CS 210 Laboratory Exercise One

Logic Design with MAX+plusII

In this lab, you will learn the basics of a computer-aided design tool called MAX+plusII ("Max Plus 2"), which can be used to design digital circuits and simulate their behavior. In this first lab, you will learn about the basics of the program and design a simple digital circuit involving only a couple of gates. In future labs and assignments, you will learn how to design more complex circuits and analyze their behavior, concluding with the design of a basic processor.

Starting MAX+plusII

To use MAX+plusII in the CS lab, there are a couple of things you need to do <u>once,</u> the first time you run the program in the PC lab:

1. After logging into the PC, start X windows on your PC by clicking on **Programs → Unix Tools → Exceed**, choose the "desktop" option, then open a terminal session to CSA
2. Next you must set the path in your **.cshrc** file. Using your favorite editor (e.g., Emacs) open the file **`.cshrc`** in your home directory and look to see if you see a line of the form **`set path=(. . .).`**
   a. If not (the default case), then insert the following line at the end of the file**:** **`set path=( $path /usr/local/maxplus2/bin )`**. Use all lower case.
   b. If so, look for **`/usr/local/bin`** in this list (probably it is the first entry) and type in **`/usr/local/maxplus2/bin`** right after this entry, <u>all lower case</u>. Make sure there is a blank space on each side of what you type.

   Save the file, exit the editor, and then type **`source .cshrc`** to the shell prompt.
3. Now create a directory to save your work; from your home directory type **`mkdir maxplus`** and then change your working directory to this directory by typing **`cd maxplus`** .
4. Start the program by typing **`max2win`**. <u>From now on you only need to do this step to run MAX+plusII in the main terminal room on the Unix machines.</u> The first time, you get a message saying *Enumerating fonts…* If you get an error window about missing fonts, just click "OK." When a new

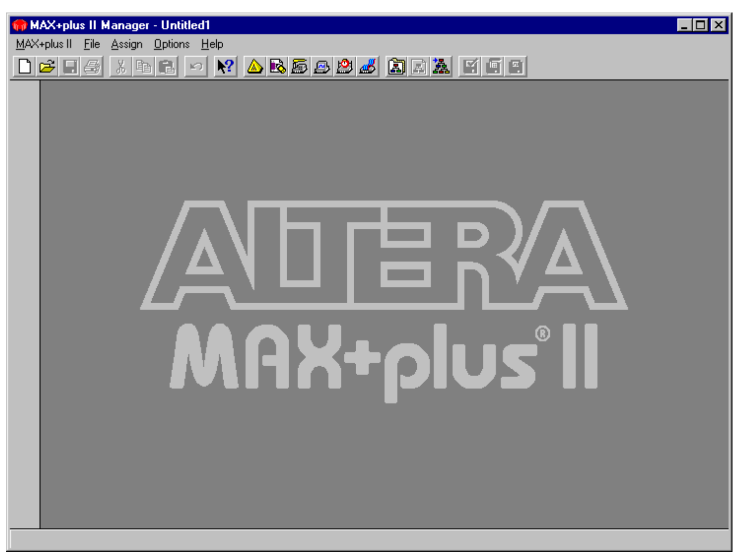

Figure 1

screen pops up, you should be looking at the main window for Max+plus II, with the Altera logo in the background (Figure 1).

Using the Graphic Editor

Now that you have started Max+plusII, you are ready to start working with the Graphic Editor, which is used to draw circuit diagrams (usually called "schematic diagrams" in Max+plusII). A schematic diagram can be compiled and run using a simulator, producing a waveform, which shows the values on various wires over time. In this lab, we will just learn how to draw circuits, using as an example the circuit shown in Figure 6 on the last page of this lab writeup. You should study this circuit before proceeding.

1. From the leftmost menu, called simply **Max+plus II** , select **Graphic Editor**. This will open a blank Graphic Editor window as shown in Figure 2. Make sure this is the active window; you may maximize both the main window and the Graphic Editor window to get more workspace. Move the cursor along the buttons across the top and on the left side and read the descriptions that appear at the bottom of the main window.
2. Now, make sure that the **Select Tool** (a large arrow button on the top of the left toolbar) is currently selected (click on it to select).

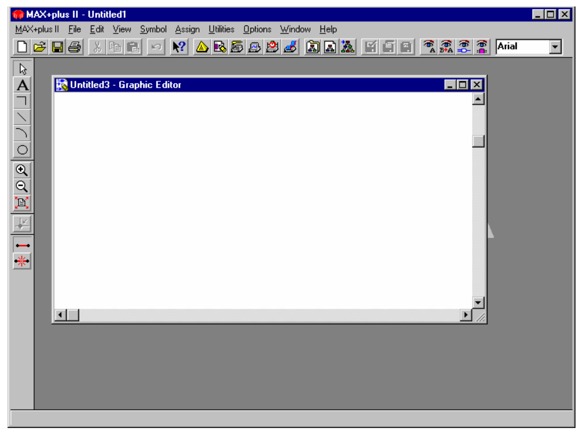

Figure 2

3. Next click on the Graphic Editor window, in the approximate center of the empty space. This defines a position to paste an object.
4. You will now select an object to paste into the diagram from a library of gates and basic circuits that is built into Max+plusII. From the **Symbol** menu (fifth from the left along the top), select **Enter Symbol**. This will bring up the Enter Symbol dialog box. From the Symbol Libraries section, double click on **/usr/local/maxplus2/max2lib/prim** , which is the logic primitives library. A list will appear in the Symbol Files area. This is a list of the most primitive logic gates, including **and**, **or**, **not**, **nand**, and **nor** gates with various numbers of inputs (e.g., **and2** is a two-input **and** gate, and **or8** is an eight-input **or** gate); take a moment to look through this list. Then, select **and2** and click OK (or just double click on **and2**). A symbol for a 2-input **and** gate is placed in the Graphic Editor Window.
5. Now add the second **and** gate. Since the first **and** gate is selected, you can simply copy it to the clipboard by typing **Control-C**, select a position below the first gate by single-clicking with the mouse, and then paste by typing **Control-V**. (As you might expect, you can delete an object by selecting and then typing **Control-X** or **Delete;** you can also use the Edit menu for basic manipulation.)[1]

[1] Max+plusII offers a variety of shortcuts, most of which you should already know if you have a PC; you should explore the help system and the documentation listed on the CS 210 web page to learn more about the features of Max+plusII as the semester goes on.

6. Now we will add an **or** gate. Double click to the right of the two gates added so far (another shortcut). Now add an **or2** symbol using the Enter Symbol dialog box.
7. If you do not like the position of one of the gates, you can drag it to a new position.
8. If you want to zoom in, use the View menu. Figure 3 shows the result of zooming in to get a better look at these three gates.

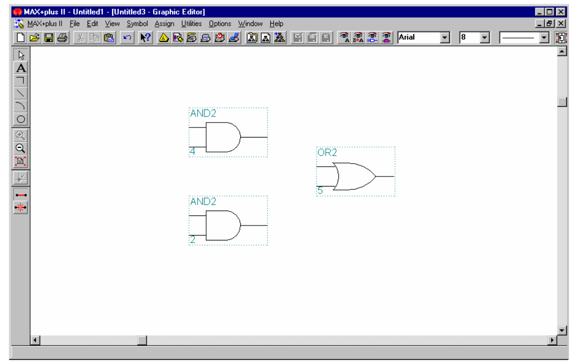

Figure 3

9. Next, add a **not** gate to the left of the **and** gates. Rotate it so that the triangle is pointing down using the **Rotate** command from the **Edit** menu.
10. Lastly, we will add three inputs **and** one output to the design. Inputs **and** outputs to a circuit are symbols in Max+plusII, **and** (like parameters to a C++ function) must have names. Using the same procedure as above, add three **input** symbols (simply called **input** in the **prim** library) to the left of the **and** gates, and one **output** to the right of the **or** gate. You may need to zoom out to fit all these in the drawing space.
11. Your inputs and outputs (like parameters in a C++ function) must have names. They come with a default name, but you must rename them; to do this, double click on the **PIN_NAME** of the top-most **input** symbol and type in "A" as its new name. Similarly name the other two input symbols "B" and "C" (you may use lower case if you wish). Finally, rename the **output** symbol "Out". <u>Make sure that the name has no spaces.</u> Figure 4 shows the workspace after adding the inputs and output and zooming back out to show all the devices.

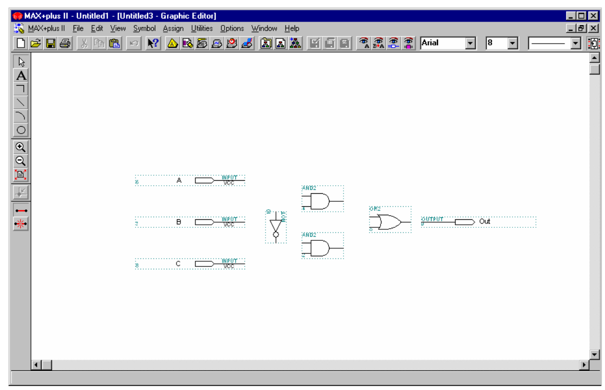

Figure 4

12. Now we must connect the logic devices with wires. A useful default behavior for wires is "rubberbanding," which means that once two devices are connected, they will stay connected if one is moved (you can look this up in the Help system if you like). Look in the left toolbar, and make sure the **Rubberbanding** icon (two black dots connected by a red line, second from the bottom) is selected (you can toggle between the bottom two buttons to turn rubberbanding on and off).
13. In the same toolbar, select the **Wire Drawing Tool** by clicking on the icon with two straight lines joined at a right angle (third from the top). Position the cursor over the right side of the "A" input symbol, at the right end of the wire. Click and hold the left mouse button, and drag the cursor to the top input of the top **and** gate. Release the mouse button. A wire has now been drawn from the input to the **and** gate. Now click on the "B" input, and draw a wire to the second input of the top **and** gate. Similarly

connect the “C” input to the bottom input of the bottom **and** gate, the output of the **not** gate to the top input of the bottom **and** gate, and the connections involving the **or** gate (following the diagram in Figure 6). Do not connect the input of the **not** gate yet.

14. If you change your mind, you can select a wire and hit **Delete**, but note that sometimes only a part of the wire is deleted. Try using the **Select Tool** to move an input, and watch how rubberbanding works. Also, you can experiment with dragging on part of a wire using the **Select Tool** to move the wire around a bit if you don’t like its shape.
15. The last step is to connect the input of the **not** gate to the wire between input “B” and the top **and** gate. Simply draw a wire from the **not** gate to the wire, and observe that a connection dot appears in the joining of the two wires. This is the default when connecting the end of a wire to the middle of another wire. Now choose the **Select Tool** and click near the connection dot, and click once or twice on the **Connection Dot Tool** (third from the bottom) on the left toolbar. Observe that this toggles the dot on or off. In this way, you can make sure there is a connection (say between two wires that cross) or not. The final circuit diagram is shown in Figure 5.

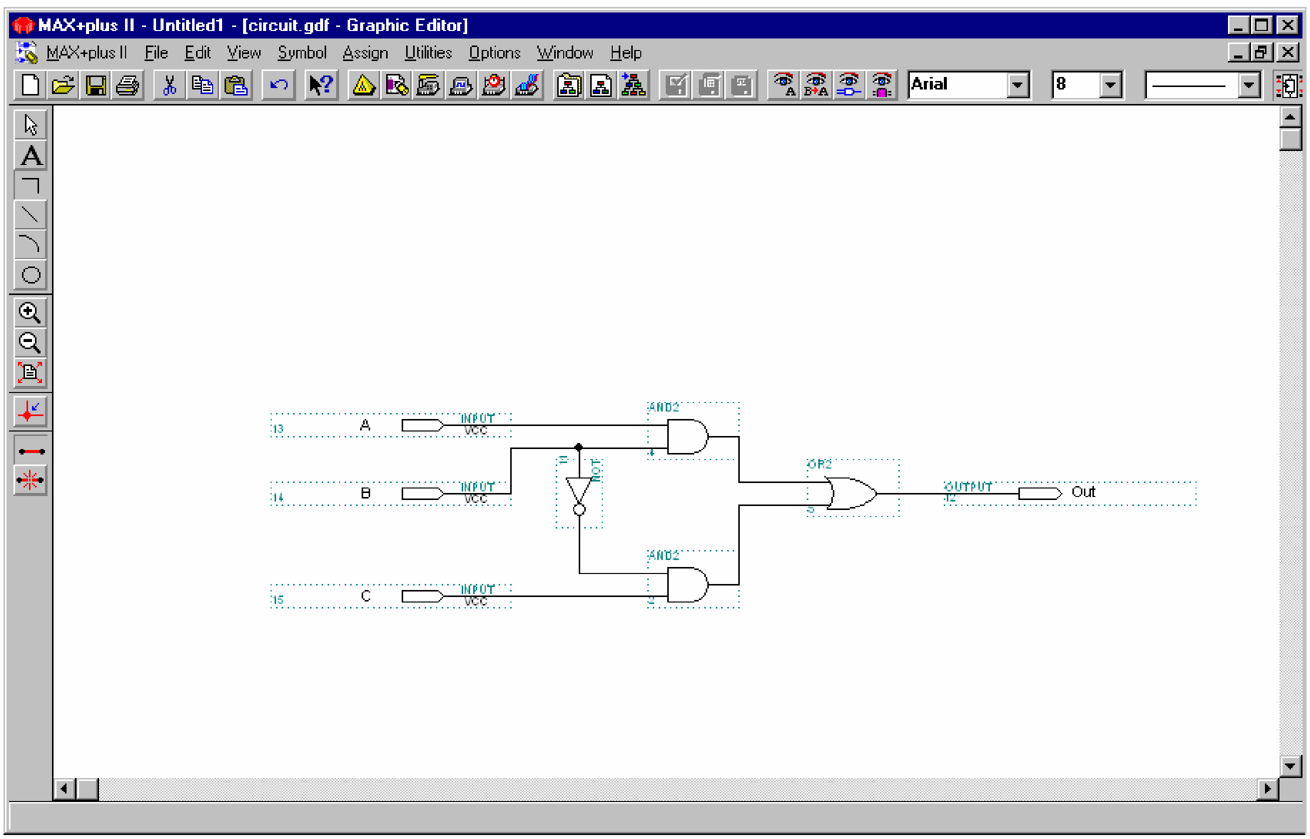

Figure 5

16. Now we will save our work, and compile it to check for errors. Choose **Save As** from the **File** menu and save the file as **circuit.gdf** in the **maxplus** directory you created in step 3 of this lab.

17. Finally, from the **File** menu choose **Project → Set Project to Current File** and then choose **Project → Save and Compile**. This will pop a compiler window to compile the design. As the Compiler processes the project, any information, error or warning messages appear in a Message Processor window that opens automatically. When the compilation is finished, icons representing the output files generated by the Compiler appear below the module boxes. Max+plusII will give you advice on errors if you select an error message using the **Message** button, and then click on **Help on Messages**. Ask your TF about any errors that you can't figure out.

18. A typical error that is easily fixable is a connection that did not connect. If you are unsure if two wires are connected, or a wire and a device, <u>start to pull your devices around and see if they stay connected</u>. (This is one reason to use rubberbanding.) A connection will rubberband, and a missing connection will separate. This is also a good technique to see if you have accidentally drawn a wire underneath another wire, or under a device; moving things around will verify that there are no extra pieces causing problems.

19. Now you must do an extremely important thing to affect the way compilation proceeds in the future. While the compiler window is on top, you will see a menu **Processing** on the top of the Max+plus II window. If you pull down this window you will see that **Timing SNF Extractor** is checked; you should click on **Functional SNF Extractor**. This configures the Compiler so that it does not do many time-consuming steps (such as fitting your design into the constraints of an actual chip) that we have no need for; you should see the number of steps in the Compiler window reduce to just three, ending in Functional SNF Extractor, which creates a file ending in .snf that will allow us to simulate the circuit (which we will cover in Lab Two).

20. Exit the program using **Exit** from the **File** menu.

21. Note: when running the program next time, just type **max2win** to the Unix prompt; if you wish to work on this same diagram, select **Open** from the **File** menu, and double click on **circuit.gdf**. The file extension "gdf" stands for "Graphic Design File." Note that a large variety of other files are created when you compile your program; you can just ignore these for the present.

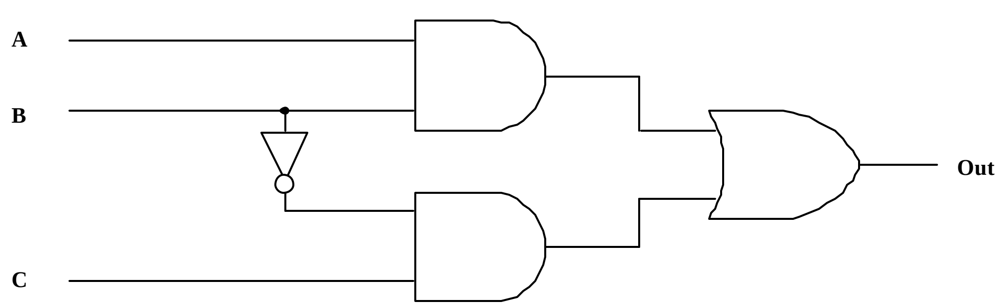

Figure 6