

## Lab 8

# Printed Circuit Board Artwork Generation

In Phase I of the class project (lab 7), you designed and simulated a circuit that would test an input signal and light up an LED depending on a HI or LO signal. You also prototyped your design on a breadboard and found issues that need to be addressed. In Phase II, you will implement a revised design and generate artwork which will be used to create a printed circuit board (PCB).

There are some extremely important things to keep in mind for this lab:

- It is absolutely imperative that you have a working circuit before you move ahead with this lab. We are about to design the physical layout of your design, so if your circuit is not currently working properly, you will build a non-working project. If you have simulation troubles with the new design, or any other problems that suggest your circuit is not 100% correct, we need to fix that **NOW!**, or your finished product will not work.
- You can not assume your circuit is identical to the example diagrams in these instructions. There are certainly some significant differences between your schematic and the one that generated these images. Therefore you will need to identify your equivalent components by figuring out what they are connected to, rather than using the labels. **DO NOT** simply match up your labeled components to the same labels on the diagrams here, this will most definitely generate an incorrect layout.

## Packaging the components

Before we convert the schematic to a PC board, we need to specify a package type for each component. This assigns a “footprint” to each component, which defines the physical characteristics including its size and shape, pin configuration and spacing, and other physical attributes. This allows us to save the appropriate amount of board space for each component when we lay out the board. It also determines where the components will penetrate the board which indicates where holes will be drilled and where solder pads will be located to attach the components.

Each component must have a footprint type defined in order to be included in the PCB design. There are several methods we can use to do this, and different methods are available depending on the component type. In some cases, this can be assigned in the *Place Component* screen when the schematic is originally created. But since we didn't do that when we created this schematic initially, we will use a combination of the other two methods. For most components, double-clicking on its schematic symbol will open a screen where we can assign various attributes, including the footprint. For certain other components (namely the resistors) we will need to use the *footprint* column of the spreadsheet view to make the assignment.

## 1. Packaging the standard components

- (a) Start the Multisim Schematics program.
- (b) Create a schematic for the revised design given in lab discussion.
- (c) Assign a package type to each device (different methods are required for different component types):
  - i. Resistors, Diodes, NAND chip, Header pins
    - Double click on the component
    - In the component attributes window, select the *Value tab*.
    - The footprint type is listed near the bottom of the window. If you know the footprint type you can type it into this box directly. If you want to select it from a list (a safer choice), click the button to open the *Edit footprint window*.
    - Click the *Select from Database* button.
    - Be sure you're viewing the Master Database,
    - Scroll through the list to find the appropriate footprint for each component, listed below. Be sure to look closely and get an exact match, there will be many similar-looking options. They are ordered by footprint number.
    - Note that you can also just type the footprint into the
      - Resistor: No. 3110 "RES10"
    - Leave the symbol and footprint pin lists at their defaults.
    - When you click *Select* it will open a pin editor window. You will need to assign each pin on the schematic symbol to the proper polarity footprint pin. Check your netlist to determine which Symbol pin is connected to ground (Net0), this pin gets mapped to the negative Footprint pin. Most likely this will be pin 2 but there is no guarantee, you must confirm this on your netlist.
  - ii. LEDs
    - Footprints for these components can only be assigned from the spreadsheet view. The default location for the spreadsheet is at the bottom of the Multisim window. If you do not see a table there, go to the *View* menu and select *Spreadsheet View*.
    - Click on the *Components* tab below the table. A list of components now occupies the table. Scroll down until you can see the *LEDs* (probably named LED1 and LED2). Scroll right until you can see the *Footprint* column.
    - Click twice on the cell for one of the LED footprints and the *Edit Footprint* window will open.
    - Click *Select from Database*, be sure you're viewing the Master Database, and assign the footprint as follows:
      - LED: No. 2005 "LED9R2\_5Vg"
    - Click *Select*
    - Keep the default pin mapping
    - Click *OK*
  - iii. Test Pin and Supply Lines
    - For the test pin, place a 1x2 header pin in your schematic. Remember to tie both pins together
    - For the supply lines, place a 1x3 header pin in your schematic. Connect your positive line to one end, skip the middle pin, and then connect your ground line to the other end

## 2. Finishing the packaging process

- Look at the spreadsheet view and confirm that every component has a footprint specified in the *Footprint* column. If so then move on to Part II *Converting the Schematic* to a PCB
- Be sure to remove any supplies used in simulating the design

# Creating an Ultiboard PCB from the Multisim schematic

Now that the preparatory work is complete we will now convert the schematic into a printed circuit board layout. We will export the Multisim work to Ultiboard. The software will keep track of what components are in the design and how they connect to each other as it converts from theoretical symbols to physical characteristics of the actual components.

## 1. Transferring from Multisim to Ultiboard:

- With your schematic open in the Multisim program, open the Transfer menu and select Transfer to Ultiboard: Transfer to Ultiboard 11.0.
- First it will ask you to save the file. Save the file in the same new folder where your new packaged schematic. Name the file with your first and last name. YES this matters, I will need to assemble all the individual projects from the whole class into a single project to send out for production so I need to be able to easily identify your file. It would be a very good idea to use File: Save As and save new copies with incremental numbers at various stages of this project so you can backtrack easily if need be.
- When you click Save it will open the Ultiboard program and display the Import Netlist window that lists all of the components and the connections between them. If you did your packaging properly you shouldn't need to worry about this stage. Click OK.
- The new ultiboard design will appear on the screen as a jumbled mess of components. We call this the "rat's nest." The components are shown as blue outlines which represent the size and shape of the actual parts. The yellow lines indicate the necessary connections that need to be made between the component pins. The yellow box is called the keep-in boundary. This represents the actual dimensions of the finished board. The brown lines are called Force Vectors and are supposedly a tool to help in positioning components, but their usefulness is widely disputed so we will ignore them. Your rat's nest should look similar (but not identical) to the one below in Figure 2.

## 2. Preparing the board and untangling the rat's nest:

- First we must assign the proper size to the keep in boundary (the yellow box). It is important to get this right because it is the size of board that you are allotted. Components can not be placed outside the boundary or there will be no way to attach them.
  - Right-click on any keepin boundary line. It will become a dashed bold line and a shortcut menu will open.
  - Select *Properties*
  - In the *Rectangle* tab, change the units to inch
  - Leave the X1 and Y1 alone and change the width to 0.875 inch and height to 3.0 inches.

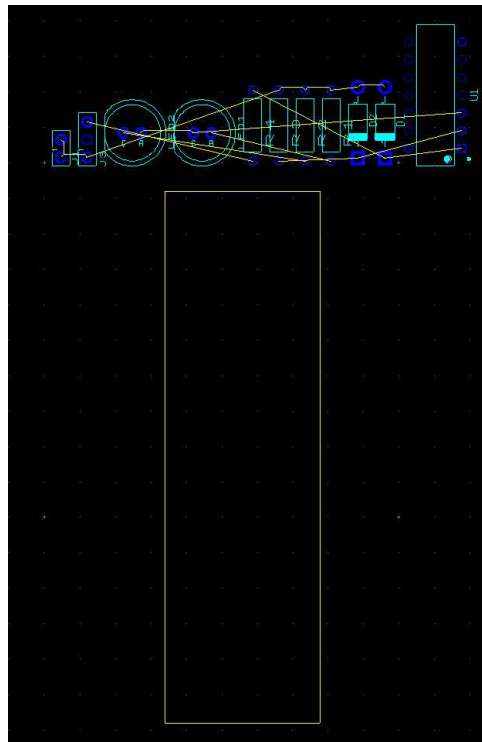


Figure 1:

- Click *OK* and it will return you to the design window and the box will now be smaller and square, as shown in Figure 2.
- Next we will untangle the rat's nest and place components on the board.
  - One by one, drag the components into the keepin box. Arrange them by putting the NAND dip package in upper left corner, the test probe at the bottom center, and the supply lines above the NAND chip in the center. Place the remaining parts on the board keeping a narrow channel in the center of the board for gripping the board. Keep in mind that the footprint shapes may not be exact so it's a good idea to leave plenty of space around each component and to not place any component too close to the edge of the keepin box.
  - Components that are connected to each other directly should be located near each other whenever possible. For example, each LED connects to a current-limiting resistor. Try to place them near each other to keep the copper trace paths as simple as possible.
  - Selected components can be rotated with **Ctrl-R**. For ease of observation it is a good idea to rotate the NAND DIP package. The other components can be rotated as needed to simplify the design.
  - In general it is best to minimize the number of crossing yellow lines. This will simplify the complexity of the copper trace line routing. You won't be able to eliminate all of them but you can reduce the crossings by repositioning and rotating components.
  - Be sure to leave some open space along one edge as seen on the right side of Figure 2. We need this space for labels in a later step.
  - If you like you can try to copy the example in Figure 2, but remember yours probably won't be an exact match so don't compare the designs by component names, instead identify parts by what they connect to. Regardless, your design will differ enough to not be able to copy it exactly, rather use the sample as a guide for where to place components in general.

### 3. Routing Traces

Once the parts are arranged on the board, each yellow connection indicator must be replaced by a copper trace line (a conductive path on the surface of the PCB) to provide the electrical connection for each node. We can do this manually but this is only possible in a reasonable time on a relatively simple board. It can be done with this board but most likely in the future you'll be designing boards that are too complex for that. Instead, we will let the software route them automatically for us.

Something to keep in mind first... the PCB is built of many separate layers. The board has a fiberglass base material (substrate) which provides the mechanical structure for the other layers. On each side of the substrate is a layer of copper which is where the conductive trace lines will be. These are sometimes called solder layers because they are typically coated with a thin layer of solder. Yes, there are two separate copper layers which is what allows trace lines to cross each other since they can run on two different planes with the substrate as an insulating layer between (more complex boards will have multiple substrates layered as insulation between additional copper layers). On top of both copper layers is a solder mask. This is the (typically) green colored layer that gives circuit boards their characteristic look. It looks like paint and insulates the copper from the outside world (except for where there are pads which are locations where parts will be soldered on). Finally, on top each solder mask there is a silkscreen layer on which we print labels, component outlines, and other useful indicators.

- Before you start routing traces, have the instructor or TA inspect your layout and approve it for the next step. This is important because it is much easier to fix layout issues now than it will be after trace lines are drawn.

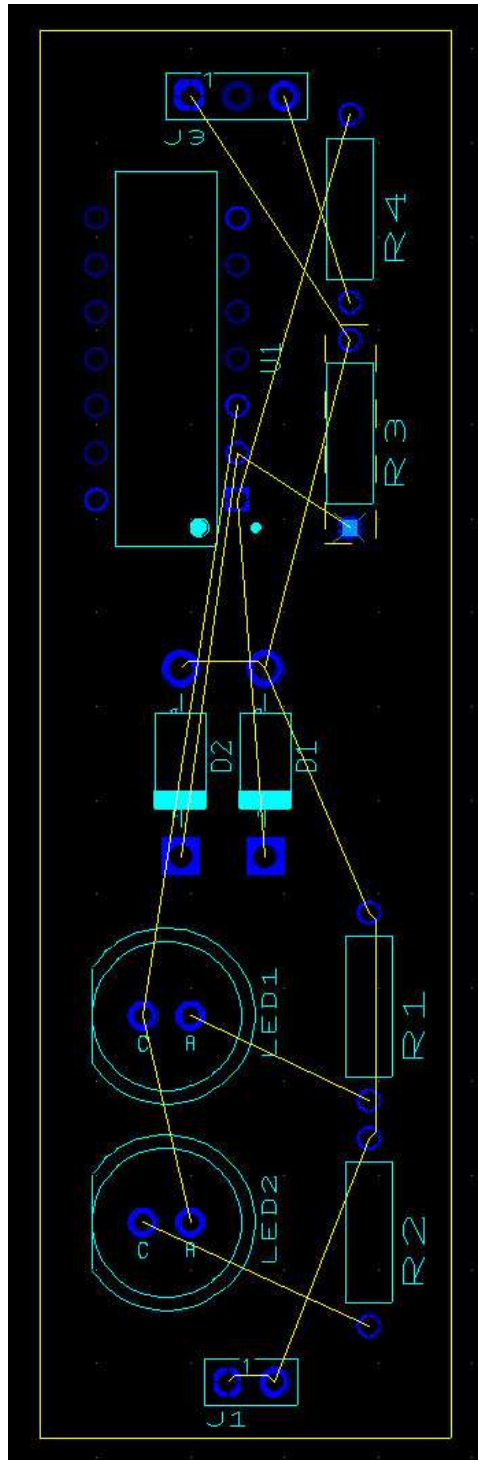


Figure 2:

- Run the traces from the supply lines to the supply pins of the NAND chip. From the Place menu, choose Line and then click the center of the header pin you are interested in. Draw a line to the appropriate pin on the NAND chip. Be aware that red circles on corners of traces are not good. You might need to angle the traces to correct these issues. Make sure that the lines and pins match up. Go back to your schematic in Multisim and verify the connections you have made.
- From the *Autoroute* menu, select *Start/Resume Autorouter*
- The screen will flash a few times and the yellow connection indicator lines will be replaced by red and green copper traces. Red lines represent the traces on one side of the board, and green lines indicate the traces on the other side. Yellow lines indicate places where the red and green traces cross over or run parallel to each other on opposite sides of the board. Every auto-route is unique so there is very little chance that your routes will look very much like the example in Figure 3.
- It is worthwhile to understand that you can re-route any traces you like. Click and drag to re-draw trace lines. They can be a little finicky so it takes some practice.

#### 4. Labeling the silkscreen layer

There are three labeling steps. First we will add some labels to identify the board itself, such as the name of the device and the designer's name (you!). Then we'll add some labels to assist the user such as the diodes and the LEDs. Finally, we will add labels for the builder so they will know what components to solder in each location. Use Figure 3 above as a guide for these steps.

- Name labels
  - Start by adding your name.
    - \* We need to be sure we're working on the "Silkscreen Top" layer, so look in the drop-down box just above the workspace and confirm that's the layer displayed. If it displays anything else, switch it before you proceed.
    - \* Click the *Place Text "A"* button on the toolbar. The Text window opens.
    - \* Type your name in the Value box. After your name, it would also be good to add something to identify the class such as "NMT - EE 101" or similar. The default size in 'normal' font should be adequate.
    - \* Click *OK* and it will put the text on the cursor for you to place. Put it on the edge of the narrow channel you left open for gripping. Right-click or hit Esc to get rid of the placer when you're done. If you want to rotate the text, you must first place it then select it and press Ctrl-R.
  - Now add a label for the name of the project. Follow the same process but this time make a "Logic Probe" label and place it near your name.
- User interface labels: Follow the same procedure as above for the following labels...
  - LEDs:
    - \* Add text labels to each LED to indicate which one is on for the HI output and which is on for the LO output. This can be determined by inspecting the traces and following one back to ground and the other back to VCC. By looking at your original schematic you can tell which one will be on in each case. YES you will have to think about this to get it right.
    - \* Text labels should say something like "HI" and "LO". This also helps you get the LED colors in the right places. Green should indicate HI and red should indicate LO.

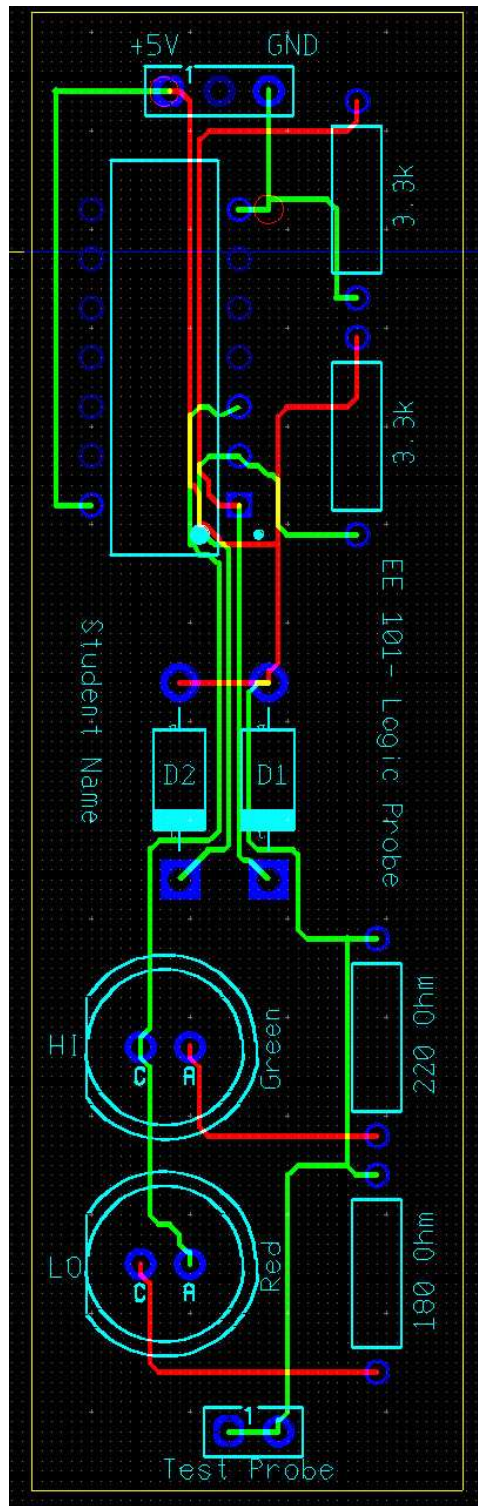


Figure 3:



- Component identifier labels:

This is a step that you would normally not need to perform. However, because the instructor will be assembling all of the students' designs onto a single master board to keep the production costs low, there is an additional complication. When the instructor pastes all of these designs into a single board, Ultiboard will not be able to tolerate more than one component of the same name. Since each student has named their components by the same names, this would normally result in multiple components of each name (R1, D1, etc.). In that case Ultiboard renames the components as more designs are inserted which would make it difficult for each student to match up the parts later when it comes time for construction. Therefore, we are going to hide the component name designators that were provided with each footprint (which would otherwise get printed on the silkscreen layer), and replace them manually with text labels. That way Ultiboard's renumbering won't impact what is displayed on the silkscreen layer.

- This is a good time to save your design, then re-save it as a new name in your incremental system. That way we can return to this point easily if something goes awry.
- Start with one component and follow the process below. Then repeat that process for each component. It is important to do this one component at a time. Do not hide all of the component names at once and then expect to remember what each one was originally named!
  - \* Double-click on the component name for a footprint (for example, start with R1). The Attribute Properties window will open.
  - \* In the *Visibility* section, select *Invisible*. This will prevent that name from printing on the silkscreen layer without affecting the name of the part.
  - \* Like you did for the other labels, place text that matches the component name. The default size should be adequate. In the example case the text should obviously say "R1". If you like you can also put the component value in the label to make it easy to place and replace parts. In the R1 case it should be the  $220\Omega$  resistor so you could label it "R1 220  $\Omega$ ". For the LEDs you could label the color, etc. Be sure to confirm the values with your schematic first!
- When you are done, save the file again.

## Finalizing the artwork

PCB creation software is generally capable of creating an industry standard set of files called Gerber files. These are universally recognized by other PCB software such that designs can be exchanged regardless of platform. Each file in the set defines the layout of each layer, plus there's another for where the holes get drilled (for inserting the components, one hole per pad). Normally you would export your project to an array of gerber files for upload to your PCB manufacturer. However, since in this case your instructor will first create a master board containing the entire group's designs, the Gerber files will not be created until after that has been done. Some day soon you'll be doing that for yourself so you'll have that to look forward to as another learning experience.

For now, you need to forward your basic Ultiboard project file to the instructor.

- First and foremost, have the instructor or TA verify that you have an adequate final design before you proceed. DO IT. Find the file in the location where you saved it. It will have a ".ewprj" extension. Be sure to get the latest version. Remember that the name of the file must contain your first and last name. If you ignored that instruction earlier, make a copy of it and rename it now.

- Attach it to an email and send it to your instructor. The subject line of the email should say “EE101 PCB” followed by your name. Yes it must say exactly that, with your name AFTER the required text. It might seem nit-picky but that’s what allows me to sort and locate the emails easily.
- Verify that the instructor has received the file before you leave. You will receive an email reply with a confirmation code. Print it and put it in your lab book. Without the instructor’s confirmation that you have completed this step you run the risk of having your project left out of the production run (and flunking a major part of the final project).
- “Being sure that you sent it” is not adequate. Really, you must verify with the instructor that the file was \*recieved\* before you are done!

As always, record all of this in your lab book, including printouts of your design at the relevant stages.