

Learning to Love Xilinx Foundation 4.1iä in 40 Easy Steps

You all know how to design and implement a digital circuit with Foundation. But sometimes going from a good design to a working bit file is more stressful than it needs to be. Here are some steps you can take to make the process a little more pleasant. Ask a TA if you need more details.

Essentials

1. Any two nets on the same level that have the same name are the same logical net, even if they are not explicitly wired together. This has some important consequences:
 - * You do not have to draw a wire to connect two symbol pins. You can just give the same name to the nets connected to these pins. Figures 1 and 2 are equivalent, but Figure 2 is easier to understand and debug.

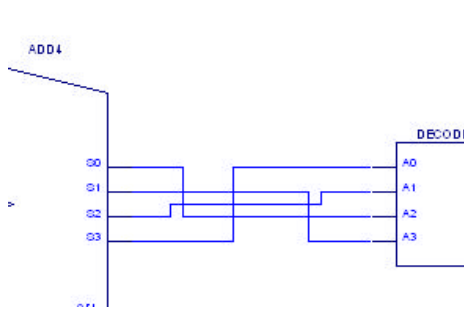


Figure 1

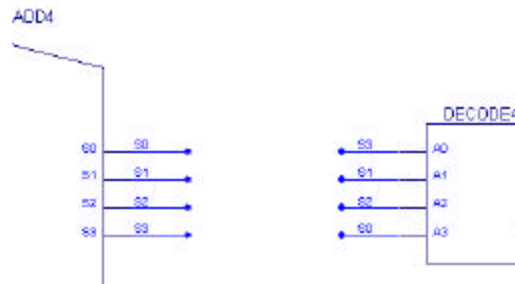


Figure 2

* Two nets on the same level of hierarchy that you want to be distinct will actually be connected if you give them the same name. The circuit in Figure 3 will not work because both the OR gate and the AND gate drive the same net.

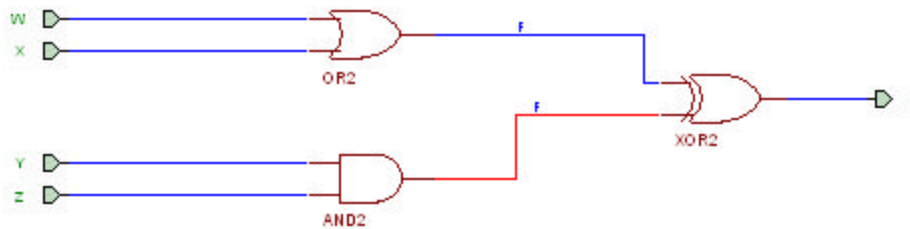


Figure 3

2. Only have one schematic listed in the Project Manager: the highest-level schematic in your project. If you have two schematics listed, they will be on the same level of hierarchy and Foundation will treat them as if they were two parts of the same schematic.

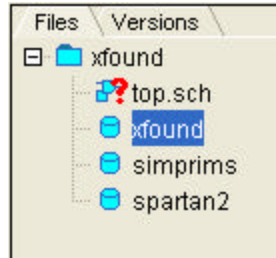


Figure 4

3. By default, the simulator will load the netlist of the schematic(s) in your Project Manager window. To simulate a macro by itself, choose Tools ->Simulate Current Macro from the Schematic Editor...

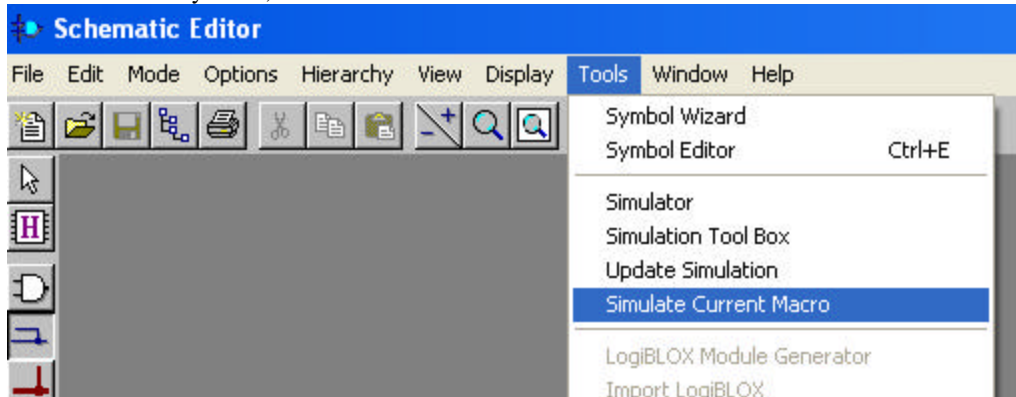


Figure 5

...or choose File->Simulate Single Component from the Logic Simulator and choose the macro that you want to simulate.

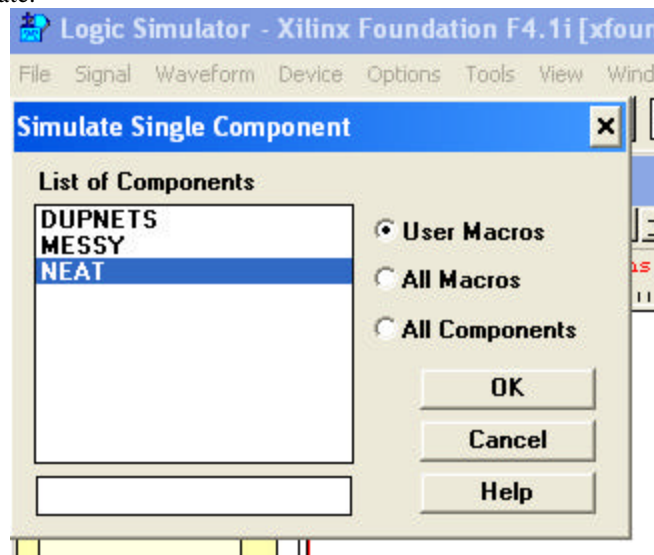


Figure 6

4. If you want to look at signals deep within the hierarchy in simulation, just string the reference names together, separated by slashes, in your script. For example, if we are interested in a signal X inside a macro with reference H1; and H1 is inside macro H3, H3 is inside H5, and H5 is in the highest-level schematic; then we can refer to X in our script as "H5/H3/H1/X."

5. The macro and the schematic from which the macro is derived are different. Always open the macro, either with the H “hierarchy” toolbar or with File->Open Macro. If you open and edit the schematic instead, the macro will not reflect your changes.
6. Before you declare a macro done, you should *always* perform an integrity test on the current sheet. Look at all the warnings in the Project Manager window and correct them if necessary. You should also test your entire design with Options->Integrity Test.

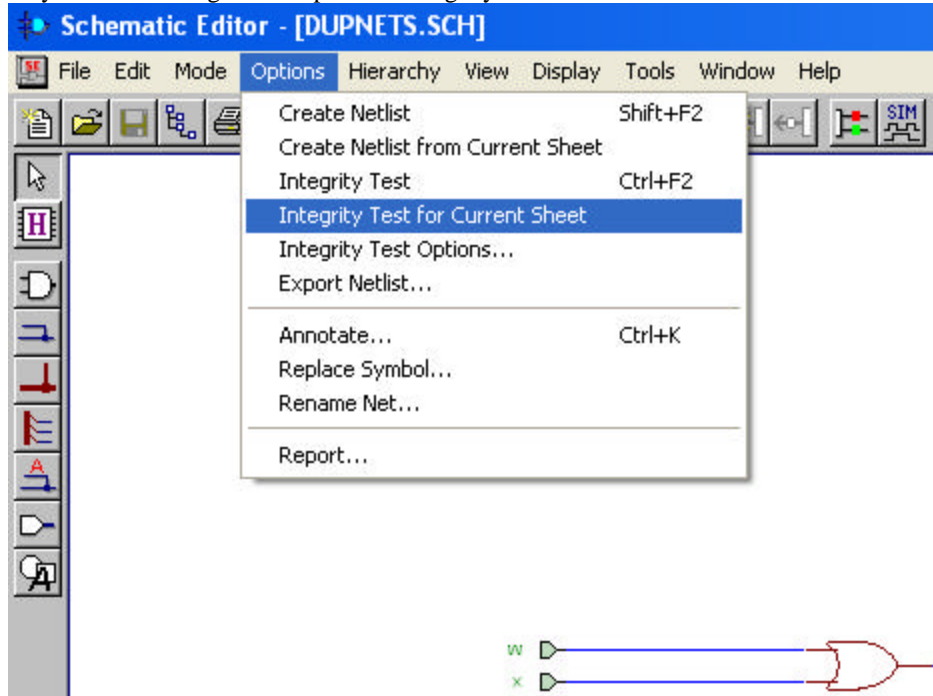


Figure 7

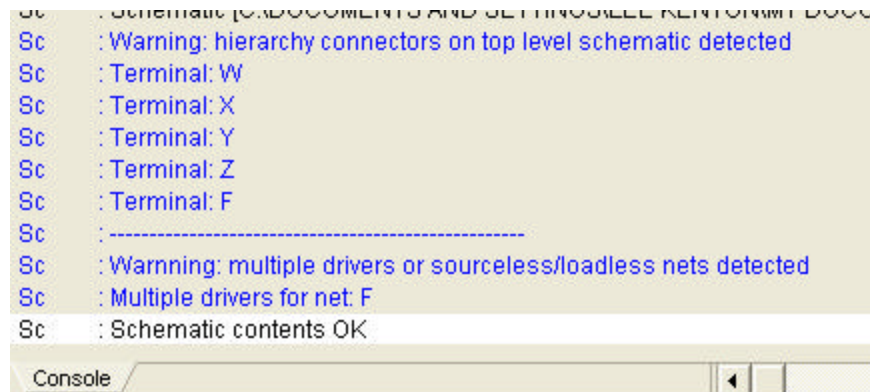


Figure 8

Most warnings are serious matters. A few are not. The Integrity Test will warn you if you haven't connected every output of a symbol, but that's often not a problem. For example, if you only need to use four of the eight outputs of a 3-8 decoder, you could connect nets only to the four you intend to use and leave the others disconnected. *Never* leave any inputs disconnected.

The warning about “hierarchy connectors on top level schematic detected” is OK only if you are about to make the schematic you tested into a macro.

7. What do all the components in the Spartan2 library do? Find out in the Libraries Guide, (<http://toolbox.xilinx.com/docsan/xilinx4/manuals.htm>) a link to which is in the Resources section of the class web page.

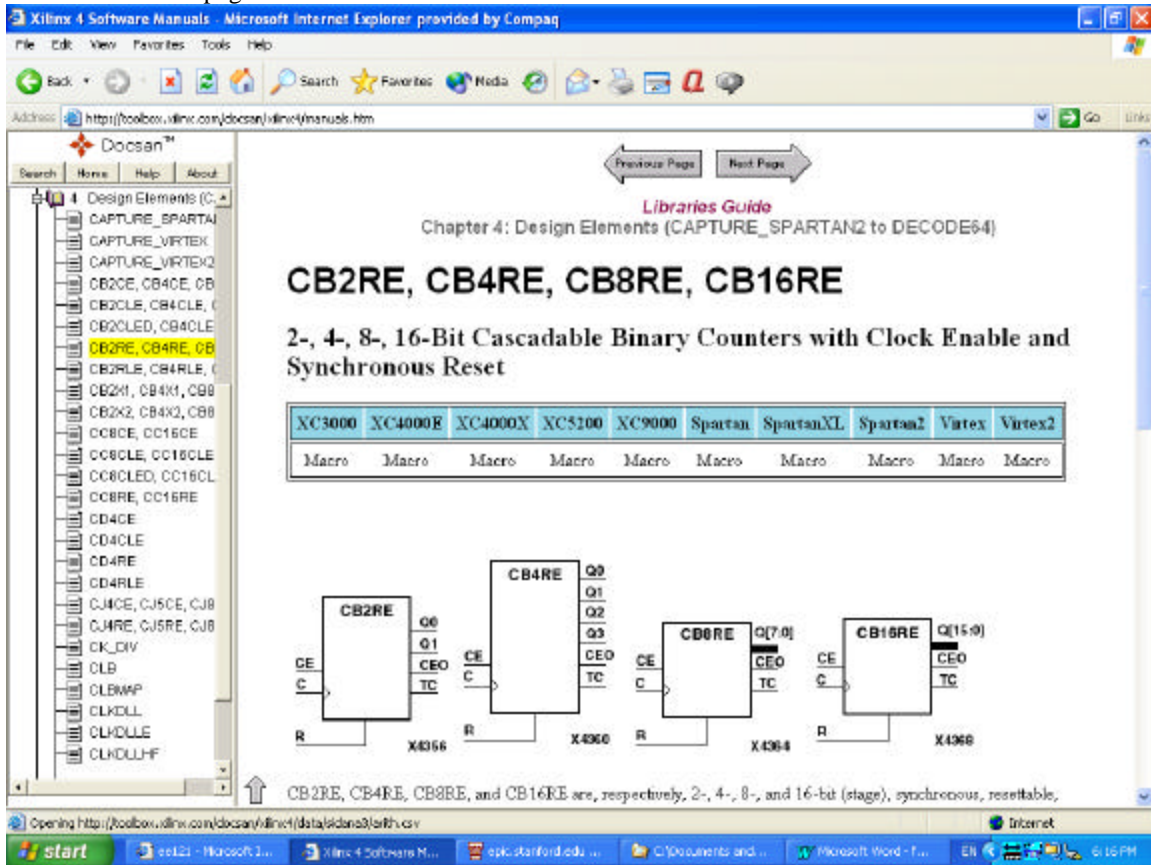


Figure 9

8. Don't copy your project in Windows. Archive it or use the Copy Project command.

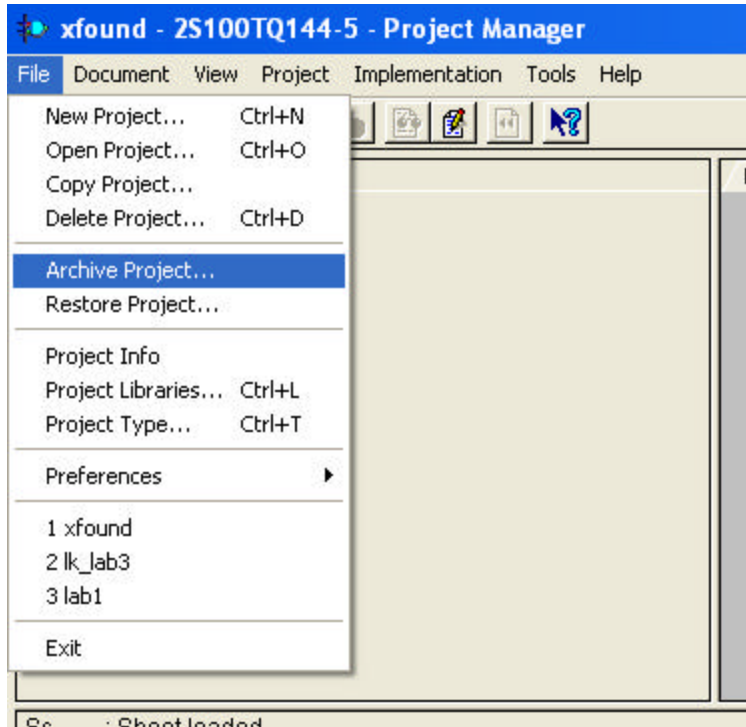


Figure 10

Useful

- Use the “Draw Bus Tap” tool to number and draw bus taps automatically. All you need to do is click the source bus, then click the pins to which each consecutive bus tap should be attached. The software does all the drawing and labeling for you. You can use the arrow keys to switch among bits of the tap.

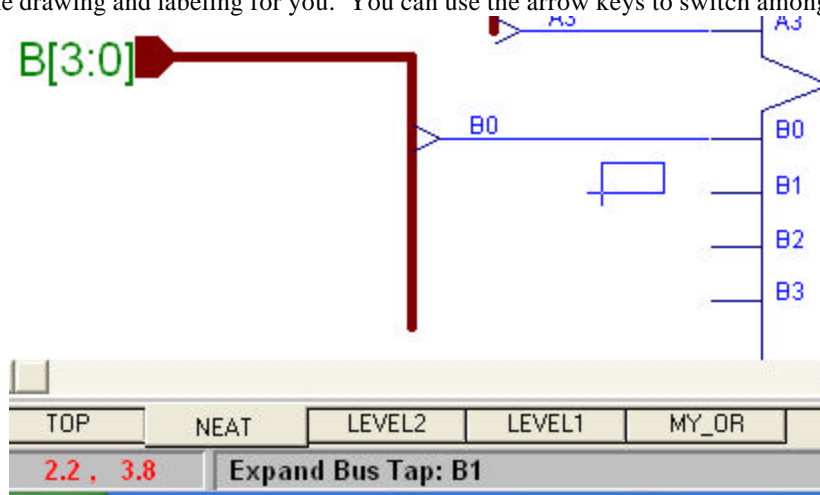


Figure 11

- Use the “Add Net or Bus Name” toolbar in increment mode to increment net names automatically. Enter the name of the net corresponding to the lowest bit and click Repeat. Then click the nets you want to label in order of increasing bit number.

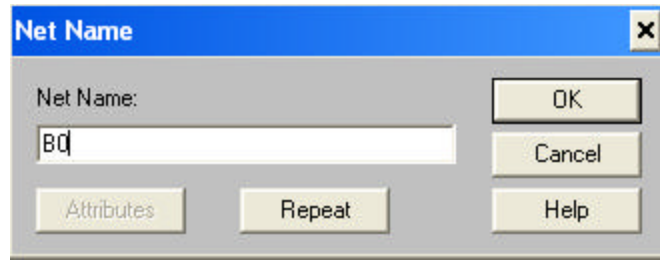


Figure 12



Figure 13

11. Make sure that you have labeled all your bus taps. The arrangement in Figure 14 won't work.

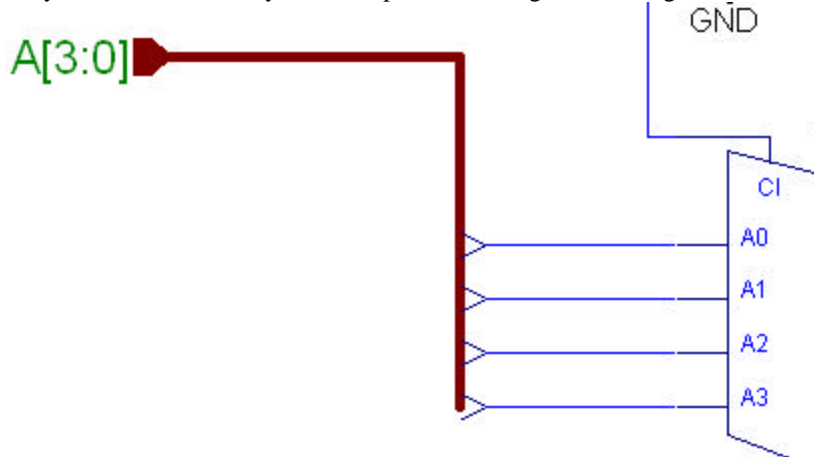


Figure 14

12. If the schematic looks weird, refresh the display by striking the F10 key.

13. The buffers and pads (Figure 14) are I/O for the chip. Hierarchy connectors (Figure 15) are the I/O for macros.



Figure 15

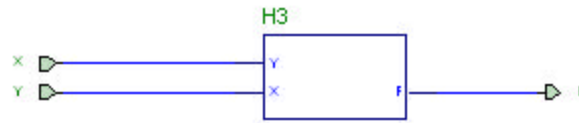


Figure 16

14. Read the report files when implementation doesn't work (or even when it does).

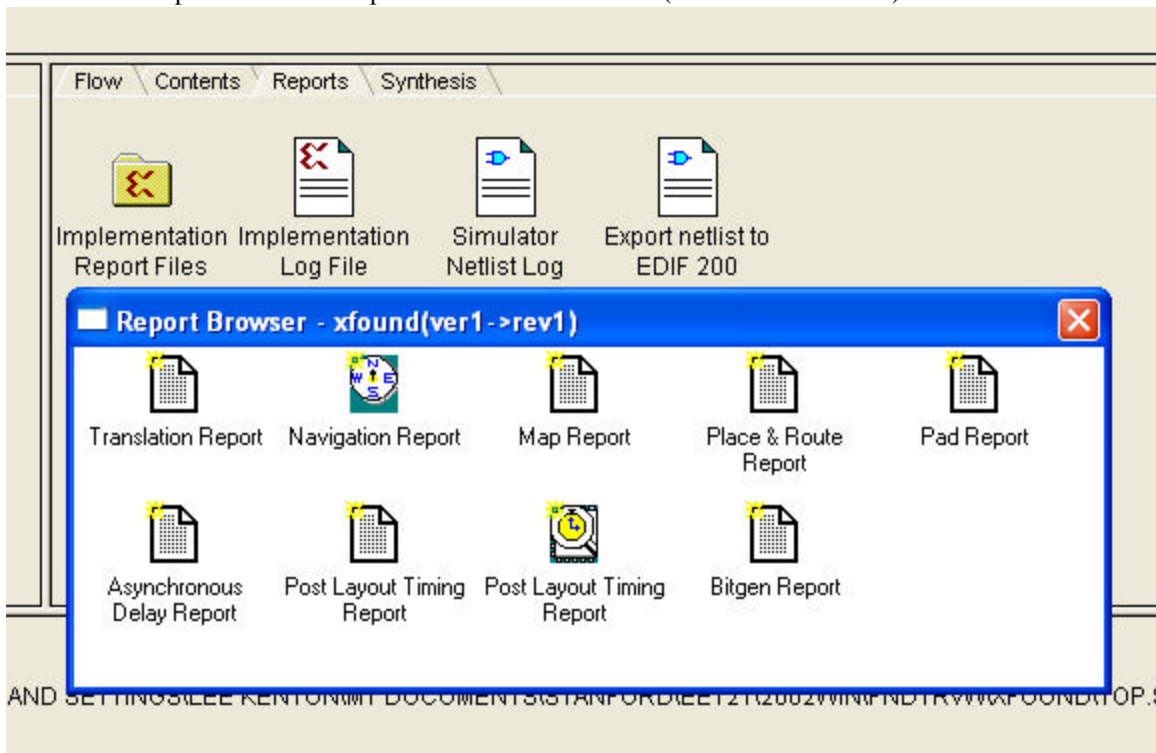


Figure 17

15. Use the View->Show All References to see the names of all your components. This will make it easier to find components in error and warning reports.

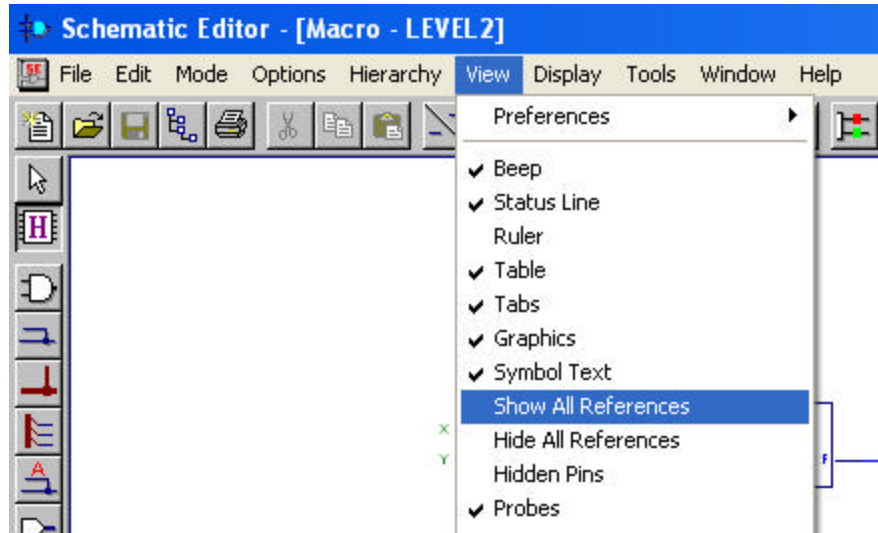


Figure 18

16. Every net has a name, whether or not you explicitly name it. Select the net and look at the bottom of the screen to find out its name. Better yet, give it a name yourself.

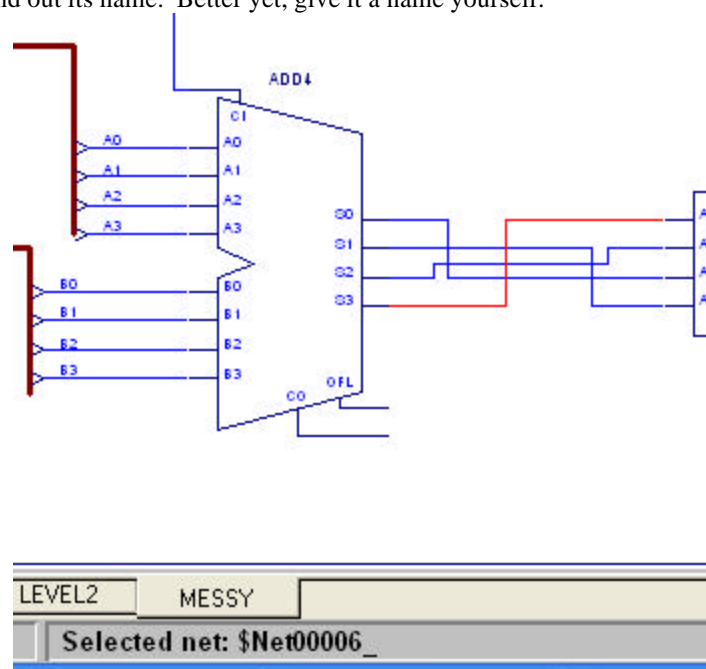


Figure 19

17. The name of the chip output is specified by the signal between the OBUF and the OPAD. Likewise for chip inputs and IBUFs and IPADs. In Figure18, F_OUT is the name of the signal that connects to the pin and the name you will find in your PAD report file.

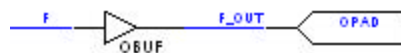


Figure 20

18. Assign pin numbers to every OPAD and IPAD. If you don't, Xilinx will do it for you.

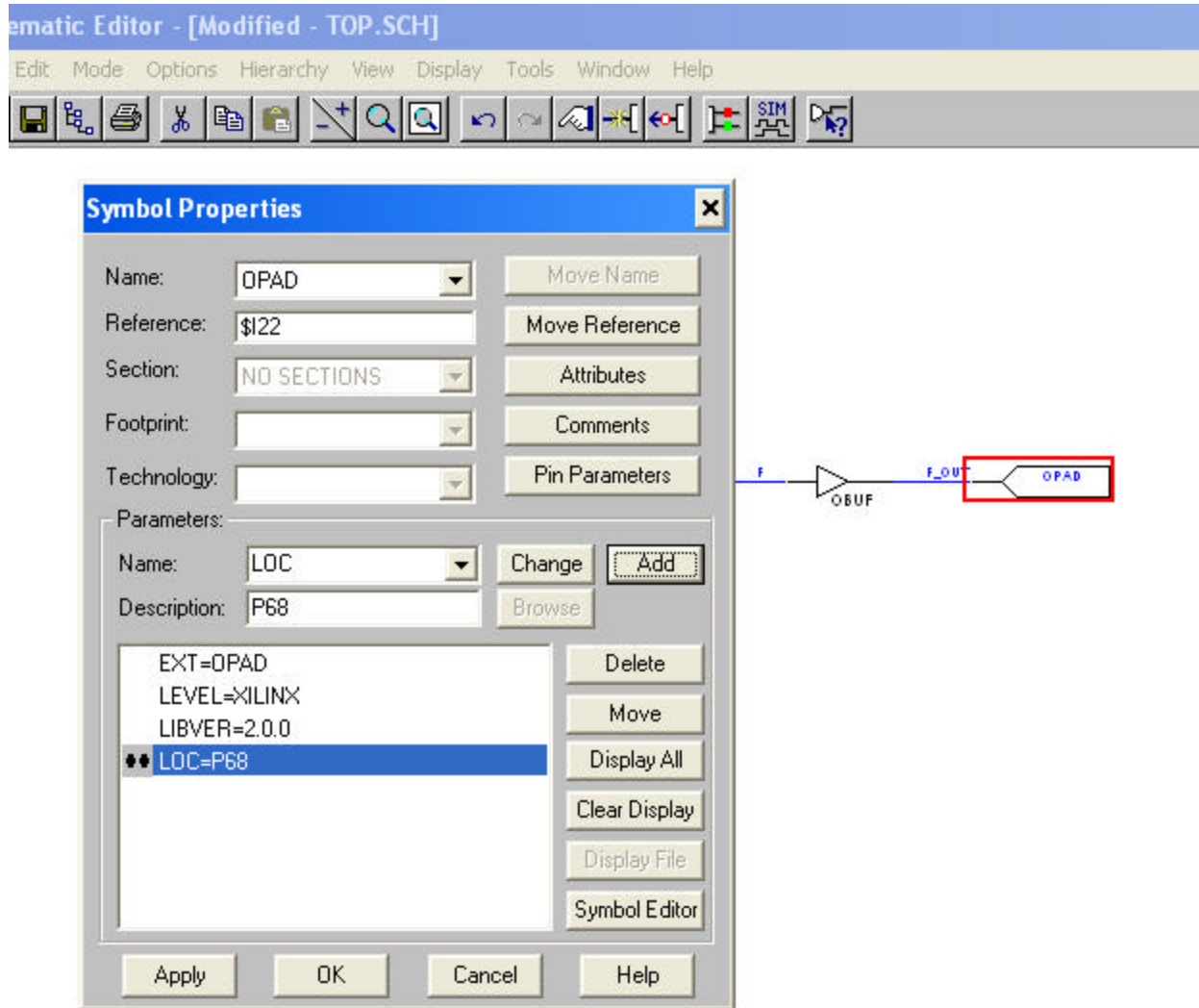


Figure 21

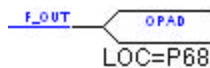


Figure 22

19. Beware of disconnected nets! Disconnected single nets will usually have blue dots, but not always. Disconnected busses will never have blue dots even if they are disconnected. Move components around to make sure the connected nets move with them. The components in both Figures 23 and 24 are disconnected, although you can not tell this from visual inspection of Figure 24. Disconnected nets often occur when you replace one macro with another. The Replace Symbol command (right-click on the component, then choose Replace Symbol) is less likely to lead to disconnected nets than other methods of replacing symbols, but only if the symbol interface of the new macro is the same as that of the old one.

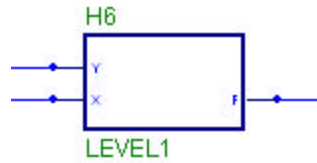


Figure 23

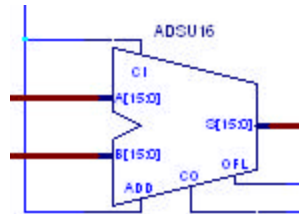


Figure 24

20. If Foundation complains about “no permissions or file does not exist” while archiving your project, you have to reboot. It’s a bug and we know of no other solution. ☹
21. Update the simulator if you make a change in the schematic. You can’t change the schematic in the middle of simulation. The safest thing to do if you need to change your schematic is to close and restart the simulator.
22. Use the vector script command to put things in the right direction. “vector x x[3:0]” and “vector x x[0:3]” arrange the bits in opposite orders. This doesn’t change your schematic, but it does change the way your “assign” statements and simulator output appear.
23. Use the clock command to simulate clocks. “clock CLK 0 1” makes the clock signal toggle from 0 to 1 after one simulation step, then from 1 to 0 after another simulation step, *ad infinitum*. “clock CLK2 0 0 1 1” would give CLK2 twice the period of CLK.
24. Simulating in timing mode is a good idea since your components will have non-zero delay, as they do in real life. But before you place and route your design, the delays the simulator uses in timing mode will only be wild guesses.
25. Don’t confuse your projects with those of others. If you’re working on the lab or cluster computers, archive your project, save it on your network drive, and delete it from the local hard drive. In addition, give it a unique name; for example, George W. Bush would name his Lab 3 “gwb_lab3” so it would not be confused with Tom Daschle’s Lab 3.

Geeky

26. Change the orientation of a symbol. Use Ctl-M to mirror, (Figure 25) Ctl-L to rotate. (Figure 26)

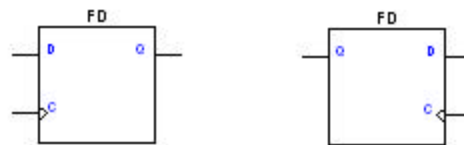


Figure 25

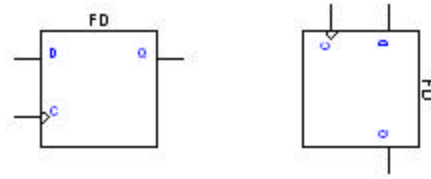


Figure 26

27. Use complex busses selectively to make busses that are conglomerations of different signals.

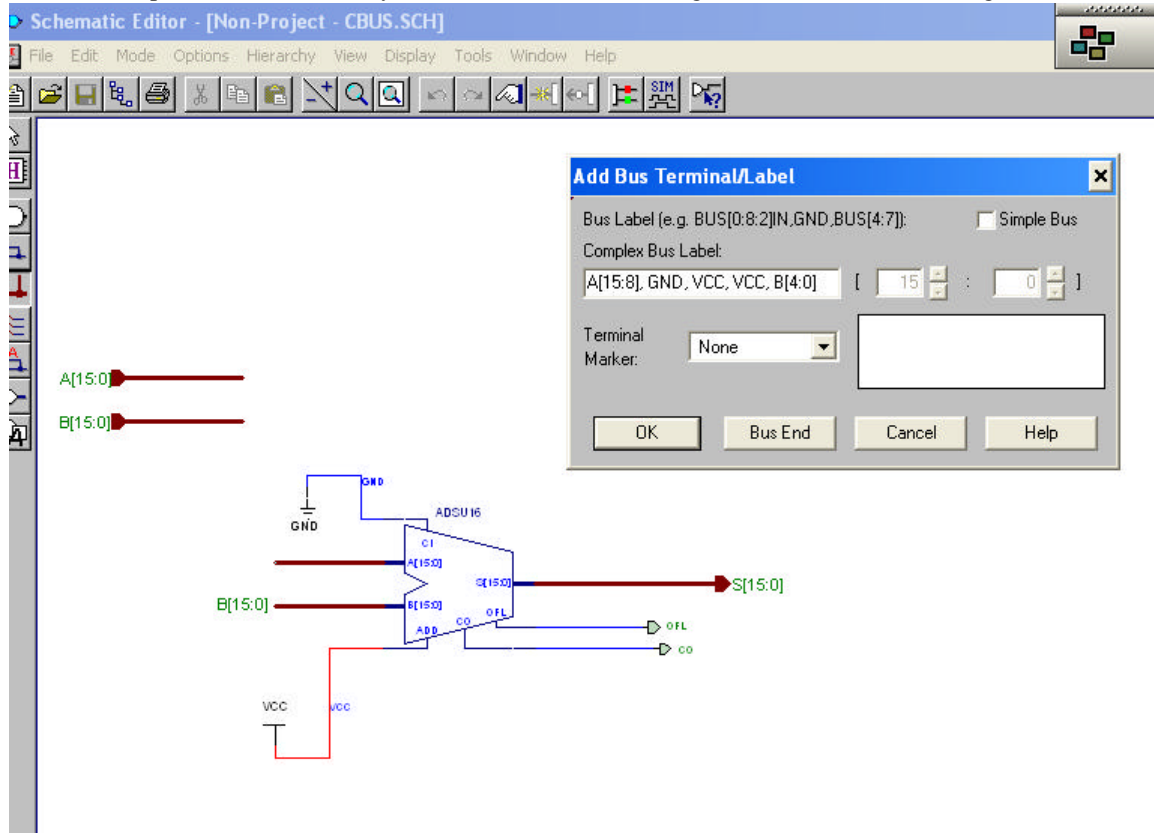


Figure 27

28. Use the \$ARRAY parameter to create multiple instances of the same component. Although this works, the display doesn't necessarily reflect your changes very accurately.

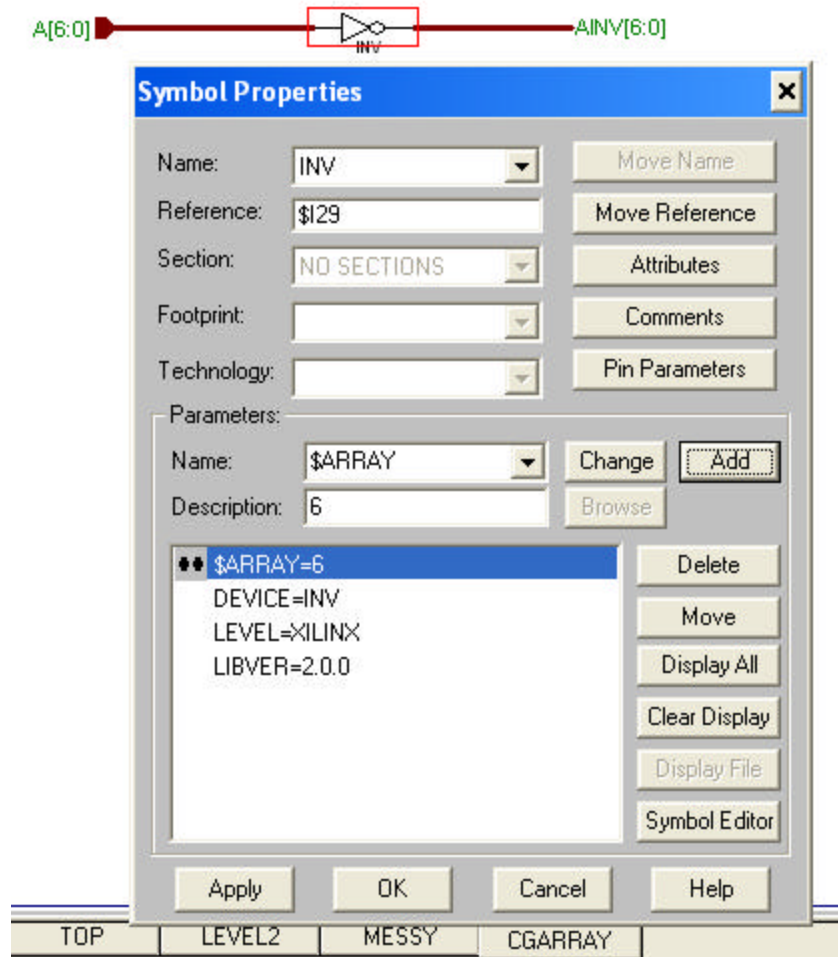


Figure 28

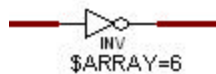


Figure 29

29. Better yet, use the Core Generator to generate all kinds of components. (But remember that you have to make your components in Lab 3.) Be sure to set your Design Entry environment to “Schematic-Foundation” and your Target Architecture to “Spartan2.” The Core Generator can be very useful, but its uses are beyond the scope of this document. Feel free to play around with it and see what it offers.

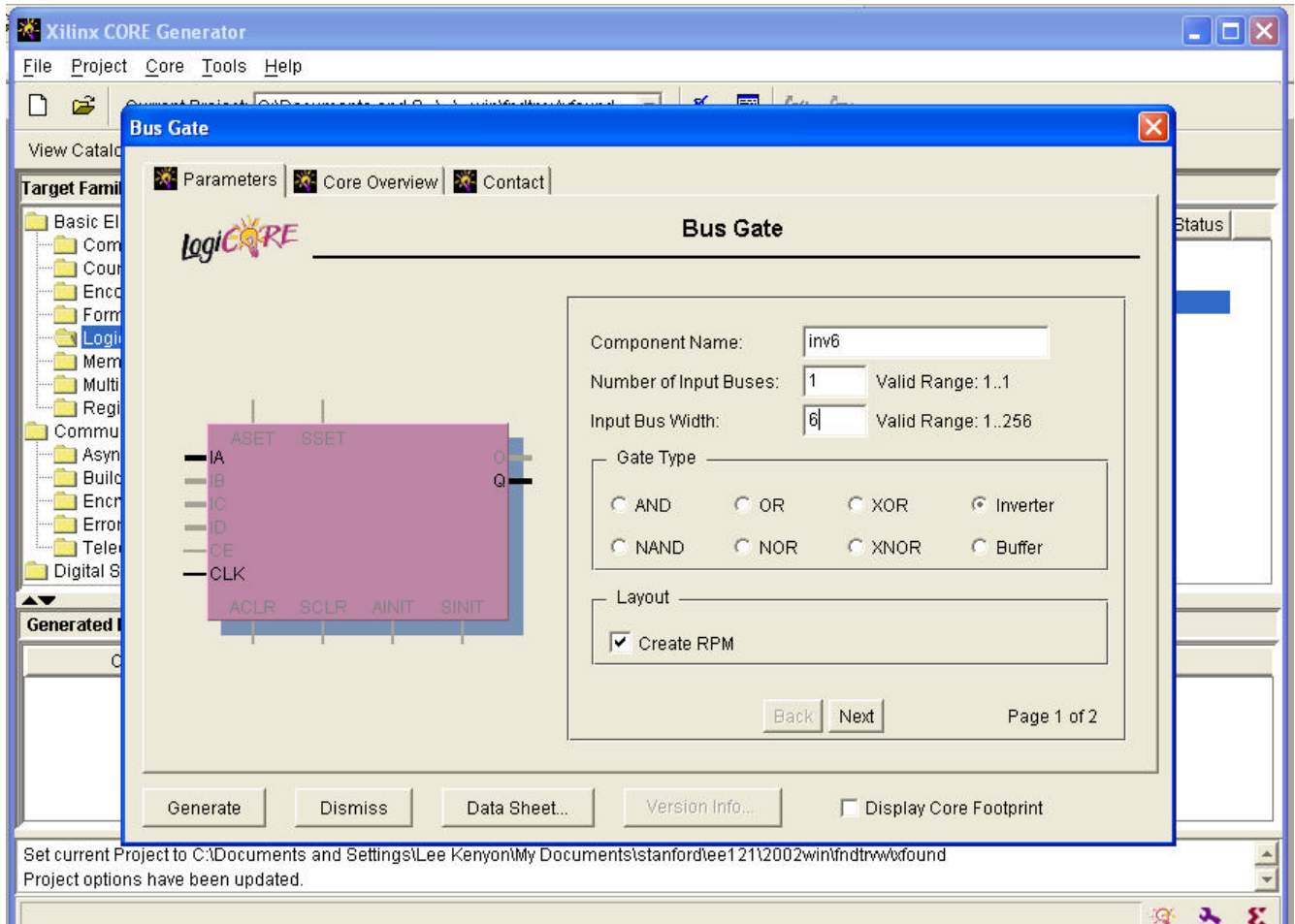


Figure 30

30. Find components fast by typing their name in the "SC Symbols" window.

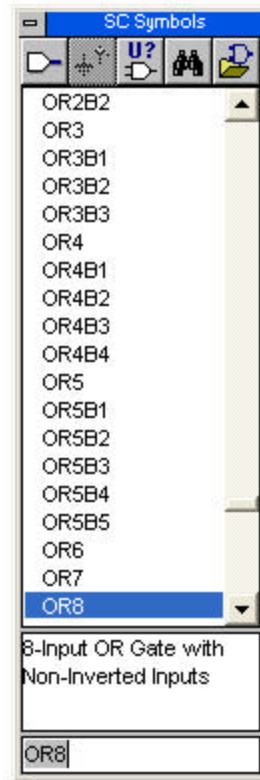


Figure 31

31. The left-most bit of a bus is the most significant, regardless of whether it is the highest bit or not. Simplify your life by ordering busses consistently.
32. Busses are connected to symbol pins in a left-to-right order. For example, in Figure 32, INT[15:8] is connected to FF2\D[7:0]. INT[7:0] is not connected to anything. What a mess! It's best always to have the same number of bits on both sides.

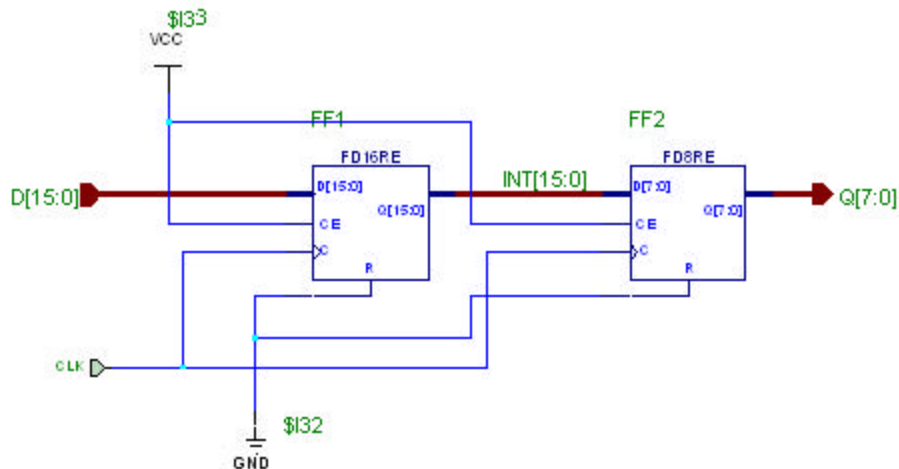


Figure 32

33. Beware bus names that end with a digit. For example, if you had a DATA1[10:0] and a DATA[10:0], then DATA10 would exist twice.
34. Use File->Table Setup to identify the schematic as yours. (We don't care whether you do this or not.)

Stanford University - EE121	Crafted by: Horatio Q. Student
350 Serra Mall	Macro: BUSCON
Stanford, CA 94305	Date: 1/26/84
Date Last Modified: 1/27/02	

Figure 33

35. Use the autoroute feature (Edit->Autorouting Table) if you think it's worth it.
36. If the simulator tells you that you have a "combinational feedback error," this probably means that a net that is a function of an output of a combinational logic element feeds back into an input of the same element. Look for naming errors. In a few cases, the simulator has generated a combinational feedback error with no apparent reason, and students have solved the problem by replacing their logic with equivalent components.
37. Use the Project Libraries Window (Tools->Project Libraries) and the Library Manager to manipulate the contents of your libraries. Generally, EE121 students have not used this very much.

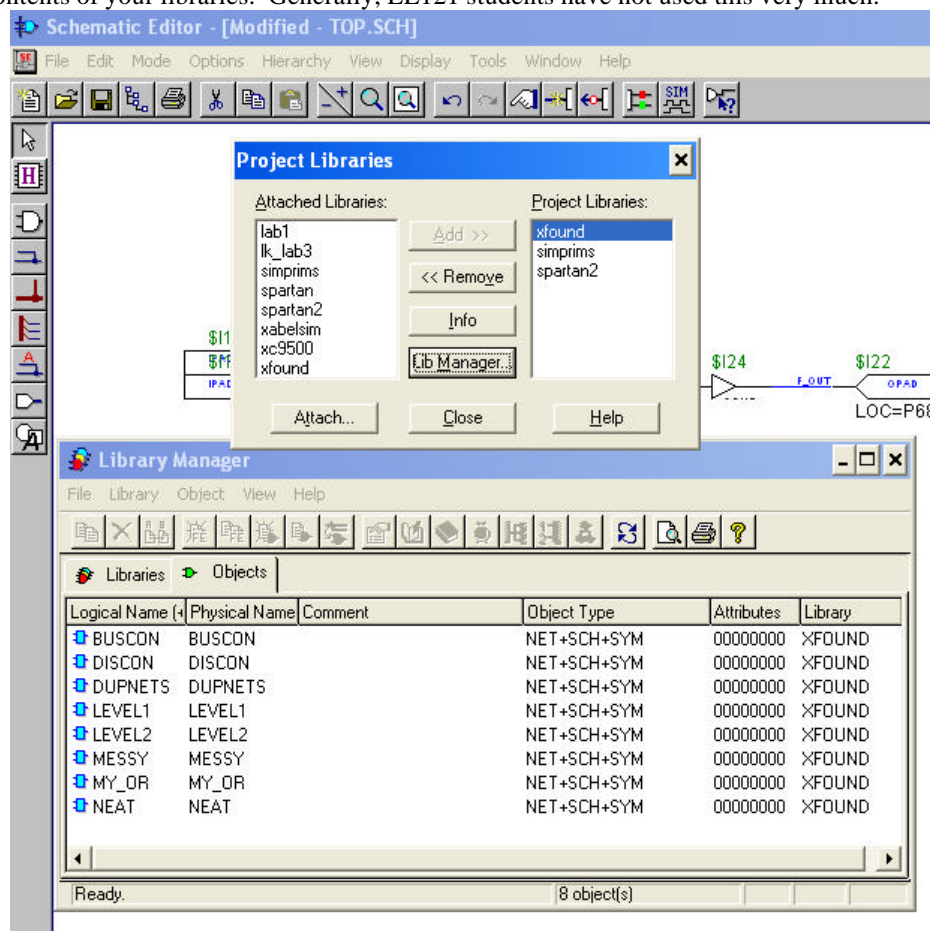


Figure 34

38. Using the symbol editor to change the symbols corresponding to your macros. Right-click on the component and choose "Symbol Editor." You can do all sorts of things, like move pins around, make bus I/O pins thick, and place dots on active-low pins.

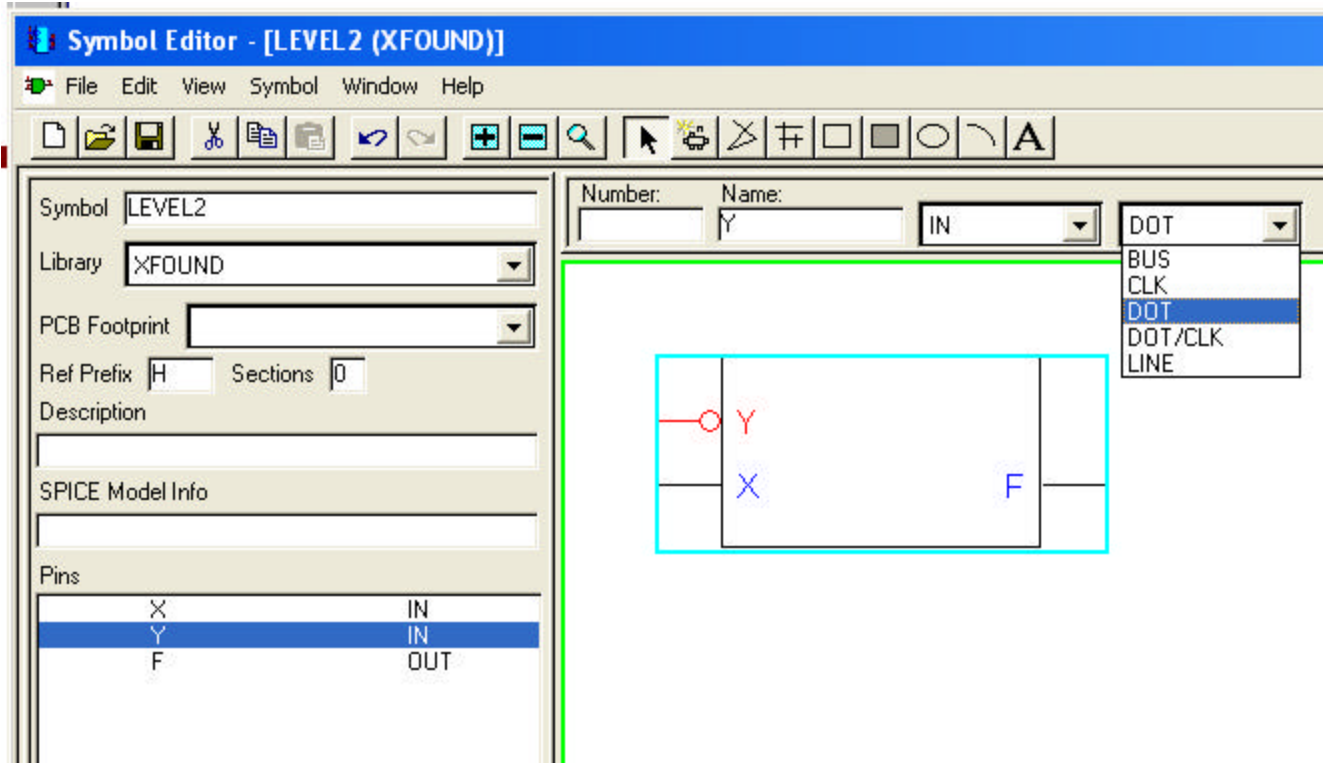


Figure 35

39. I/O buffers and pads can go inside macros; they don't have to be on the highest-level schematic. (But they should be in Lab 3 because that's the way the skeleton file is.)
40. IFDs and OFDs replace IBUFs and OBUFs. You can register your inputs and outputs by replacing OBUFs and IBUFs with IFDs and OFDs.