ECE C03
LAB #2
Design of a 4-bit Adder with Delays

Spring 1998
Assigned: Thursday, April 9, 1998
Due: Tuesday, April 28, 1998
at the BEGINNING of class

Helpful Hint: Good organization is important in these labs. It is recommended that for each lab you create a new directory (called folders inside Mentor Graphics) and put all of your designs for the corresponding lab in that directory. For instance, you should create a new directory for this lab called something like "lab2". You should create this directory in your <ECEC03 WORKING DIRECTORY>. (Your <ECEC03 WORKING DIRECTORY> is the directory you are placed in after executing the ecec03 script.) You can use the UNIX command *mkdir* to create a "lab2" directory by typing "mkdir lab2" at the *ecec03>* prompt. Alternatively, you can make a directory inside Mentor Graphics *Design Manager* by using the *Add>Directory* command. You should put both your Full Adder and Ripple Adder designs for this lab in this directory.

1 A Full Adder Cell

1.1 Circuit Diagram

Shown below is a diagram of a full adder cell. Note that the circuit is not the same as the full adder you constructed in the *Mentor Graphics Tutorial*, but its functionality is the same. Create a new directory (folder) for this lab, and use *Design Architect* to enter the full adder circuit as shown. (**Warning**: ***do not*** call the full adder circuit "*lab2*" or any other name similar to your directory name. Choose a name like "*full_adder*", etc. Also, ***do not*** include spaces in the filename-this will cause problems! Be sure to start the file and save it in your *lab2* folder.) Use components only from *gen_lib* and use borders to comment your designs as in Lab #2. Ignore for now the numbers near the gate outputs (initially zeros); these correspond to delay times and will be discussed in Section 1.2. Also, don't bother to label the gates G1-G5. These names are only given for later reference.

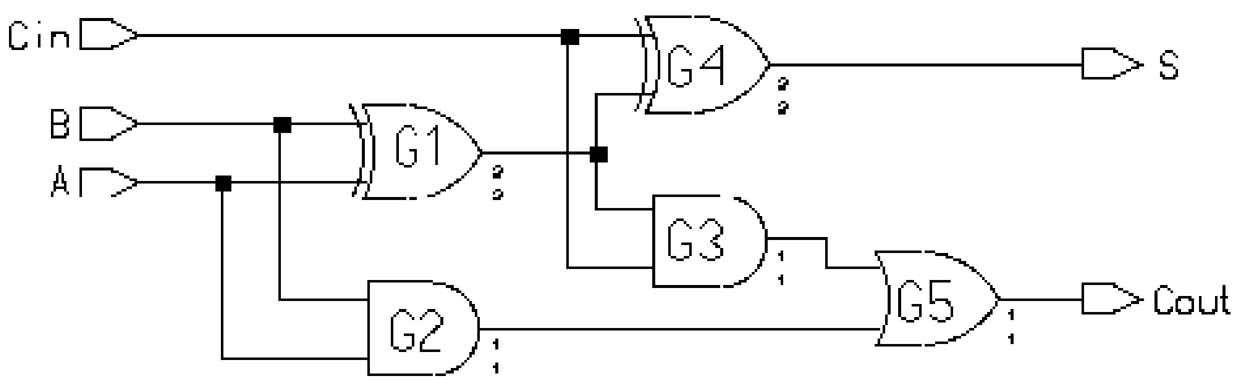

Figure 1 Full Adder Cell

1.2 Component Delays

A crucial part of this lab is the consideration of component delays. So far, we have been using ideal components with no gate delays. With an ideal gate, we neglect the time required for the output of the gate to change from low to high (rise time) or to change from high to low (fall time). However, actual gates cannot change output values instantaneously, thus imposing a limit on the number of computations per unit time a gate can perform. The numbers near the gate outputs in Figure 1 are used to simulate the rise and fall times of each gate. Change the rise and fall times of each gate by following these steps:

1. First press <F2>. Then point the mouse pointer directly at the number you want to change (delay value), and press the <F1> key. This will select the value (the number should change color to white when selected)
2. *(Popup)> Change Values:* (or hit shift-<F7>)
3. When the Change Dialog box is displayed, enter the new value in the <u>Value</u> field.
4. Click on the OK button of the dialog box. You should see the value change. Repeat for all delay values.

You will have to do this for each gate in the circuit. Set the RISE and FALL values of the *and* and *or* gates to 1. Set the RISE and FALL values of the *exclusive-or* gates to 2. You should see the two zeros near the gate outputs change to ones or twos as in Fig. 1. Save your full adder schematic (use *File>Save)*, then check it by using *Check>Schematic*. Generate a Sheet Plot of this full adder circuit with delays.

1.3 Testing and Simulation

You are required to demonstrate that your full adder circuit works. Follow these steps:

1. Enter Quicksim and set up the windows. You are required to show the traces A, B, Cin, S, and Cout in that order (do this by adding one by one in that order). Add forces to add these numbers at the times given:

time	A	B	Cin
0	0	1	0
10	0	1	1
20	1	1	0
30	0	0	1

2. Execute the item *Setup>Kernel>Analysis...* from the Quicksim menu bar.

   Change the timing mode to *Delay*, then click on the Visible button next to detail.

   Make sure that "typ" is selected for Timing mode. Now you will see the delays on your plots.
3. Initialize the simulation by doing *Run>Initialize:* and entering "0r" when prompted for a state value. (Note that this value is a ZERO followed by a lowercase "r")
4. Simulate for 40 ns.
5. Put the mouse in the Trace window and click the left mouse button to highlight the Trace window, and then execute *(Trace Popup)>Setup>Window...* Enter "4" for "Domain label interval" and "60" for "Domain pixels/interval" and press the OK button. (The Quicksim

window needs to be large enough so that the entire 40ns simulation fits on the screen) Plot the trace of the simulation as in Lab 1.

1.4 Full Adder Symbol

Create a symbol for the full adder using *Miscellaneous>Generate Symbol* from the menu bar of Design Architect. (Note: this method of creating a symbol is different, and easier, than the method you used in Lab#2.) Save the symbol using *File>Save Symbol*. Now check the symbol by using *Check>Symbol*. The symbol created by Design Architect will look similar to Fig. 2, although the inputs and outputs may be ordered differently. (Hint: You can move the pins to better suit your needs. First, press F2, then select both the line and diamond associated with the pin you want to move. To select the diamond, place the cursor over it and press F1. Next, choose *(Popup)>Move*, and place the pin where you want it.)

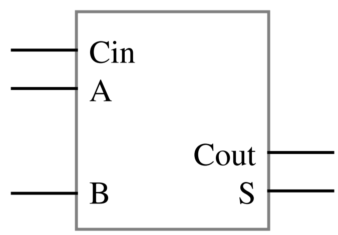

Figure 2 Full Adder Symbol

2 Four-Bit Adder

Next you will construct a *ripple adder* and simulate it.

1. Use Design Architect to build a 4-bit ripple adder by cascading together four of the full adder cells you just created as shown in Fig. 3 on the next page. Use the *Choose Symbol* button on the schematic palette to retrieve your full adder symbol. Your full adder symbol may be slightly different from the ones in the diagram, so be careful that your connections are consistent with those in Fig. 3. Generate a Sheet Plot of the complete 4-bit adder.
2. Enter Quicksim and set up the windows. Display the following signals in the order: C3, C2, C1 C0, S3, S2, S1, and S0. Remember to use *Setup>Kernel>Analysis* and *Run>Initialize* as you did in Section 1.3.
3. Generate the necessary forces for each input to have your circuit add A=-7, B=-3 at time 0 and add A=-3, B=6 at time 20. You will have to do the addition in 2's complement. Set Cin=0 for both cases. (Note that A consists of the four bits A3 A2 A1 A0, and similarly B consists of the bits B3 B2 B1 B0. To set a force for A, set forces for A3, A2, A1, and A0. These four inputs can only take the value 0 or 1.)
4. Execute the simulation command file and simulate for 40 ns. Use *(Trace Popup)>Setup>Window...* and enter the same values as in Section 1.3. Generate a plot of the simulation.

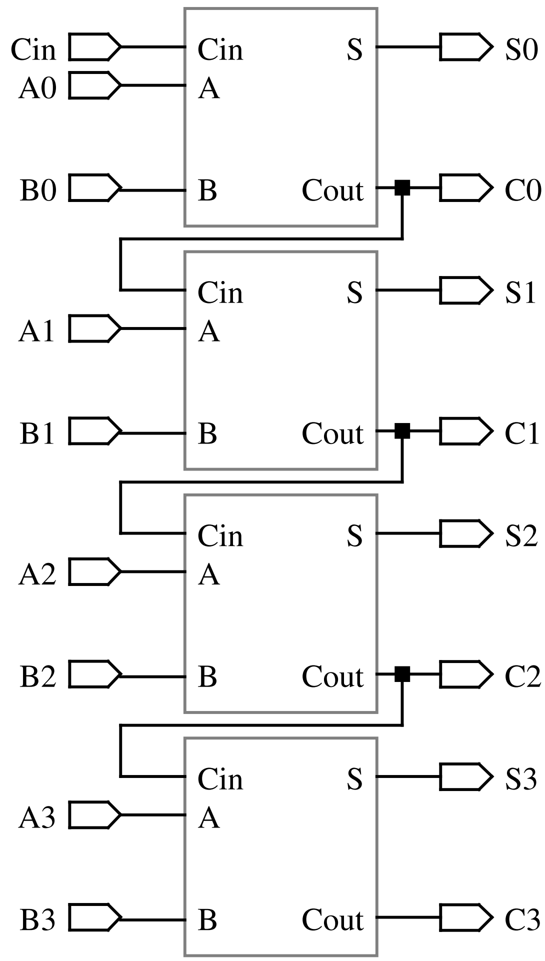

Figure 3 Four-Bit Ripple Adder

TURN IN the following items for a grade (**stapled** in this order):

√ A Sheet Plot of the single bit full adder cell and the quicksim output trace.

√ A Sheet Plot of the 4-bit adder circuit and the quicksim trace showing correct operation of the 4-bit adder.

Be sure to include your name and section number on each page.