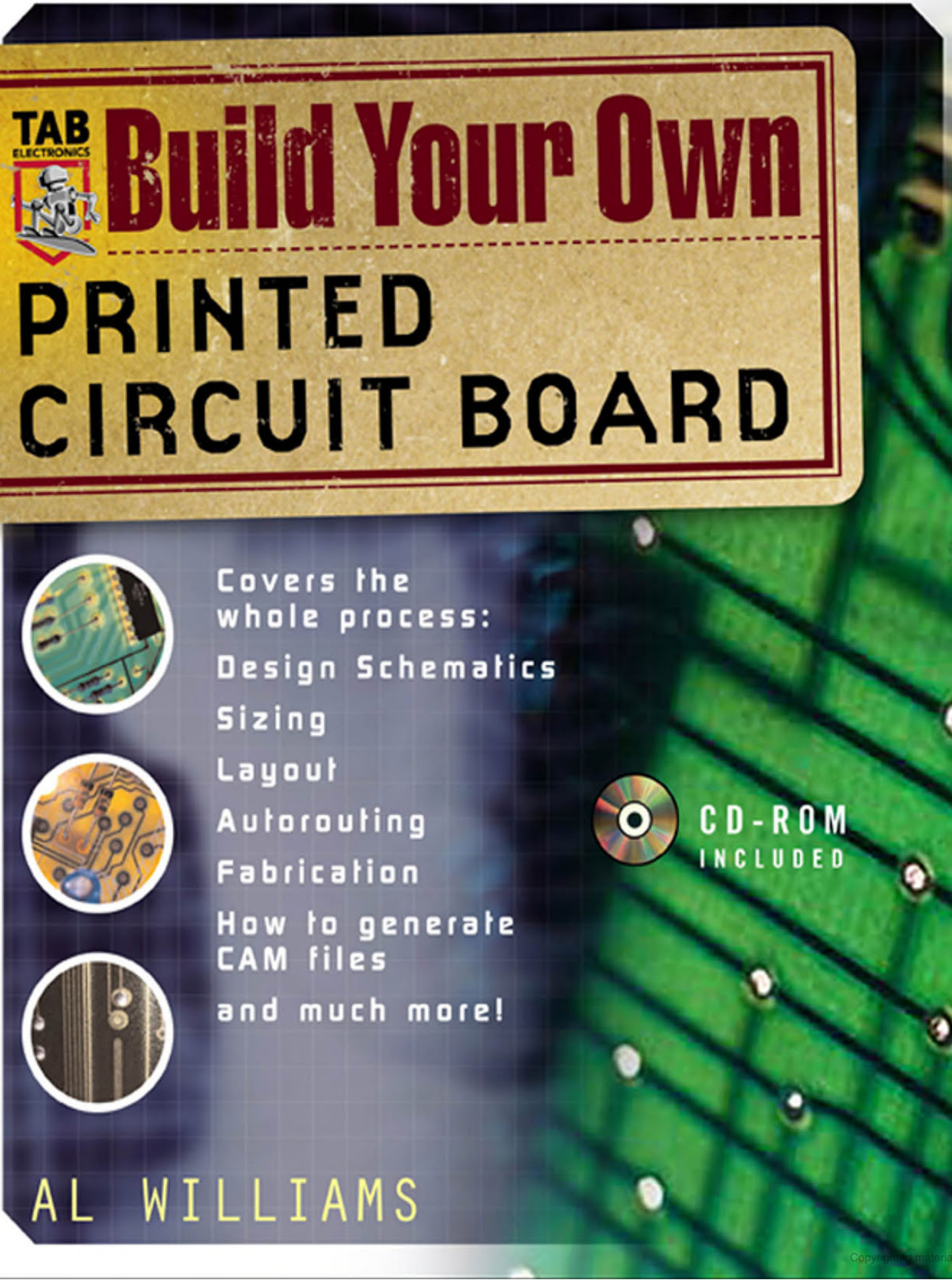

TAB ELECTRONICS

Build Your Own
PRINTED CIRCUIT BOARD

Covers the whole process:

Design Schematics

Sizing

Layout

Autorouting

Fabrication

How to generate CAM files

and much more!

CD-ROM INCLUDED

AL WILLIAMS

Build Your Own Printed Circuit Board

Al Williams

McGraw-Hill
New York Chicago San Francisco Lisbon
London Madrid Mexico City Milan New Delhi
San Juan Seoul Singapore Sydney Toronto

The McGraw-Hill Companies

Copyright © 2004 by The McGraw-Hill Companies, Inc. All rights reserved. Except as permitted under the United States Copyright Act of 1976, no part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

ISBN: 978-0-07-170905-7

MHID: 0-07-170905-3

The material in this eBook also appears in the print version of this title: ISBN: 978-0-07-142783-8,
MHID: 0-07-142783-X.

All trademarks are trademarks of their respective owners. Rather than put a trademark symbol after every occurrence of a trademarked name, we use names in an editorial fashion only, and to the benefit of the trademark owner, with no intention of infringement of the trademark. Where such designations appear in this book, they have been printed with initial caps.

McGraw-Hill eBooks are available at special quantity discounts to use as premiums and sales promotions, or for use in corporate training programs. To contact a representative please e-mail us at bulksales@mcgraw-hill.com.

Information contained in this work has been obtained by The McGraw-Hill Companies, Inc. (“McGraw-Hill”) from sources believed to be reliable. However, neither McGraw-Hill nor its authors guarantee the accuracy or completeness of any information published herein, and neither McGraw-Hill nor its authors shall be responsible for any errors, omissions, or damages arising out of use of this information. This work is published with the understanding that McGraw-Hill and its authors are supplying information but are not attempting to render engineering or other professional services. If such services are required, the assistance of an appropriate professional should be sought.

TERMS OF USE

This is a copyrighted work and The McGraw-Hill Companies, Inc. (“McGrawHill”) and its licensors reserve all rights in and to the work. Use of this work is subject to these terms. Except as permitted under the Copyright Act of 1976 and the right to store and retrieve one copy of the work, you may not decompile, disassemble, reverse engineer, reproduce, modify, create derivative works based upon, transmit, distribute, disseminate, sell, publish or sublicense the work or any part of it without McGraw-Hill’s prior consent. You may use the work for your own noncommercial and personal use; any other use of the work is strictly prohibited. Your right to use the work may be terminated if you fail to comply with these terms.

THE WORK IS PROVIDED “AS IS.” McGRAW-HILL AND ITS LICENSORS MAKE NO GUARANTEES OR WARRANTIES AS TO THE ACCURACY, ADEQUACY OR COMPLETENESS OF OR RESULTS TO BE OBTAINED FROM USING THE WORK, INCLUDING ANY INFORMATION THAT CAN BE ACCESSED THROUGH THE WORK VIA HYPERLINK OR OTHERWISE, AND EXPRESSLY DISCLAIM ANY WARRANTY, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. McGraw-Hill and its licensors do not warrant or guarantee that the functions contained in the work will meet your requirements or that its operation will be uninterrupted or error free. Neither McGraw-Hill nor its licensors shall be liable to you or anyone else for any inaccuracy, error or omission, regardless of cause, in the work or for any damages resulting therefrom. McGraw-Hill has no responsibility for the content of any information accessed through the work. Under no circumstances shall McGraw-Hill and/or its licensors be liable for any indirect, incidental, special, punitive, consequential or similar damages that result from the use of or inability to use the work, even if any of them has been advised of the possibility of such damages. This limitation of liability shall apply to any claim or cause whatsoever whether such claim or cause arises in contract, tort or otherwise.

Copyrighted material

CONTENTS

ACKNOWLEDGMENTS

While I was writing this book, the United States, Great Britain, and a few other countries invaded Iraq. Regardless of what you thought about the war, politically and morally, you could not deny that the video sent by satellite by embedded reporters was astounding. Watching live footage of warfare gave most of us the closest taste we will ever have (hopefully) of the sheer terror the young men and women on both sides face.

What I found most interesting, though, is that the front line of an army is like the point of a knife. Behind every solider there is an enormous support apparatus that isn't apparent from the news video. Quartermasters and other solders provide the front-line solder with food, clothing, ammunition, fuel, and intelligence among other things. Even the payroll clerk that handles the soldier's pay is vitally important. But you rarely, if ever, see any of these people who make the front line job possible on the news channels.

Books are much the same way. My name may be on the cover, but in fact, I'm just the tip of knife. The book you hold in your hand is possible only because of a tremendous infrastructure that includes editors, typesetters, printers, publishers, and marketing staff. This team—including Judy Bass and Scott Grillo at McGraw-Hill and Beth Brown and her team at MacAllister Publishing Services—made this book, as you see it, possible and they have my gratitude for it.

I'd also like to thank the folks at CadSoft (especially Klaus Schmidinger) for creating Eagle and allowing us to distribute it with the book.

My family—which includes my wife Pat, three kids, three grandkids, two dogs, and a cat—also have my thanks. Not only did they put up with another book, but they also didn't complain too much about the ferric chloride stains on the patio!

This page intentionally left blank

CHAPTER 1

Printed Circuit Fundamentals

In 1798, Eli Whitney secured a government contract to produce 10,000 muskets for $130,000. Many didn't believe he could do it because producing guns at that time required skilled gunsmiths. Also, because each gun was slightly different, parts were not interchangeable. What Whitney envisioned, and successfully implemented, was a system where precise machine tools would produce individual parts of exact sizes and shapes. Operated by relatively unskilled labor, these machines would produce parts that could be assembled into many identical rifles. Other industrialists—Henry Ford, Cyrus McCormack, and Isaac Singer (of sewing machine fame)—were quick to follow suit, and the result was the Industrial Revolution.

The field of electronics has had several similar revolutions. If you looked into an old-fashioned radio receiver, the first things you'd notice would probably be the tubes. But, after the tubes, you'd probably notice the mess of wires connecting all the components together. Someone had to physically connect those wires and solder them. Of course, that means mistakes happen—you can't expect people to build radios without making errors. The more complicated the project, the more mistakes you can expect.

In the 1950s, companies started using *printed circuit boards* (PCBs) to reduce costs and errors. The idea is quite simple. You start with an insulating substrate (the board) made of phenolic (a material made from paper and resin), fiberglass, or ceramic. Then you arrange for copper to be placed on the board in a particular pattern. This copper, which is bound to the substrate, conducts the electricity just like wires. Holes are drilled to allow components to mount to the board.

Offhand, this doesn't sound much better than wires. The trick is that automated processes can create the copper patterns on the board easily. Just as Whitney's machines turned out perfect rifle barrels, it is possible for modern manufactures to mass-produce perfect circuit boards. Of course, the trick is how to actually do this. Several possible methods are available. You can take a board that already has copper laminated to it and remove the copper you don't want. This is usually done with a photographic process and chemicals that attack copper. However, it can also be done with a computer-operated milling machine that mechanically chips away the copper you don't want.

Other ways to build PCBs exist. For example, you can start with a blank substrate board and electroplate the copper on the board. Some PCBs are not made photographically, but are made with a silk screen.

No matter what the method, the key is that, armed with some negatives (or a milling machine program), you can produce dozens or hundreds or even tens of thousands of boards, all precisely the same. The cost of producing these boards is relatively inexpensive, and there can be other advantages as well. For example, with precisely placed components, it is possible to use machines to insert the components into the board. Another machine can solder the parts, practically eliminating human involvement in the production process.

When PCBs first became widespread, producing them was high technology indeed. First, artwork (the pattern that shows where the copper should be on the board) was done by hand. Because the tolerances were important, designers typically used drawings that were four times larger than life size. Then a special camera would take a picture of the artwork and reduce it. The resulting film would then be used in a complex manufacturing process.

Several things have happened to make PCBs more accessible than ever to the small company or hobbyist. First, computer programs are available that enable you to design PCB layouts easily. As with anything you do on the computer, you can correct errors, experiment with different layouts, and make changes very easily. You can even have the computer do a great deal of the work for you.

The industry has a standardized format for describing PCB layouts. When you are satisfied with your design, you can simply e-mail your files to any number of companies that will build the board to your specifications and ship it back to you. If you are willing to pay enough, you can design a board on Monday and have it in your hands by Wednesday.

However, it is also possible to build your own boards with a modest investment in tools and chemicals. You can create film with a common computer printer. You can also use a laser printer (or copier) to directly transfer the pattern to the PCB material. Then several inexpensive and relatively safe chemical processes can be used to finish the job.

Like the gun, what was once the province of specialized labor is now accessible to anyone who wants to build electronic circuits. That's a good thing, too. Modern circuits are getting harder and harder to build by hand. Working with normal integrated circuits is hard enough. With the trend toward tiny surface-mounted parts, PCBs are almost a necessity, even for prototyping or one-off projects.

A PCB

Figure 1-1 shows a typical PCB. Like most boards these days, it has multiple layers of copper (this board has two layers). A homemade board is easier to make with a single layer, but when having boards made commercially, two layers are practically as inexpensive as a single layer. (Because you can

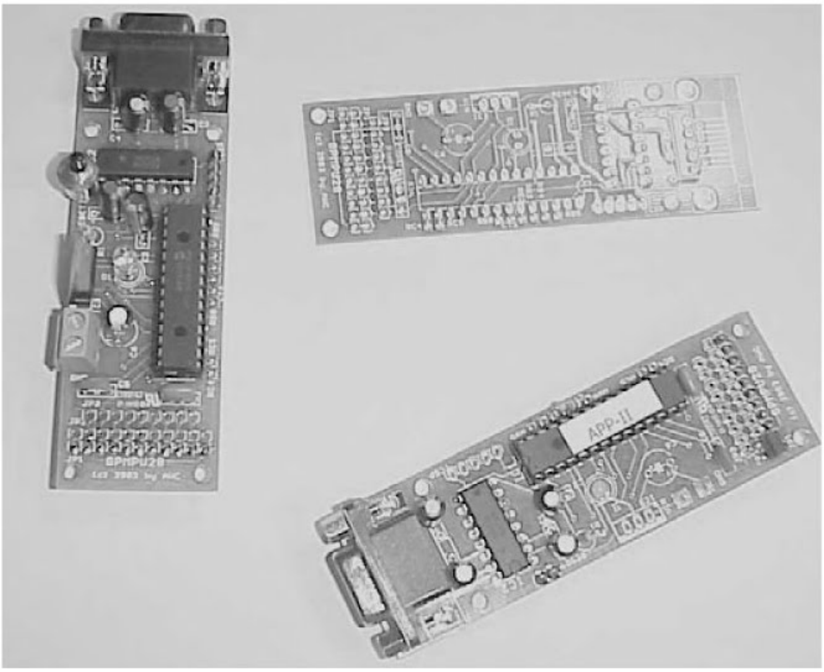

Figure 1-1 Boards made by a professional board manufacturer

almost always make a two-layer board smaller than one with a single layer, two-layer boards are typically less expensive than a single-layer board to produce.)

In fact, designing a board like one in the photograph requires more than just two layers. This board has six different design layers:

- The bottom copper layer
- The top copper layer
- The markings on the top side that identify the components
- A mask layer for the bottom
- A mask layer for the top
- A layer that defines where the holes are drilled

The mask layers are not visible, but they define where a special chemical coating prevents solder adhesion to the board. This makes it much easier to build the board by hand and is essential for automated soldering equipment.

Even the innocent-looking holes in the board are special. First, they must be precisely placed so that the components will fit. Integrated circuits are especially finicky about hole alignment. Also, on a two-layer board, the holes are plated internally with copper so that the top and bottom of the board are electrically connected at every hole. Most of the holes are for fitting components, but some exist just to connect a copper line (known as a track or a trace) on the top to one on the bottom. These special holes are known as *vias*. Plated through holes also increase the board's strength because it is harder to delaminate the copper from the substrate when the copper is attached on both sides.

Because this is a professionally produced board, it also has a tin or solder plating on top of all the copper. This makes the copper easier to solder and protects the copper against corrosion.

Homemade boards, like the one in Figure 1-2, tend to be simpler. This board has a single copper layer, no component markings or masks, and no plating. Because it was drilled by hand, the holes are not as precisely placed as the other board.

Typical Design Flow

If you don't think too hard about it, it seems that laying out a PC board would be easy. The problem is that you can't simply put components anywhere you want on the board. Of course, electrical issues must be considered. For example, a crystal or bypass capacitor might need to be close to its associated integrated circuit. However, the most vexing problem with layout is that copper traces on the same side of the board must not cross (unless they are supposed to connect, of course). Unlike wires, a copper track cannot be laid on top of another—except by placing it in another layer.

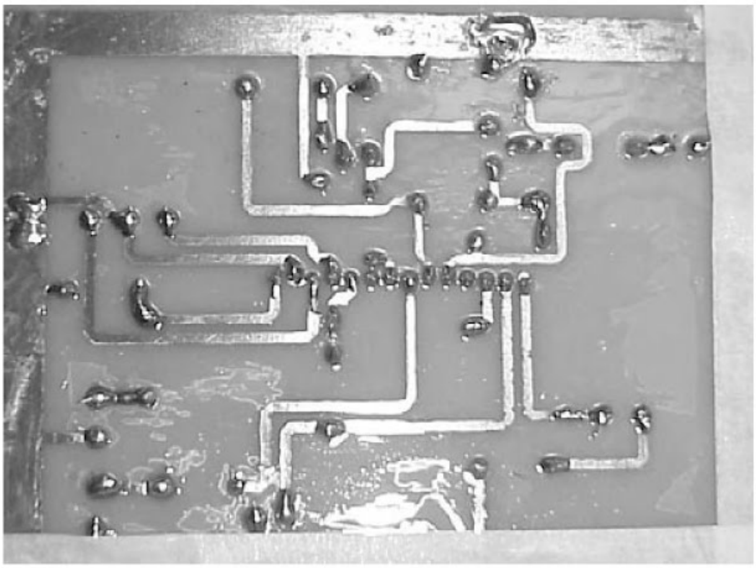

Figure 1-2
A homemade PC board covered with a coating to protect the copper surface

It is somewhat more difficult, but you can in fact lay out boards with more than two layers. For most small uses, this is too difficult to fabricate, build, and repair, so unless you are working for a large company, you probably won't run into many boards with more than two layers. Most PC motherboards, for example, have multilayer boards.

Nearly every PCB begins life as a schematic. Schematics don't directly relate to the PCB—they are just an abstract representation of the circuit you want to build. However, the schematic does contain all the connections the circuit requires to operate properly. It is possible for the designer to simply lay out the board by looking at the schematic, but modern tools give designers a better method: schematic capture.

Schematic capture is a fancy way of saying "drawing a schematic on the computer." Sometimes you'll copy a schematic from paper or another program into your *Computer-Aided Design* (CAD) program. If you are the original circuit designer, you probably just drew the schematic in your CAD package in the beginning. Even if you are using a different CAD package than the designer, many computer programs can export their schematic or at least a net list (a computer file that describes a schematic in a machine-readable format).

Once you have the schematic, you can start the layout. You'll need to decide on the size of the board (which is often dictated by other demands). You arrange the components (or at least make an attempt at arranging them) and then have to define the traces that connect the components together so they match the schematic.

You can do this manually—it is reminiscent of solving a maze or a puzzle. However, many CAD tools include an autorouter that will try to fit the traces so they match the schematic. Some of these do a better job than others, and it is a rare board that doesn't require a little manual intervention to successfully route.

This is often an iterative process. After you find yourself at an impasse, you may have to rearrange the components on the board and try again. If you are really desperate, you might even make the board larger to give yourself more room.

Once you have the board laid out, you can tweak the additional items, such as the component marking. Hopefully, you'll have room for mounting holes and any other extras you need to include.

At this point, you'll usually print out a copy of the artwork and examine it closely to make sure everything will really fit. Most CAD programs can also check to make sure you don't have any obvious blunders (such as wires not connected, traces touching, and so on).

Your next step depends on how you'll produce the board. If you are going the professional route, you'll generate some Gerber files (machine-readable files that describe the layers on your board) to describe the board for the automated machines that will produce the board. Files that describe the drills are usually in Excellon format (which is just another machine-readable file format). You can also use these files to have a photographic image professionally made (or a photo plotted). You can use it to make your own board.

If you are going to produce your own board, you'll probably just make a high-quality 1:1 printout of the artwork onto film or certain types of paper. You'll use the artwork to make a layer of resist on a piece of special copper-clad board and then use a special chemical that removes the copper from any place that isn't covered with resist. When you remove the resist, the copper that is left is your circuit design.

Making your own board also means drilling holes. An ordinary drill is not suitable for this job because the tiny drill bits you need break with any horizontal movement. A regular drill press may work, but it isn't really made for this purpose. Not only is it usually too slow, but it has *runout*, which means the bit makes a small circle instead of spinning exactly on its center line. A better option is a Dremel tool in a drill press attachment or a jeweler's press (which is a special drill press made for fine work).

Either way, you wind up with a finished board, ready to build. This sounds fairly simple, but as you'd expect, the details are where you get stuck. You'll explore each of these steps in future chapters. To summarize, the basic flow is as follows:

- Schematic capture
- Board layout
- Routing
- *Computer Aided Manufacturing* (CAM) output or film creation
- Board production

Don't confuse routing—the design of the copper pattern on the board—with the machining process known as routing. Usually, boards are mass-produced by making a single large board that has several repeated patterns. When completed, the boards are cut apart to produce individual boards. This is also known as routing (if it done with a router). In some cases, the boards are separated with a sharp blade (known as shearing). Routing is preferred if the board is not square or rectangular or has slots cut in the interior of the board.

Other PCB Advantages

You've already seen that PCBs can reduce wiring errors, decrease assembly costs, and improve the prospects for an automated assembly. PCBs have other advantages as well. Because mass-produced PCBs are uniform, it is possible to take advantage of their physical properties. For example, you might design a long trace of copper to form a small current-sensing resistor. In radio work, making small-value capacitors and inductors from PCB patterns is quite common. Many wireless devices (such as *Wireless Fidelity* [WiFi] network cards) use board patterns for antennas. Solder masks—a coating that repels soder from areas where it does not belong—are another advantage of printed circuitry. A board with a solder mask is easy to assemble. For high-density surface-mount parts, a solder mask is almost essential to prevent solder bridges between pins. Unfortunately, solder masking is not something you can usually do yourself, so if you require this feature, you'll have to plan on having your boards made professionally.

PCBs typically consume less space than a traditionally built circuit. This is particularly true when using surface-mounted components. Cell phones, personal organizers, and laptop computers could not be as small as they are without surface-mounted components and printed circuit boards.

Are there any disadvantages to PCBs? Yes. First, it takes a small investment to produce PCBs (although with the free software included in this book, you are ready to produce boards that will be made by a professional shop). Also, although PCBs reduce wiring errors in a repetitive assembly, if you make a mistake in the layout, it can be difficult and expensive to fix.

The Economics of PCB Production

The next time you are at an office supply store, ask them how much they'd charge you to print five business cards. They will probably tell you they won't do it. Then compare the price between 100 cards and 500 cards. All the cost in printing is up front. The primary expense is setting up to print that first card. Once you print the first card, the second card adds just a few pennies to the cost.

PCBs are the same way. The professional production of a board requires *tooling*—the production of reusable tools used in the manufacturing process. Most professional board houses will charge you $100 to $200 to produce these tools. That's $200 and you haven't even bought one board! However, you only pay this fee once (a *nonrecurring engineering* [NRE] *fee*). The

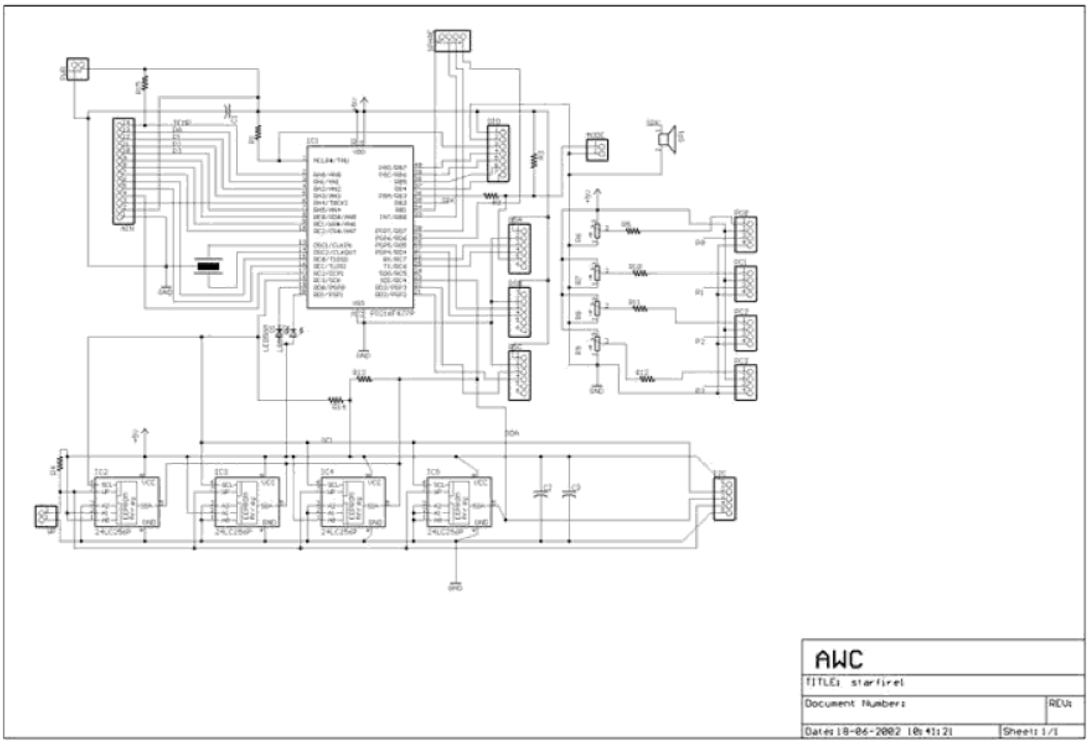

Figure 2-1 A typical Eagle schematic

When running in freeware mode, Eagle restricts you to schematics that only contain one page. However, this is more than adequate for many projects. In particular, Eagle freeware also restricts the PCB's size as well, so you couldn't use a very large schematic anyway.

When you start Eagle, you'll see the Eagle Control panel. This screen (see Figure 2-2) lets you explore several categories:

- **Libraries** Libraries contain the components you use to build circuits (such as resistors, capacitors, and integrated circuits). In Eagle, each library can contain many components that include both a schematic symbol and an associated PCB pattern (the footprint).
- **Design rules** Eagle can check that a board meets certain design criteria known as design rules. For example, you may not want traces closer than 10 mils and Eagle's design rules can make this check. You may want different sets of design rules to differentiate between boards you make yourself and different manufacturers (who may have different capabilities). Even when using a single manufacturer, it may be less expensive to have boards produced to lesser specifications, so you may want to manage different design rule sets.