Mentor Graphics Tutorial ece361 October 5th, 2004

Notes: Do not copy anything in dmgr using UNIX commands, it will break things. Use the copying tools in design manager to preserve the functionality of your components.

1. Copy the mentor environment from ~kcoloma/mgc.env into your own home directory.

```
squirrel:~ % mkdir MGC
squirrel:~ % cp ~kcoloma/mgc.env MGC
squirrel:~ % mkdir MGC/ece361
squirrel:~ % source MGC/mgc.env
squirrel:~ % dmgr
```

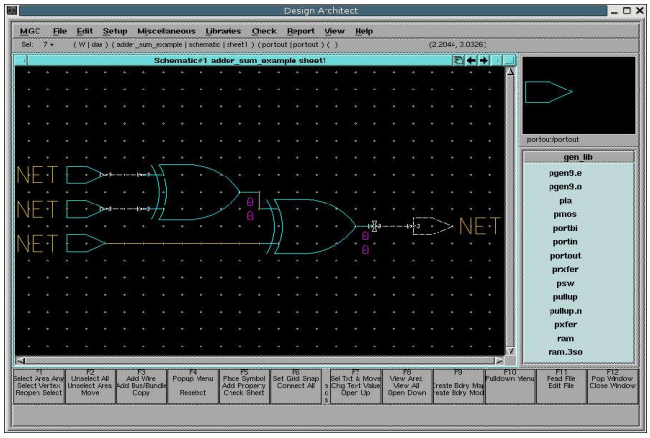
This will source the correct environment, create directory for your ece 361 files and start up design manager. There is folder named Tools on the left and a File navigator on the right.

2. We will be creating and simulating a 1-bit adder. First right click->open on the design arch tool on the Tools folder on the left. This starts up design architect. By the way, hitting ESC usually cancels any command so if you make a mistake try hitting ESC.

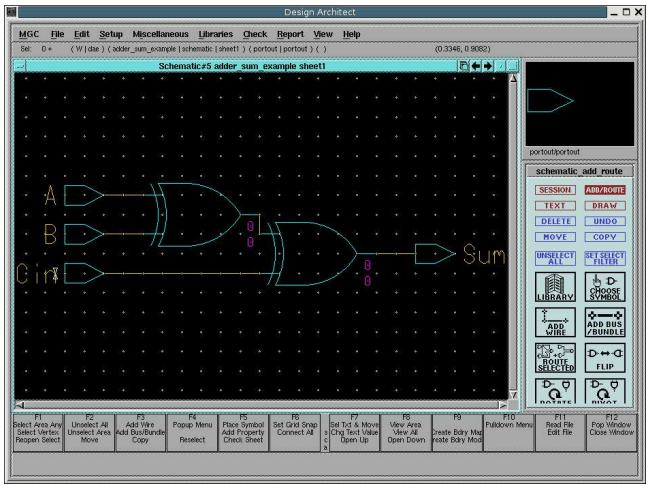
3. File->Open->Sheet and rename the Component Name to /homes/<your username>/MGC/ece361/adder_sum_example. Then hit OK. A new window will appear in design architect. Now to manipulate this window, you can either use keystrokes or mouse gestures. If you would like to use the keystrokes, they are all listed below in design architect and use F1-F12 in combination with the shift, ctrl, and alt keys. For the mouse gestures, make sure that the schematic is highlighted (you can do this by clicking its label) and then Help->On Strokes. Usually, you can also select the object by clicking on it with your mouse and then right clicking to see a menu of actions. You can use F2 to deselect everything.

4. On the right side in design architect, there is a window entitled schematic_add_route. Click on LIBRARY->gen_lib to find all the gates that are available to build your circuits. Unfortunately, the scrolling interface in design architect is quite poor in the gen_lib window, so the way to go up or down is to place your mouse cursor near the top or bottom of the window and hit the left mouse button to begin scrolling.

5. Now a 1 bit adder is simply Sum = A xor B xor Cin, Cout = Cin * (A + B) + A*B. Therefore, we use the gen_lib and the wires (F3) to construct the various circuits. In this first component, we will construct the Sum portion of the adder. We will need to use the following components from the gen_lib (portin, portout, xor2). Start by placing xor2 component and laying out wires in the correct order to use a 2 xor gates to make a 3 input xor gate. When laying out the wires, you start them by clicking once and clicking once to make turns and then double clicking to end them. Then use portin port on the left and portout on the right so that your design looks like this.



6. Now we have to rename the ports so that they make sense. Do this by first hitting F2 to unselect everything. Then mouse over the name of the first portin port 'NET' and hit F1 to select it. Then right click and choose 'changes values'. Fill in 'New Value' with 'A'. Then do the same to change the other portin ports to B and Cin. Change the portout port to Sum. When you are done you should get something like this.

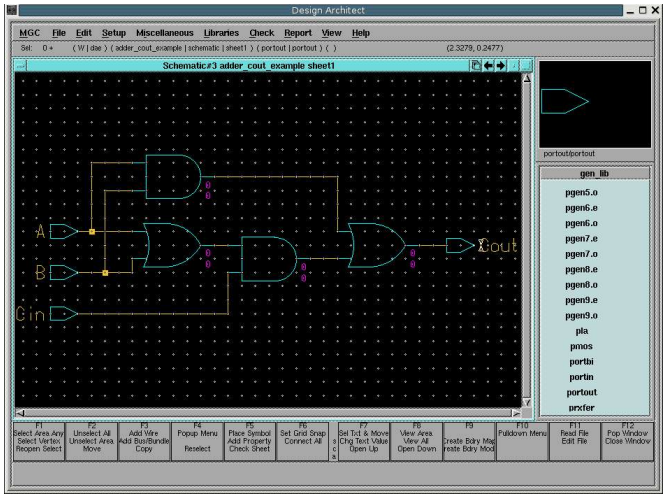


7. Now we should save our work. File->Save Sheet. We then Check->Sheet and Check->Schematic to see if we have any problems. A warning in Check Schematic is expected regarding no pins in the interface.

8. We will create a symbol that will be used in the final adder design. Miscellaneous->Generate symbol. Click Yes for 'Replace existing' and then hit OK. A symbol sheet will pop up. Check->With Defaults. It may give a warning. This is okay. Then save the symbol by File->Save Symbol. Then Check->With Defaults again and everything should be fine. Close the symbol. Check->Sheet and Check->Schematic on the schematic. Then, you can close the schematics. Congratulations you have just created a usable symbol in design architect!

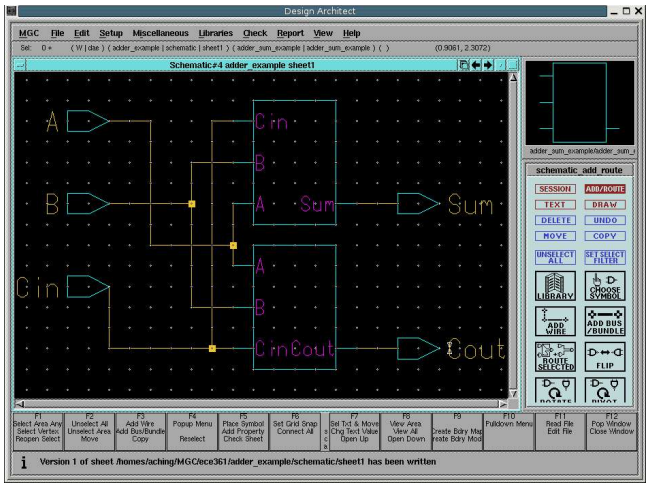
9. Now, use the same techniques to create the Cout portion of the adder. File->Open-

>Sheet and rename the Component Name to /homes/<your username>/MGC/ece361/adder_cout_example...and various other steps as we did with the Sum part. You should end up with something like this.



10.Now do the checks and create a symbol in the same way as before. We will use these symbols to create our 1-bit adder.

11.File->Open->Sheet and rename the Component Name to /homes/<your username>/MGC/ece361/adder_example. Now in the schematic_add_route window->choose symbol to add your symbols to the schematic. Add portin ports and portout ports and wires to make it look something like this (hopefully a little less messy).



12.Now Check->Sheet and Check->Schematic before saving. Generate a symbol as we did in the past. Make sure to do all the proper checks after generating and saving a symbol and also after creating the symbol for the schematic. Close everything in design architect and then close design architect.

13.Simulation of the circuit. We will create a design viewpoint for digital simulation. In

the design manager window, select your adder_example and right click on it to Open->4 DVE.

14. Setup->(Quick)Sim, Fault, Path, and Grade.

15. Edit->Add->Primitive and enter 'model' for Name and 'analog' for Value. Hit OK to execute the dialogue box.

16. File->Save Design Viewpoint->Save As.... and then enter 'digital' in the dialog box and hit OK. Right click on the empty space in the window and choose Close Design Viewpoint. Close DVE.

17. Double click on adder_example in the File navigator in design manager. Click on 'digital' and then right click on it to Open->7 Quick Sim II.

18. File->Check Design. Change 'Simulation Checks' to Yes. Change 'Expand Messages' to Yes and then hit OK.

19. Setup->Kernel. Change 'Hidden' to 'Visible' and then choose 'Change...' to select 'Full Delays Min'. Change 'Visible' to 'Hidden' and then hit OK.

20. File->Open Sheet. Then select the port objects for A, B, Cin, Sum and Cout. Then right click and Add->Traces. This will open up a trace window with all of our portin and port out ports.

21. Now for each input (A, B, Cin), highlight it and only it and then right click to Force->Clock. For A, make the period 50 ns, B period 100ns and C period 200ns. Also change the Force type to fixed and check the 'Times are Absolute' dialog box for each force.

22. You have set up the inputs to act like clocks to see what kind of output we are going to get. Type 'init 0r' anywhere on the quicksim window and then 'run 200' to see what the outputs are. You should see the proper behavior for a 1-bit adder. When closing quicksim, choose the 'without saving' option. Congratulations! You have designed a 1-bit adder that could be chained to make a 32-bit or 64-bit adder.