

How to make a PCB for Me

Matthew Spencer, Harvey Mudd College, 2018, rev 2

If you're reading this document, then we're probably making your first printed circuit board together, that's great! Building your first printed circuit is a milestone in your career, and I look forward to coaching you through it.

This document collects bits of wisdom, best practices and personal preferences that I often impart to students during design reviews, so following the instructions in the document should help us speed up the design process. This document is not intended to replace documentation or tutorials for specific CAD software, so don't look here for the details of what menus to click on.

Designing a printed circuit board proceeds in four stages: bill of materials / library generation, schematic, layout, and manufacturing. We'll discuss each below.

Bill of Materials

Building a circuit board begins by selecting components that will appear on the circuit board. You need to write down which components you've selected by recording them into a spreadsheet: this will help you keep track of the components and put together a parts order later.

The spreadsheet should have the following columns in this order:

- **Description** – a short name for the part. Eg: “bus transceivers for motors,” or “right angle connectors for daughter board.” “Resistors” is a weak description because it doesn't help a future reader of this document to figure out why those resistors were purchased.
- **Reference Designator** (abbreviated Ref Des or Ref) – the name of the part on the schematic.
- **Datasheet link** – a link to the part's datasheet. Also download the datasheet and keep it in a folder with the rest of your PCB project in case the link changes. Try not to rely on datasheet aggregators like Alldatasheets, instead prefer using the vendor's/manufacturer's website.
- **Manufacturer Part Number** (Mfg PN) – The part number assigned by the manufacturer
- **Vendor** – Where are you buying it? Be sure to check that the part is in stock. Lead times on restocking parts are often several weeks, which is too long for the projects we do at Mudd.
- **Vendor part number** (Vendor PN) – vendors use different catalog numbers than manufacturers, so recording this number ensures you can find the part again.
- **Vendor description** – descriptive text assigned by the vendor. Used to search for similar parts.
- **Unit cost** – the cost per unit specified by the vendor. It's OK to use the cost for 1 unit even if you'll actually be buying in larger quantities and getting better prices.
- **Quantity** – how many do you need for one board? If you are buying a component that will be split over many boards (e.g.: a big piece of header you plan to singulate) then this quantity might be less than one.
- **Total line cost** – the product of the unit cost and the quantity.
- **Vendor link** – a link to the website page where you would purchase the product.
- **Notes** – Additional notes about the part: why you're buying this, labels that need to be on the parts rack, alternative parts numbers or vendors, etc.

Sometimes bills of materials will call for additional information. For instance, if you are providing a bill of materials to an assembly house then you might need to provide a column describing the footprint of a component, a column containing its X location and a column containing its Y location. It could also be helpful to include the layer on which a component resides (top/bot). Include these supplemental columns after the notes column on your master BOM, and reformat it for different vendors.

Many PCB CAD programs will generate a bill of materials for you based on a schematic you've entered. I've never had this work well enough that I was comfortable skipping the step of manually keeping this spreadsheet. Further, you often need the bill of materials in order to generate the schematic.

Notes on Component Selection

Be sure to select components with convenient packages. The package type of a component refers to its shape and arrangement of the component's pins.

One way to classify packages is by asking if they are designed to be surface mounted on a PCB (attached only to flat pads of copper on one side of the board) or to be through-hole soldered (attached to holes drilled all the way through the PCB). Through-hole components generally require less sophisticated techniques to attach to your board, but also require more manufacturing time and more board space.

DIP-x and SIP-x are common through-hole packages; DIP stands for dual inline package and x is the number of pins, SIP stands for single inline package. TO-220 is another common through-hole package. The chips that you have used in breadboards are generally DIP packages.

TSSOP, SOIC, QFN and BGA are common surface mount packages. SOIC and TSSOP packages are generally easy to assemble at Mudd. QFN packages are difficult to assemble at Mudd and BGA are impossible: you should probably use an assembler if you are using either QFN or BGA packages. Surface mount resistors come in packages like 0402, 0603 and 0805, which specify the resistor's dimensions in tens of mils: each mil is 1/1000 of an inch, so 0603 refers to a package that is 60 mils by 30 mils. I recommend 0805 packages for assembly at Mudd. It is possible to assemble 0603 or 0402 components by hand, but it is time consuming to work with 0603 and very difficult to work with 0402.

Assemblers charge more for through-hole components than they do for surface mount components, so consider using mostly surface mount components on a board that you plan to have assembled.

Making a Library

After selecting your parts you need to make sure that they can be represented in your PCB CAD software. This representation is captured in a component library. A component library is a collection of components, and components are containers which hold relevant information about a part. The two main pieces of information a component holds are the part's symbol, which is how the part is represented in a schematic, and the part's footprint, which are the physical dimensions of a part that appear on a layout. A component might also contain information about naming conventions for a part, the purposes of each pin (input/output/power/etc.), vendor information, and ways the part should be labeled in the silkscreen on a board.

Make a component for each part you've selected in your bill of materials. If you are using KiCAD, be sure these components are packaged with your project so that someone else can use the project later.

PCB software often contains common symbols and footprints which can be combined to make components quickly. Use the default footprints from the PCB software where it makes your life convenient.

Schematic

Drawing a schematic is a way of codifying how you want your components to be connected together. This is accomplished by populating a document called a schematic with the symbols you have generated in your library and connecting them with imaginary wires which are referred to as nets. This step is kept separate from layout because it is easier to express your intent clearly without also including all of the physical details of how things are connected. The schematic is also used to check the layout for errors. As a result, your major goals for your schematic should be that it includes every detail of your board, that it accurately conveys your design intent, and that it is correct.

Including Every Detail

Including every detail of your board is more work than you might expect. You need to be sure to include bypass capacitors and other supporting passives on your schematic. You also need to include every connector that you hope to use. This can be particularly subtle if you are using unusual connections or metal patterns, like unusual ground planes or grounded screw holes. Those metal structures will need some kind of symbol to represent them, and you need to think carefully about how many terminals that symbol will need. This thoroughness pays dividends because it forces you to consider every detail of your design: it's hard to cut a corner when it leaves you with an unconnected wire in your schematic.

In keeping with this practice: unconnected pins (called no connects) and test points (convenient via holes or metal stubs) should be explicitly called out in your schematic so that users know they are intentional. You may need to introduce special symbols (with corresponding empty footprints) into your library for these unusual features.

Expressing Design Intent

Expressing your design intent clearly requires some considerations of graphical design. A designer who has never seen your project before should be able to look at your schematic and understand it at a glance. This is analogous to a beautifully written piece of code: understandable code has comments (and so should your schematics), but the variable names and the operations on each line should also clearly convey what is happening.

The most simple way to express intent is to separate different circuits or functional blocks into different parts of the schematic. For instance, you might place amplifiers and the associated resistors and bypass capacitors in the lower left corner of a schematic, while a microcontroller and associated components occupy the upper right. You can also split the schematic over multiple pages if your first page becomes cluttered.

Within these groupings, place your symbols so that they suggest the purpose of the circuit. For instance, place the input of an amplifier chain on the left of the schematic, and then connect the outputs of each amplifier to the input of the next so that signal flow in the chain moves from left to right. Keep power at the top of the schematic and ground at the bottom. You may break these rules if it greatly simplifies your schematic, but do so with care.

Connections from one group to another, and complex connections within a group, should be made with symbolic connections. A symbolic connection is usually created by placing a label on a net. If you place the same label on another net then those nets are considered connected even though there is no explicit line drawn between them. This is extremely helpful to reduce clutter in a schematic.

If you must draw crossing wires, note that a dot indicates that the wires intersect and an absent dot indicates that they do not. Often dots are omitted in three-way junctions because it is rare to have a dangling wire in the middle of your schematic. However, the easiest way to avoid confusion is to not draw crossing wires: replace them with symbolic connections where necessary.

Every net should have a label even if it's not symbolically connected: this will help with debugging later. Various CAD rule checkers will refer to nodes by difficult-to-decipher automatically generated names if you don't provide labels.

These labels can be great assets in expressing design intent if you follow a standard convention. I recommend the following:

- All node labels are composed of a capital name and an optional lower case suffix. For instance, you might use the name VDD33 for a 3.3V rail or the name RESETb, which has a suffix of a lower case b, to indicate an active low reset signal.
- The suffix b is short for "bar" and indicates logical inversion of a signal or that a signal is active low.
- The suffixes p and n indicate that the signal is part of a matched differential pair; a differential input might be on the nets INp and INn.
- Power supplies should have VDD in their names, analog supplies may have the special label AVDD. Ground nodes should have GND (or AGND) in their names.
- Analog signals should include the name of the previous stage in the amplification chain (GAIN1OUT or FILTER1OUT).
- GPIO signals should be labeled with a signal direction, a purpose and often a bus number (ADC_IN_8, PWM_OUT_30, or IO_15). If your program supports it, I like to wrap bus numbers in angle brackets (INPUT<8>).
- Communication protocols should be labeled according to common conventions RX/TX for UART, MOSI/MISO/SCK/CSb for SPI, and SCL/SDA for I2C.
- FPGA pin labels require special care: they should be labeled both with pin information about the FPGA and with the signal they are intended to connect to in the pin map.
- If a pin can have multiple functions, as in many microcontrollers, then only label it with the function that is being used in this design.

Similarly, components should be named according to a convention. These names are referred to as reference designators, and are used to refer to the components in the bill of materials and the layout. The labeling convention demands components be labeled with a type label followed by a number that the CAD software will automatically generate. For instance, you might have resistors R1 to R5 in a design. Some special components are deserving of additional labels to indicate their function; e.g.: I have often labeled feedback resistors as Rf. Be careful that these special names don't ultimately make more work for the assembler or the user, and be sure the silk screen reflects them; I mostly find these

labels useful for instructional purposes so that a user can infer the purpose of a resistor just by looking at the silk. A table of type labels for common components is below.

Component	Ref. Des. Prefix
Resistor	R
Capacitor	C
Bypass Capacitor	XC
BJT	Q
MOSFET	M
Inductor	L
Ferrite bead or choke	F
Integrated Circuit	U
Test Point	TP
No Connect	NC
Drill Hole	H
Jumper	J
Switch	SW
Crystal	X or Y

Table 1: Mapping from component type to reference designator prefix.

Ensuring the Schematic is Correct

Once you have a well-drawn schematic you should use all available tools to check that it is correct. The tool used to do this is called an electrical rules check or ERC. This check compares the stated purpose of each symbol pin (which is specified in the component library) to the purpose of the pins to which it is connected. For instance, the ERC might check that output pins should be connected only to input pins and issue warnings if it detects that two outputs will be driving the same net.

The warnings issued by the ERC can be cryptic and frustrating, but they are your friends. You should resolve every error you encounter until you have a clean ERC. This may involve editing your library to change the function of pins on your symbols (e.g.: "I'm really using this power pin as an input" or "this resistor is an InputOutput because it has no direction") or you may edit the wires on your schematic to correct errors or design or intent.

Another way to ensure that your design is correct is by copying it from available resources. Datasheets, application notes and development boards often contain reference designs that are endorsed by the chip vendor. If you don't use a reference design, then find a way to prototype your circuit before committing it to a PCB.

Layout

After specifying how you would like to connect components with a schematic, you need to specify physical details of how they will be connected by creating a layout. Layouts express where every scrap of metal is located on a PCB, and there are many details about how that metal is arranged that can have a big impact on the function of your board.

Stackups

Layouts specify the XY positions of sheets of metal on the large flat faces of the PCB, but the arrangement of the thin vertical Z dimension of the board has a big impact on how the layout will proceed. You need to specify the Z arrangement, called the stackup, of a board before you begin layout in earnest. Specifying a stackup involves picking the number of copper layers in your board (usually 2 or 4 layers for boards that we make at Mudd), the thickness of each layer, the separation between layers, and the intended purpose of each layer. Circuit fabrication vendors usually have already designed stackups for 2 or 4 layer boards. Unless you're doing something highly specialized, you will choose from amongst these pre-generated stackups, so you don't often need to meditate on the copper weight of your layers or the number of fiberglass sheets separating them.

The purpose of a copper layer refers to the type of signals which flow on the copper layer. There are three basic layer types: signal, power and ground. Power and ground layers are often designed so that there is a single large sheet of copper on the layer rather than individual copper traces. This reduces the impedance of the layer so that power can be delivered with less loss. Layers designed in this way are referred to as power or ground planes. Having good grounds is so important that ground is usually given an entire plane, but it's common to split power planes so that two or more supply voltages occupy different rectangular regions of the layer. It is possible to add signal traces to power or ground planes by cutting out copper around the trace, but internal routing is difficult to rework if there are mistakes and increases the impedance of the planes.

Signal layers are usually assigned a routing direction -- vertical or horizontal -- with the intent that long routes on that layer will travel in the assigned routing direction. This minimizes the chance the wires will collide with one another on the same layer, which helps ease routing congestion. Because wires need to run both vertically and horizontally, collisions are almost inevitable if you have only one routing layer unless the board is very simple.

Unused area on signal routing layers is usually filled with metal which is connected to ground. This increases the capacitance of all of the traces on the board, which can be desirable for power traces, especially on two layer boards. This metal fill is referred to as a flood or a pour.

There are significant benefits to having your ground plane near your most active signals, and boards are easier to assemble if components are only on one side. Consequently, the stackup for four layer boards at Mudd is usually Top Copper (routing and components) – Ground – Power – Bottom Copper (routing). Two layer boards usually have signal routing on both top and bottom copper, but the bottom is treated as a "pseudo-ground-plane" by putting relatively few traces on bottom copper and using most of the space for a large ground pour.

Placement and Routing

With the stackup decided it's time to place components and draw the traces. Components footprints and the intended connections between them should be automatically imported from your schematic. The footprints will be represented by collections of metal pins, and lines denoting which pins need to be connected will be drawn between components. These lines are often dense and confusing at the start of the routing process, so they are referred to as a ratsnest.

The first step in drawing is to make a board outline. Some CAD packages automatically generate an outline. Unless you have particularly good reasons, pick board outline dimensions that are easy to work with and which have a small number of significant figures. For instance, 3" is a great board dimension, 4.5" or 2.25" are fine dimensions, 1.2" or 2.125" are OK, and 2.23" is an annoying choice. Put one corner of your board, usually the lower left, at the 0,0 coordinate in the CAD program.

You need to select the units in which dimensions on the layout will be recorded. Usually CAD programs offer a choice between mils and millimeters. US vendors mostly specify their manufacturing capabilities in mils, so use mils unless you have a good reason not to do so. Good reasons not to use mils might include copying from a reference design that is specified in mm. using a part which has a datasheet specified in mm, working with an international vendor, or working with a client that uses mm in their internal designs.

The next step is arranging the footprints on your board outline. This step is called placement. Placement has a huge impact on the complexity of your routing, so it should be considered very carefully. For instance, if you need to route a bus of 8 wires, you might consider lining the two chips connected to the bus such that those 8 wires are all short, straight lines. A simple and easy to interpret ratsnest is a sign that your placement efforts have been successful.

I find that I'm most successful in placement if I sketch out the routing on a sheet of paper before I begin placement. This process of pre-placement is referred to as floor planning. I strongly recommend using a sheet of paper to do some basic floor planning: please sketch some preliminary placement and routing plans before you begin placement on the computer.

Try to pack the components close together during placement, but leave enough room for routing, clearance around connectors and switches so that fingers will fit, and enough space between components that you can debug or rework the board if necessary. Your floor plan will help you figure out your placement density requirements.

After the components are placed it is time to replace the ratsnest with actual routed pieces of metal. There are programs that can handle this process called auto-routers, but I find that they are significantly less efficient than human routing. Further, driving an auto-router requires expertise and several trials, so I find that they don't save much time. Route your board manually unless it is extremely simple; I only auto-route breakout boards that consist of 16 straight wires between two connectors.

While laying down metal routes you need to decide on their thickness. Your PCB manufacturing vendor has the final say on the minimum thickness that can be created on your board because their process has some finite manufacturing resolution and tolerance. However, a common minimum trace width is 6mils with 6 mil clearance to adjacent traces or to vias (drill holes used to connect different layers of copper). I prefer traces to be much larger than this six mil minimum: ~ 15 mils for signal traces, ~30 mils for power and at least 50 mils for high power traces like battery supply lines or motor output wires. It's usually OK for vias to remain at their default size (though note that the PCB CAD program called PADS makes them unusually large by default), but if you need a low impedance connection between layers you will need to place many vias in parallel.

Design for Convenient Assembly and Use

Make the pads of your footprints larger than the pins of the chips. Some stock footprints come with very small pads, which allows for dense layouts. You should explicitly sacrifice a little density for ease of assembly and rework.

If you are unsure of whether a component will work, lay out the board such that the component can be removed and the traces it is attached to can be accessed. This allows you to cobble together a replacement for the failed chip off the board and to prototype with the rest of the board intact. Zero ohm resistors can help you to add these “contingency plan connections” in high frequency designs.

Add test points to access hard to reach or interesting signals. These test points could be vias that you can probe or footprints for test loops that you intend to solder to the board. Test points should also be reflected in the schematic, so be sure to add them there and then import them into the layout.

Sometimes debugging or repair requires you to cut a trace on a PCB. You can only cut traces on top copper or bottom copper, so putting uncertain routing on inner signal layers is very risky.

Add standoff holes to the corners of the board so that it can be mounted in a mechanical chassis or just supported off the surface of the table. Ground the standoff holes so that the board will be grounded to the chassis. This has the added benefit of sharing ground through the standoffs if you make a multi-board stack.

The single most important feature determining if your board is easy to use is the quality of the silk screen, a layer of printed text on the outside of the board. Use the silk screen to add documentation to every part of your board. If you do a good job with silk screen, a user will never need to go back to the schematic or layout to figure out what is going on. It is totally reasonable to sink significant time into designing your silk screen; it is totally unreasonable to skimp on silk screen design. Don't make the mistake of thinking that silk is irrelevant just because it isn't carrying current between chips.

Verification

Just like schematics, you want your layout to be correct. There are two major checks that help you identify that your layout is correct: the design rule check (DRC) and the layout vs. schematic check (LVS).

LVS verifies that the connections on your layout are the same as those on your schematic. It will catch unintended shorts, unconnected wires and misrouted traces. This is an important check: obviously you want your layout to reflect the design you intended to make.

DRC checks the arrangements of metal and drill holes on the board to ensure that they match the capabilities of the PCB manufacturer. For instance, it checks that there is clearance around all drill holes, that traces are separated by a large enough distance, and that no trace is too small. Often these rules can be imported from files supplied by the PCB manufacturer.

You must correct all warnings in DRC and LVS, the tools are there to catch your mistakes.

FreeDFM is an online DRC and design for manufacturability (DFM) checker provided by a PCB vendor called Advanced Circuits. You should subject your board to a FreeDFM check after you pass it through DRC and LVS.

Reference designs exist for layouts in datasheets and application notes just like they do for schematics. If you are designing a layout sensitive component like a switching voltage regulator, an antenna, or any very high speed circuitry, then do your best to copy a layout that is known to work from a design reference.

Manufacturing

Before sending a layout out for manufacturing you must bring it to me for a design review. Be prepared to answer detailed questions about the schematic and to defend your layout choices. My goal is to catch mistakes before we ship. I usually look especially closely at trace widths, high current paths, power routing, and silkscreen. You will probably need to make revisions after the design review, which means you will have to do a second design review so I can look at the changes. After ~2 revisions I usually agree that a board is ready to ship.

You send a board to a manufacturer by asking your PCB CAD software to export tool control files called Gerbers and drill files. Export your Gerbers and drill files in the format specified by your PCB vendor. Usually that means exporting Gerbers in the RS-274X format and drill files in the Excellon format. Bundle the exported files into a ZIP file so that they can be easily uploaded to a PCB vendor's site.

You want to verify that your design was correctly exported to Gerber files before you ship your board. You need to use a 3rd party Gerber viewer to verify that your Gerber files match your expectations. I like Gerbv as my 3rd party viewer, but there are online Gerber viewers as well. For instance, the PCB vendor OshPark also has a charming visualization of your board if you upload your Gerbers to that site.

There are two PCB vendors that we usually use at Mudd. OshPark is a PCB aggregator that offers good price and turn time for small numbers of small 2 layer boards with the super swift option. Their regular option is too slow for most Mudd projects, so always opt for super swift fabrication. If you need to make a large number of 2 layer boards, large two layer boards or any number of 4 layer boards then Advanced Circuits will give you the better price. The cost advantage for Advanced Circuits is largely due to special pricing deals the company offers for small runs of boards: you can get four layer boards for \$66/board and two layer boards for \$33/board, so be sure to leverage those options when you use advanced circuits for small runs. If you need very simple boards with a very short turn time, then consider the Bare Bones special pricing that Advanced Circuits offers: you don't get silk or a solder mask, but you can get 1-2 day turn for reasonable prices or a 5 day turn very cheap. Advanced Circuits is a giant in the industry, so it has a wide assortment of PCB fabrication options if you need a specialized board.

Students are often tempted to manufacture their boards at Mudd using a PCB mill or by etching their own boards. I've never seen students etch their own boards, usually because the chemical safety protocols required to get the etch working are pretty onerous. When students try to use a PCB mill they usually spend three weeks trying to figure out how the tool works, then throw up their hands and go to a PCB vendor at the end. I recommend just using a vendor instead of attempting in-house manufacturing. Extremely simple boards, like patch antennas, are an exception to this rule. I've seen those manufactured gracefully in the machine shop with a combination of carbide bits and attention to safety details.

After your manufacturer receives your Gerber files they will make the boards and ship them back to Mudd. While you are waiting for your boards, you should purchase all of the components on your bill of materials. The quantity in your bill of materials reflects the number of parts needed for one board, so you will have to use your bill of materials to put together an ordering spreadsheet which reflects the actual number of parts you want to buy. You should buy more than enough to populate all of your boards: I recommend buying the number of parts needed to populate and an additional 10% overage. This overage allows you to break a few parts in handling and still succeed in manufacturing all of your boards.

If you need an assembler, you should also make arrangements with the assembler before the boards and parts arrive at Mudd. Consulting an assembler before ordering the boards is wise: some assembly houses require special design features on boards so that their machinery will operate. (One common example is fiducials, alignment marks that help assembly machines see where boards are.) I like to work with Sierra Assembly Technology, which is located in Corona, for my assembly jobs, but there are a few other local vendors (Action Assembly, for instance). Take/ship the boards and the components from the BOM to the assembler when both arrive, and pick up completed boards when the assembler is finished. Relying on local assemblers usually saves you two days of waiting for shipping and the cost of two overnight shipments.