ECE C03
LAB#1
Introduction to Mentor Graphics Tools

Spring 1998
Assigned: Tuesday, March 31, 1998
Due: Thursday, April 9, 1998
at the BEGINNING of class

1 The Assignment

Your assignment is to work through the entire *Mentor Graphics Tutorial: Design of a Full-Adder*, which you should have purchased from the ECE Department Office. This assignment should be an effective way to learn how to use the Mentor Graphics software. Note that the Tutorial refers to sections of the Mentor Graphics Reference Manual (the items by the bullets). You are not expected to read the Reference Manual from cover to cover, but be sure to read through the sections of the Reference Manual listed in the preface of the Tutorial. If you are unable to do what the Tutorial is asking, you can refer to the Reference Manual for more details.

This lab will take a **long time**. You can complete it over several sessions. Be sure to save your work each time. Part 3 of this lab assignment specifies which items must be printed and turned in for grading.

2 Hints

Before you start, read the tutorial first, then read these hints. They may not make much sense the first time you read them, but they're important, so refer back here first if you get stuck.

- Don't forget to type *ecec03* after logging in.
- Maximize windows to see more! To maximize a window, click on the button with the large square box in the upper right corner of your window.
- To arrange icons in the *Tools* window, select *View>Update Window.* (You'll be able to see them better that way.)
- Wherever the manual says *<random_chars>*, you will see or type something like *ecec03*. Look at the title of your *Navigator* window to determine the exact string.
- When typing file names **don't forget the underscore** (e.g. the file name *my_xor* is the word *my* followed by an underscore followed by the word *xor*) and distinguish between upper and lower case. UNIX is case sensitive. You can type an underscore ("_") by holding shift and typing the key to the right of the zero.
- Make sure that the *tutorial* directory gets created correctly, and that the *my_xor* and *full_adder* are created in the tutorial directory. Once the tutorial directory has been created, it exists until it is explicitly deleted. If you work on Lab #1 in more than one session you do not need to recreate the tutorial subdirectory but you do need to execute *ecec03* each time you login.
- Pop-up menus may be obtained by positioning the cursor in the window and pressing the right mouse button. In particular, after adding symbols to your schematic, you will need to open the pop-up menu in the *gen-lib* window and select *Display Schematic Palette* in order to get the original menu panel.
- Hit *F2* to unselect all components in your schematic.
- If you make a mistake simulating, select *(Run menu)>Reset values.*

- The simulation trace of your *full_adder* circuit may not look exactly like the one in the tutorial. The overall shape of the waveforms should be the same, but you may notice some short "glitches" or transient erroneous values occurring in your output waveforms at the points on the trace where one or more inputs to the circuit are changing. We will examine the cause of these glitches in Lab #3.

3 Questions

TURN IN the following items for a grade

√ A sheet plot of your *xor* circuit generated in Section 2.8 of the Tutorial.

√ A sheet plot of your *full-adder* circuit generated in Section 4.4 of the Tutorial.

√ A simulation trace of your *xor* circuit generated in Section 5.4 of the Tutorial. For this and future assignments, a plot of the trace window (not the lists window) should be turned in.

√ A simulation trace of your *full-adder* circuit generated in Section 5.7 of the Tutorial.

Each sheet plot should be labeled with your name, your TA's name, and your section number. This information should appear in the comment block in the border around your schematics.

4 Explanation

For this lab, we don't expect you to be able to design the circuits or understand why they're useful. The point of this lab is only to get you familiar with the software you'll be using to enter and simulate circuits. The following is a brief description of what these circuits do.

Look at the sheet plot of your *xor* circuit. The circuit has two input variables A and B, and one output variable F. These variables take on the binary values 0 and 1 and propagate along the wires ("nets"). You specified the input values by using the *Force>Multiple Values...* command in Quicksim**.** Your *xor* circuit behaves according to the truth table on page 14 of the Tutorial. Now look at the simulation trace for your *xor* circuit. Note that there are three lines corresponding to A, B, and F. The numbers at the bottom represent time units. Consider for example from time 20 to time 40. Over this range A is 0 and B is 1, as represented by the height of the trace line. The output F is seen to be 1 as expected by the truth table for *xor*. Compare this truth table and the *Force>Multiple Values...* and *Run>Simulation>For Time:* commands you used in Quicksim with your simulation trace over the periods 0 to 20, 40 to 60, and 60 to 80 to understand what the simulation traces mean.

The *full-adder* (FA) circuit will be explained more fully in class and revisited in Lab #2. Page 17 of the Tutorial provides a truth table showing the behavior of a *full-adder* circuit. Basically, a FA adds two binary inputs A and B and a carry-in and produces the sum S and a carry-out. Consider the following ordinary addition of the two binary numbers 1001 and 0011:

$$
\begin{array}{rcc|c|c}
 & {\scriptstyle 0} & {\scriptstyle 1} & {\scriptstyle 1} & \\
 & 1 & 0 & 0 & 1 \\
+ & 0 & 0 & 1 & 1 \\ \hline
 & 1 & 1 & 0 & 0
\end{array}
$$

Each FA circuit is used to add the bits in a single position (column). For example, one FA would correspond to the position surrounded by the box: the A and B values are 0 and 1, the carry into that position (Cin) is 1, and the outputs are S=0 and Cout = 1.