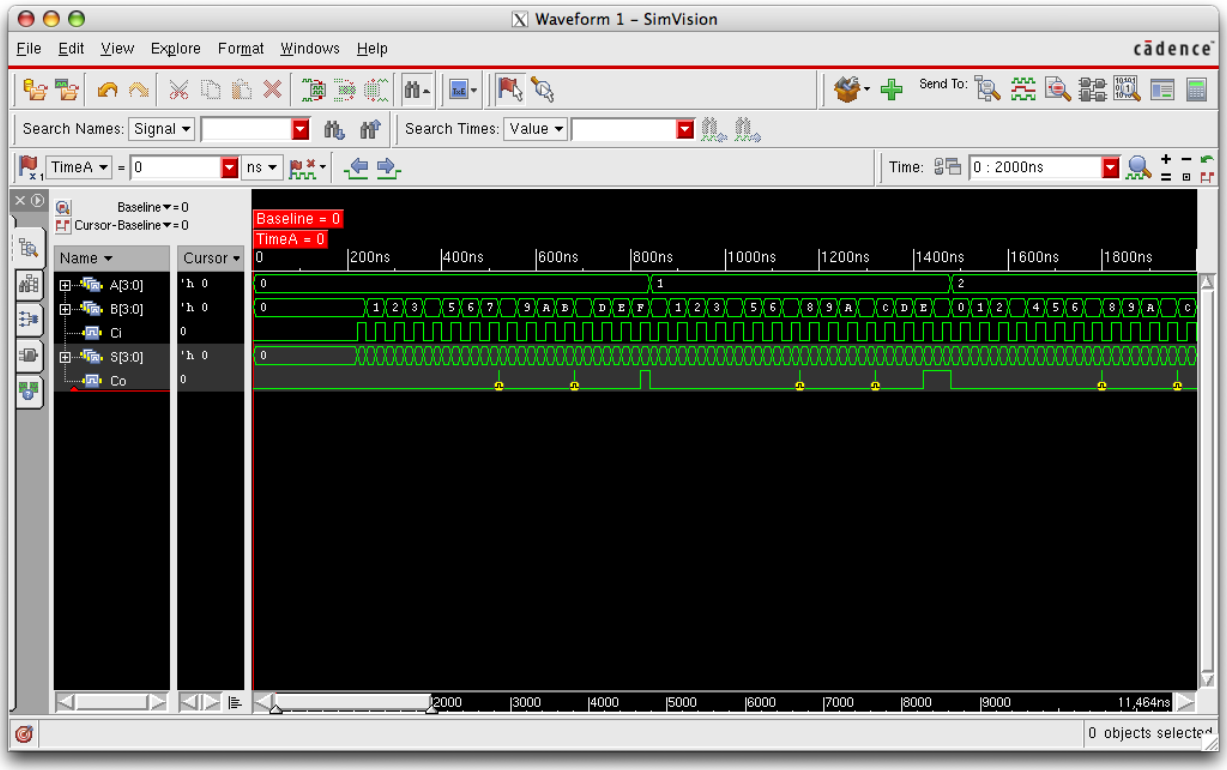


Figure 3: Four bit ripple-carry adder simulation waveforms.

testbench. If any errors appear in the adder output, fix the bugs and re-run the simulation until there are none.

It is often very useful to view the signals in Verilog simulations as waveforms. Go to Debug→Utilities→View Waveform... to launch the **SimVision** waveform viewer. Click on the Design Browser tabs running vertically down the left hand side of the window. It should open up a browser which looks like a directory browser but whose levels correspond to modules in the design. Descend to the `top` instance and click on the input and output signals of your four bit adder. You should see a window similar to Figure 3. Print the waveform plot to show the TA for checkoff and to hand in with your report.

**Part 3 Ripple-Carry Adder (Second Version) Schematic and Simulation** Copy the `FullAdder` cell you created in Part 1 to a new cell named `faddG_1x`. In the schematic, replace the NAND, NOR, and INV gates from the `UCD_Digital_Parts` library with the instances of the gates you designed in Lab 1. You can leave the XOR gates from the `UCD_Digital_Parts` library for now. Next, create a second four bit adder cell named `adderA4b` which instantiates four copies of `faddG_1x` in a ripple-carry configuration. Verify using an identical Verilog testbench to the one in Part 2 that the adder works correctly. Print out a waveform plot for checkoff and to hand in.

# MIRROR ADDER

In this section, you will create a third ripple-carry adder based on using two types of mirror adder full adder cells (the odd and even cells from Rabaey et al. [1]).

**Part 4 Mirror Adder Cells** Create **two** new cells in your library called `faddMe_1x` `faddMo_1x` and implement the mirror adder cell shown in Figure 11-6 in Rabaey with the appropriate input and output polarities for the two versions (see Figure 11-5). Choose transistor sizes consistent with the PMOS/NMOS width ratios we have been using in the simple logic gates designed in the previous lab. Lay out the cell using the same height as the inverter, NAND, and NOR cells you designed. Verify your layouts are free of DRC and LVS errors. Be prepared to show your schematics, layouts, and demonstrate that they are free of errors to the TAs for checkoff.

**Part 5 Ripple-Carry Adder (Third Version) Schematic and Simulation** Create a third four bit adder cell named `adderM4b` which is composed of `faddMe_1x` and `faddMo_1x` cells in a ripple-carry configuration. You will need to add inverters at appropriate inputs and outputs to ensure that the correct polarities are maintained. Verify using an identical Verilog testbench to the one in Part 2 that the adder works correctly. Be prepared to show your schematic to the TA for checkoff and print out a waveform plot for checkoff and to hand in.

# Checkoff

Show your completed schematics, layouts, and all waveform plots to the TA for checkoff.

# Report

You must hand in a **typewritten** report to receive credit for this lab. Your report can be brief, but must include the following sections in addition to the completed summary sheet attached at the end of this lab. The summary sheet will be the cover page of your lab.

1. Overview: Describe in one paragraph the objectives of the lab. State what you were testing and what data you expected to gather as a result of your simulations.

2. Procedure: Briefly document your methodology for verifying the correct operation of the digital circuits. Hand in your waveform plots for the three ripple-carry adder variants confirming correct operation.

3. Design: Hand in your schematics for `faddG_1x`, `faddMe_1x`, and `faddMo_1x`. Describe how you chose the transistor sizes for the mirror adder cells. Also, document any choices you made in the physical design (layout) of the adder cells, for example if you shared particular source/drain diffusion areas for a specific reason, etc.

4. Discussion: For the ripple-carry adders `adderA4b` and `adderM4b`, identify the critical path delay which would limit the adder's frequency of operation. Which implementation is likely to be faster and why?

# Acknowledgments

Parts of this lab were inspired by lab exercises developed by Prof. David Money Harris and others at Harvey Mudd College for the class E158: Introduction to CMOS VLSI Design.

# References

[1] J. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits: A Design Perspective*, 2nd ed. Upper Saddle River, New Jersey: Prentice-Hall, Inc., 2003.

[2] S.-M. Kang and Y. Leblebici, *CMOS Digital Integrated Circuits: Analysis and Design*, 3rd ed. San Francisco: McGraw-Hill, Inc., 2003.

[3] E. Brunvand, *Digital VLSI Chip Design with Cadence and Synopsys CAD Tools*, 1st ed. San Francisco: Addison-Wesley, Inc., 2010.

# EEC 116 Fall 2011 Lab #3 Summary

**Name:**

**Grading:**

| Part | Checkoff | TA Initials | Date |
|---|---|---|---|
| 1 `FullAdder` Schematic | | | |
| 1 `adder4b` Schematic | | | |
| 2 `adder4b` Waveform Plot | | | |
| 3 `faddG_1x` Schematic | | | |
| 3 `adder4Ab` Schematic | | | |
| 3 `adder4Ab` Waveform Plot | | | |
| 4 `faddMe_1x` Schematic | | | |
| 4 `faddMe_1x` Layout | | | |
| 4 `faddMe_1x` DRC | | | |
| 4 `faddMe_1x` LVS | | | |
| 4 `faddMo_1x` Schematic | | | |
| 4 `faddMo_1x` Layout | | | |
| 4 `faddMo_1x` DRC | | | |
| 4 `faddMo_1x` LVS | | | |
| 5 `adder4Mb` Schematic | | | |
| 5 `adder4Mb` Waveform Plot | | | |