Magic Tutorial #1: Getting Started

John Ousterhout
(updated by others, too)

Computer Science Division
Electrical Engineering and Computer Sciences
University of California
Berkeley, CA 94720

This tutorial corresponds to Magic version 6.

1. What is Magic?

Magic is an interactive system for creating and modifying VLSI circuit layouts. With Magic, you use a color graphics display and a mouse or graphics tablet to design basic cells and to combine them hierarchically into large structures. Magic is different from other layout editors you may have used. The most important difference is that Magic is more than just a color painting tool: it understands quite a bit about the nature of circuits and uses this information to provide you with additional operations. For example, Magic has built-in knowledge of layout rules; as you are editing, it continuously checks for rule violations. Magic also knows about connectivity and transistors, and contains a built-in hierarchical circuit extractor. Magic also has a *plow* operation that you can use to stretch or compact cells. Lastly, Magic has routing tools that you can use to make the global interconnections in your circuits.

Magic is based on the Mead-Conway style of design. This means that it uses simplified design rules and circuit structures. The simplifications make it easier for you to design circuits and permit Magic to provide powerful assistance that would not be possible otherwise. However, they result in slightly less dense circuits than you could get with more complex rules and structures. For example, Magic permits only *Manhattan* designs (those whose edges are vertical or horizontal). Circuit designers tell us that our conservative design rules cost 5-10% in density. We think that the density sacrifice is compensated for by reduced design time.

2. How to Get Help and Report Problems

There are several ways you can get help about Magic. If you are trying to learn about the system, you should start off with the Magic tutorials, of which this is the first. Each tutorial introduces a particular set of facilities in Magic. There is also a set of

Magic Tutorial #1: Getting Started Magic Tutorial #2: Basic Painting and Selection Magic Tutorial #3: Advanced Painting (Wiring and Plowing) Magic Tutorial #4: Cell Hierarchies Magic Tutorial #5: Multiple Windows Magic Tutorial #6: Design-Rule Checking Magic Tutorial #7: Netlists and Routing Magic Tutorial #8: Circuit Extraction Magic Tutorial #9: Format Conversion for CIF and Calma Magic Tutorial #10: The Interactive Route Magic Tutorial #11: Using RSIM with Magic
Magic Maintainer's Manual #1: Hints for System Maintainers Magic Maintainer's Manual #2: The Technology File Magic Maintainer's Manual #3: Display Styles, Color Maps, and Glyphs Magic Maintainer's Manual #4: Using Magic Under X Windows
Magic Technology Manual #1: NMOS Magic Technology Manual #2: SCMOS

Table I. The Magic tutorials, maintenance manuals, and technology manuals.

manuals intended for system maintainers. These describe things like how to create new technologies. Finally, there is a set of technology manuals. Each one of the technology manuals describes the features peculiar to a particular technology, such as layer names and design rules. Table I lists all of the Magic manuals. The tutorials are designed to be read while you are running Magic, so that you can try out the new commands as they are explained. You needn't read all the tutorials at once; each tutorial lists the other tutorials that you should read first.

The tutorials are not necessarily complete. Each one is designed to introduce a set of facilities, but it doesn't necessarily cover every possibility. The ultimate authority on how Magic works is the reference manual, which is a standard Unix *man* page. The *man* page gives concise and complete descriptions of all the Magic commands. Once you have a general idea how a command works, the *man* page is probably easier to consult than the tutorial. However, the *man* page may not make much sense until after you've read the tutorial.

A third way of getting help is available on-line through Magic itself. The **:help** command will print out one line for each Magic command, giving the command's syntax and an extremely brief description of the command. This facility is useful if you've forgotten the name or exact syntax of a command. After each screenful of help information, **:help** stops and prints ‘‘--More--’’. If you type a space, the next screenful of data will be output, and if you type **q** the rest of the output will be skipped. If you're interested in information about a particular subject, you can type

:help *subject*

This command will print out each command description that contains the *subject* string.

If you have a question or problem that can't be answered with any of the above approaches, you may contact the Magic authors by sending mail to **magic@ucbarpa.Berkeley.EDU** (or ucbvax!ucbarpa!magic). This will log your message in a file (so we can't forget about it) and forward the message to the Magic maintainers. Magic maintenance is a mostly volunteer effort, so when you report a bug or ask a question, *please* be specific. Obviously, the more specific you are, the more likely we can answer your question or reproduce the bug you found. We'll tend to answer the specific bug reports first, since they involve less time on our part. Try to describe the exact sequence of events that led to the problem, what you expected to happen, and what actually happened. If possible, find a small example that reproduces the problem and send us the relevant (small!) files so we can make it happen here. Or best of all, send us a bug fix along with a small example of the problem.

3. Graphics Configuration

Magic can be run with different graphics hardware. The most common configuration is to run Magic under X11 on a workstation. Another way to run Magic is on a mainframe with a serial-line graphics display. The rest of this section concerns X11.

Before starting up magic, make sure that your DISPLAY variable is set correctly. If you are running magic and your X server on the same machine, set it to unix:0:

```
setenv DISPLAY unix:0
```

Under X10, the layout window will appear in the upper left quadrant of your screen. The X11 server will normally prompt you for the window's position and size. This window is an ordinary X window, and can be moved and resized using the window manager.

For now, you can skip to the next major section: "Running Magic".

3.1. Advanced X Use

The X11 driver can read in window sizing and font preferences from your *.Xdefaults* file. The following specifications are recognized:

```
magic.window:    1000x600+10+10
magic.newwindow:  300x300+400+100
magic.small:     helvetica8
magic.medium:    helvetica12
magic.large:           helvetica18
magic.xlarge:    helvetica24
```

magic.window is the size and position of the initial window, while **magic.newwindow** is the size and position of subsequent windows. If these are left blank, you will be prompted to give the window's position and size. **small**, **medium**, **large**, and **xlarge** are various fonts magic uses for labels. Some X11 servers read the .Xdefaults file only when you initially log in; you may have to log out and then back in again for the changes to take effect.

Under X11, Magic can run on a display of any depth for which there are colormap and dstyle files. Monochrome, 4 bit, 6 bit, and 7 bit files for Mos are distributed in this release. You can explicitly specify how many planes Magic is to use by adding a suffix numeral between 1 and 7 to "XWIND" when used with Magic's "-d" option. For example, "magic -d XWIND1" runs magic on a monochrome display and "magic -d XWIND7" runs magic on a 7 plane display. If this number is not specified, magic checks the depth of the display and picks the largest number in the set {1,4,6,7} that the display will support.

The X10 driver only supports monochrome and 7 bit displays.

3.2. Serial-line Displays

If you are running Magic on a mainframe, each station consists of a standard video terminal, called the *text display*, and a color display. You use the keyboard on the text display to type in commands, and Magic uses its screen to log the commands and their results. The color display is used to display one or more portions of the circuit you are designing. You will use a graphics tablet or mouse to point to things on the color display and to invoke some commands. If there is a keyboard attached to the color display (as, for example, with AED512 displays) it is not used except to reset the display. The current version of Magic supports the AED family of displays. Most of the displays are now available with special ROMs in them that provide extra Magic support (talk to your local AED sales rep to make sure you get the UCB ROMs). More displays are being added, so check the Unix man page for the most up-to-date information.

4. Running Magic

From this point on, you should be sitting at a Magic workstation so you can experiment with the program as you read the manuals. Starting up Magic is usually pretty simple. Just log in and, if needed, start up your favorite window system. Then type the shell command

magic tut1

Tut1 is the name of a library cell that you will play with in this tutorial. At this point, several colored rectangles should appear on the color display along with a white box and a cursor. A message will be printed on the text display to tell you that **tut1** isn't writable (it's in a read-only library), and a ‘‘>’’ prompt should appear. If this has happened, then you can skip the rest of this section (except for the note below) and go directly to Section 5.

Note: in the tutorials, when you see things printed in boldface, for example, **magic tut1** from above, they refer to things you type exactly, such as command names and file names. These are usually case sensitive (**A** is different from **a**). When you see things printed in italics, they refer to classes of things you might type. Arguments in square brackets are optional. For example, a more complete description of the shell command for Magic is

magic [*file*]

You could type any file name for *file*, and Magic would start editing that file. It turns out that **tut1** is just a file in Magic's cell library. If you didn't type a file name, Magic would load a new blank cell.

If things didn't happen as they should have when you tried to run Magic, any of several things could be wrong. If a message of the form ‘‘magic: Command not found’’ appears on your screen it is because the shell couldn't find the Magic program. The most stable version of Magic is the directory ~cad/bin, and the newest public version is in ~cad/new. You should make sure that both these directories are in your shell path. Normally, ~cad/new should appear before ~cad/bin. If this sounds like gibberish, find a Unix hacker and have him or her explain to you about paths. If worst comes to worst, you can invoke Magic by typing its full name:

```
~cad/bin/magic tut1
```

Another possible problem is that Magic might not know what kind of display you are using. To solve this, use magic's **-d** flag:

```
magic -d display tut1
```

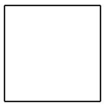
Display is usually the model number of the workstation you are using or the name of your window system. Look in the manual page for a list of valid names, or just guess something. Magic will print out the list of valid names if you guess wrong.

If you are using a graphics terminal (not a workstation), it is possible that Magic doesn't know which serial line to use. To learn how to fix this, read about the **-g** switch in the magic(1) manual page. Also read the displays(5) manual page.

5. The Box and the Cursor

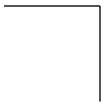
Two things, called the *box* and the *cursor*, are used to select things on the color display. As you move the mouse, the cursor moves on the screen. The cursor starts out with a crosshair shape, but you'll see later that its shape changes as you work to provide feedback about what you're doing. The left and right mouse buttons are used to position the box. If you press the left mouse button and then release it, the box will move so that its lower left corner is at the cursor position. If you press and release the right mouse button, the upper right corner of the box will move to the cursor position, but the lower left corner will not change. These two buttons are enough to position the box anywhere on the screen. Try using the buttons to place the box around each of the colored rectangles on the screen.

Sometimes it is convenient to move the box by a corner other than the lower left. To do this, press the left mouse button and *hold it down*. The cursor shape changes to show you that you are moving the box by its lower left corner:



While holding the button down, move the cursor near the lower right corner of the box, and now click the right mouse button (i.e. press and release it, while still holding down the left button). The cursor's shape will change to indicate that you are now moving the box by its lower right corner. Move the cursor to a different place on the screen and release the left button. The box should move so that its lower right corner is at the cursor position. Try using this feature to move the box so that it is almost entirely off-screen to the left. Try moving the box by each of its corners.

You can also reshape the box by corners other than the upper right. To do this, press the right mouse button and hold it down. The cursor shape shows you that you are reshaping the box by its upper right corner:



Now move the cursor near some other corner of the box and click the left button, all the while holding the right button down. The cursor shape will change to show you that now you are reshaping the box by a different corner. When you release the right button, the box will reshape so that the selected corner is at the cursor position but the diagonally opposite corner is unchanged. Try reshaping the box by each of its corners.

6. Invoking Commands

Commands can be invoked in Magic in three ways: by pressing buttons on the mouse; by typing single keystrokes on the text keyboard (these are called *macros*); or by typing longer commands on the text keyboard (these are called *long commands*). Many of the commands use the box and cursor to help guide the command.

To see how commands can be invoked from the buttons, first position the box over a small blank area in the middle of the screen. Then move the cursor over the red rectangle and press the middle mouse button. At this point, the area of the box should get painted red. Now move the cursor over empty space and press the middle button again. The red paint should go away. Note how this command uses both the cursor and box locations to control what happens.

As an example of a macro, type the **g** key on the text keyboard. A grid will appear on the color display, along with a small black box marking the origin of the cell. If you type **g** again, the grid will go away. You may have noticed earlier that the box corners didn’t move to the exact cursor position: you can see now that the box is forced to fall on grid points.

Long commands are invoked by typing a colon (‘‘:’’) or semi-colon (‘‘;’’). After you type the colon or semi-colon, the ‘‘>’’ prompt on the text screen will be replaced by a ‘‘:’’ prompt. This indicates that Magic is waiting for a long command. At this point you should type a line of text, followed by a return. When the long command has been processed, the ‘‘>’’ prompt reappears on the text display. Try typing semi-colon followed by return to see how this works. Occasionally a ‘‘]’’ (right bracket) prompt will appear. This means that the design-rule checker is reverifying part of your design. For now you can just ignore this and treat ‘‘]’’ like ‘‘>’’.

Each long command consists of the name of the command followed by arguments, if any are needed by that command. The command name can be abbreviated, just as long as you type enough characters to distinguish it from all other long commands. For example, **:h** and **:he** may be used as abbreviations for **:help**. On the other hand, **:u** may not be used as an abbreviation for **:undo** because there is another command, **:upsidedown**, that has the same abbreviation. Try typing **:u**.

As an example of a long command, put the box over empty space on the color display, then invoke the long command

:paint red

The box should fill with the red color, just as if you had used the middle mouse button to paint it. Everything you can do in Magic can be invoked with a long command. It turns out that the macros are just conveniences that are expanded into long commands and executed. For example, the long command equivalent to the **g** macro is

:grid

Magic permits you to define new macros if you wish. Once you've become familiar with Magic you'll almost certainly want to add your own macros so that you can invoke quickly the commands you use most frequently. See the *magic(1)* man page under the command **:macro**.

One more long command is of immediate use to you. It is

:quit

Invoke this command. Note that before exiting, Magic will give you one last chance to save the information that you've modified. Type **y** to exit without saving anything.