```
 -----------------------------------------------------------------------
------
|
|
|
|
|
|
|                 An Introduction to LOG with AnaLOG
|
|                              5.10                                    |
|    LOG and DigLOG, Copyright (C) 1985, 1990 by David Gillespie
|
|    AnaLOG, Copyright (C) 1985, 1990 by John Lazzaro
|
|    Correspondence to lazzaro@cs.berkeley.edu or daveg@synaptics.com
|
|
|
 -----------------------------------------------------------------------
------
```

Table of Contents

General information
Digital simulator
Analog simulator
Other simulation features
Plotting circuits and data
LOG-to-NTK conversion
LOG-to-SPICE conversion
Creating your own gates
Reference material

General Information

Getting started
Using the program
Circuit editing
Configuring gates
More about editing
LOG survival tips

GETTING STARTED

1. LOG is a large circuit editing and simulation system. It has facilities for digital simulation (the original LOG), analog simulation (AnaLOG), network generation (LOGNTK), and plotting (LPLOT). You probably won't need to use all of these, so large parts of this document may not be relevant to you. This document was originally written to describe AnaLOG, so it is biased toward that application. Another learning aid for anaLOG

are the lessons files located in the /usr/chipmunk/log/lib directory.

2. To run LOG, give the command:

```
    % diglog   or
   % analog
```

   This runs LOG and sets it up for digital or analog simulation. Both simulators are present in both of these files; the only difference is which gates you get in your default menu.

3. To exit at any time, use the Exit command from the menus.

4. To report bugs, send e-mail to:

```
   lazzaro@cs.berkeley.edu
or daveg@synaptics.com
```

USING THE PROGRAM

1. You control LOG mainly with the mouse, and also from the keyboard. It takes some time to get comfortable with the mouse in LOG, but it will soon become second nature. There are three basic mouse motions you need to learn:

2. TAP. Press and release the left button on a spot. This should feel like tapping a key on a keyboard. The spot you are tapping is indicated by the cursor on the screen, usually a small arrow.

3. PRESS/DRAG. Press down and hold the left button, move the cursor somewhere else, and release.

4. REST. Press the right button and release. This is used to cancel operations like drawing wires.

5. Some LOG functions appear right on the screen, so you can activate them just by tapping the word. Others are in "menus," which you get by pressing on one of the words Frills, Editing, Cursor, or Misc, then dragging to and releasing on the word you want. If you decide you didn't want the menu after all, just release on some other part of the screen.

6. There are also keyboard commands, though everything you really need as a novice can be found in on-screen menus. When you press a key for LOG, be sure the cursor is in one of the LOG windows (it doesn't matter which one). There are several features of the keyboard to notice:

7. SINGLE KEYS. Many commands appear on keys as well as menus, just for convenience. For example, you can press the letter "s" to do the same thing as SCOPE in the Misc menu.

8. ":" COMMANDS. You can give full-word commands by pressing the ":" key, then typing the command. This is sometimes more

convenient; for example, if you know the name of the file you want to load, typing ":LOAD FOOBAR" is faster than selecting LOAD from Misc, then pointing to the file name. [NOTE: The Unix version of LOG does not support this feature yet.]

9. <CNTRL>-C. Press the Control key and the "C" key together to exit from any mode you may find yourself in.

SURVIVAL TIP #1: If you get lost and confused, press <CNTRL>-C until you know where you are.

10. SPACE BAR. The space bar at most times means to refresh the screen. If the screen has gotten trashed by too much editing, press this key to clean it up. LOG will also silently clean up the screen if you stop editing for a few seconds.

CIRCUIT EDITING

1. At the bottom of the screen are several circuit symbols. In LOG lingo, these are called gates. The area above the blue lines is the drawing area, where you draw your circuit diagrams. When you start AnaLOG, two very special gates, the the clock icon and the AnaLOG "scoreboard," show up automatically. You are welcome to move these around or delete them if you like, but you will find them extremely handy to have nearby. (If you are using LOG as a digital simulator these gates will not appear.)

2. To begin drawing a circuit, choose the circuit symbol you want, press down on it and drag it (still holding down) to the desired place. To move a gate, just press and drag. To delete a single gate, just "throw it away" by moving it to the bottom corners of the screen.

3. If you don't see the gate you want in the menus, tap the word CAT to see the "gate catalog." You can then press and drag any of these gates to add it to your circuit. If you will be using a lot of a particular kind of gate, you can drop it into one of the several menu slots to replace another less interesting gate. If you get into the catalog by accident, press an empty part of the screen, or, as usual, hit <CNTRL>-C to get out.

4. If you look closely at a gate that has been put into the drawing area, you will see some red dots. These are called the gate's red dots, or pins, and they are the places to connect other circuit elements. The easiest way to connect two gates together is to put them down with the appropriate red dots on top of each other.

SURVIVAL TIP #2: The single most common mistake in LOG is to stick two gates, e.g. a capacitor and a ground, so close together that they look connected but their red dots pass through each other. The gates are NOT connected, and the circuit will NOT
simulate
correctly. The solution is to nudge the gates apart a bit, then refresh the screen (space bar) to make sure the red dots are aligned.

5. If the gate comes in an awkward orientation, you can tap it to rotate it by 90 degrees. This works for gates in the circuit as well as for menu-area gates. Notice that LOG distinguishes between tapping a gate, to rotate it, and pressing a gate, to move it. Practice a few times until you can tap and press at will, without getting confused. (Some gates, like switches, do special things when you tap them in the circuit. These can only be rotated in the menu area.)

6. A much neater way to connect things is with wires. To draw a wire, just tap the place you want to start, move horizontally or vertically, then tap the endpoint. This automatically starts a new wire at the endpoint, so you can make a chain of connected wires. To finish the chain, just tap the right mouse button. The whole sequence of tap-tap-rest takes a few minutes to learn but soon feels very natural.

7. You can move wires by pressing and dragging, just like gates. If you press in the middle of the wire, you move it from side to side but keep the same length. If you press one of the ends, you also get to stretch that end. As usual, to delete a wire just grab it and drag to the edge of the screen.

8. Wires are connective along their whole lengths, so you can be sloppy about connecting to gates as long as you at least overlap the red dot. Wires and gates are drawn in slightly different colors so you can tell them apart if you need to.

9. If you connect wires in a "T", LOG "solders" them automatically. If you cross them in a +, LOG assumes they are unconnected, so you must go back and tap the intersection to solder them. Because of this, "T"-connections are preferable.

10. If you should run out of room on the screen for your circuit, you have several options: press ">" or "<" to "zoom" in or out on the circuit (these are also available in the Frills menu), or use the arrow keys to "scroll" up, down, left, or right. The drawing area is essentially infinitely large; so large, in fact, that you can get lost due to excessive scrolling. If this happens, HOME in the Cursor menu will take you back to the center of the drawing universe.

11. LOG can keep up to 9 circuit diagram "pages" in memory at a time. Each page is associated with a different file. To switch to a new page, press one of the digit keys 1 through 9.

CONFIGURING GATES

1. Once you have drawn, say, a capacitor, you may wonder what its capacitance is, and how to set the capacitance you want. The way to do this is called configuration.

2. In the lower-right corner of the screen is the word ROT with a curving arrow. This tells LOG that when you tap a gate, it should rotate it by 90 degrees. If you press the word ROT, it changes to

MIR. Now if you tap a gate it mirrors itself left-to-right. Pressing MIR leads to MIR with a bar (up-down mirroring), then to CNFG in yellow, then back to ROT. If you tap a gate while in CNFG mode, instead of reorienting it you open it up for configuration.

3. Each kind of gate has a set of "attributes," usually numbers, that tell more specifically what kind of circuit element it is. For example, AnaLOG capacitors have a "capacitance" attribute, and transistors have a hatfull of attributes about width/length, kT/q, parasitic capacitances, etc.

4. In the digital world, most gates do not have attributes. An exception is the clock generator, CLOCK, which you can configure to control the clock frequency and waveform. Also, many digital gates have text on their attribute screens that describes how to use the gate.

5. One of the attributes is highlighted. To highlight a different attribute, press the up- and down-arrow keys.

6. To enter a new value for an attribute, highlight it, then just type in the value you desire.

7. Some attributes can have scale factors, e.g., "m" for milli- and "u" for micro-. These will show up as, say, "1.2mV", meaning 1.2 millivolts. When you type a new value, you can always leave off the V since that part is obvious, but don't forget to include the "m" if that's what you want! A common mistake is to change 20pA to 30A, thus seriously perturbing the operation of the simulated circuit. When in doubt, you can always type scientific notation in "1.2E-4" format.

8. Some attributes are choices instead of numbers. For example, voltage sources let you choose a waveform of either "DC," "Pulse," or "Sine." You can type whole words to change these attributes, or use the arrow keys to cycle through the choices.

9. To return an attribute to its default value, just press ENTER all by itself on that line. Some attributes can be blank, in which case whatever feature they correspond to is disabled. Others must have values, and LOG will use some default if you try to enter "blank."

10. When a gate is first created, all its attributes have their default values. However, if you change a gate's attributes, then drop that gate into one of the menu area slots, then new gates drawn from that slot will get those new values for their attributes. You can also reconfigure gates in their menu slots using Probe mode (in the Frills menu.)

11. To finish configuring a gate, press the right mouse button again, or hit <CNTRL>-C.

MORE ABOUT EDITING

1. In the Editing menu there are commands for doing large-scale circuit editing that would be unreasonable with just the basic mouse motions.

2. DELETE: Press Editing, drag to Delete, release left mouse button. Now when you move up into the drawing area, the cursor is shaped like a pair of scissors. You can now delete individual things by touching them (with the point where the blades of the scissors cross), or you can delete everything in a rectangular region by pressing at one corner of the region and dragging to the opposite corner. Press the right button to exit Delete mode. You can also press the "d" key to get the Delete mode scissors. Due to an intermittent bug, you may need to click the right button a second time for the scissors to appear.

3. PASTE: After you have deleted a group of objects, you can select Paste to "paste" down copies of what was in that region. The Paste command highlights the region that was actually deleted; you can paste back in there for a simple "undelete," or you can paste other places for moving or replicating a chunk of the circuit. This is also on the "*" key.

4. COPY: This is similar to Delete, except that it doesn't actually delete the contents of the rectangle. It is for making copies of something without disturbing the original. After you have copied your object or area of objects, you are automatically switched into Paste mode. This is also on the "/" key.

5. MOVE: Move an area. This command is like a DELETE followed by a PASTE; it moves an object, or all the objects in a given area, to a new location. It is also on the "m" key. Note that the pressing "m" and tapping on a wire moves the wire in a different (often useful) way than simply grabbing the wire.

6. Implicit MOVE/COPY: Pressing in empty space and dragging out a box automatically does a combination MOVE and COPY command. The first place you select moves the contents of the box to that place; if you press at additional places, extra copies are drawn there.

7. Open/Close commands: These are for expert users only, because using them improperly may damage LOG's internal data structures. What they do is shift things around on the screen to make extra room for additions, or to close up excess space. If you want to do this, use Delete/Paste instead, or ask someone.

8. The Labels command in the Frills menu lets you add text labels to your diagram. Select Labels, type the label you want, then grab it and drag it, as usual, into place. You can edit a label by tapping it, then using the keyboard. Labels generally have no effect on the simulation, but they make plots look much nicer. Also, certain tools such as LOGNTK and the Digital LOG hierarchical simulation gates can read the labels for additional information about the

circuit.

9. The Boxes command in the Frills menu adds yellow dashed boxes to the diagram, again for cosmetic purposes only. Once you're in Boxes mode, you can drag out boxes in the usual fashion. To edit boxes, grab a corner for reshaping or an edge for moving. Both labels and boxes can be deleted by throwing them off the bottom corners of the screen.

10. To save your circuit, select Save in the Misc menu, then type the file name to use. It's a good idea to hit Shift-S, or type ":SAVE", periodically to save the circuit under the previous name if any changes have been made. ":SAVE" with no file name argument saves all pages which have been editing back into their original files. Circuits are saved in files ending with ".lgf" (for LOG Format). The previous version of the file is backed up under the same name but with ".lfo".

11. To load a circuit you have saved before, select Load from the Misc menu. You will get a list of saved circuits in your directory.
You
can tap one of these to load it, or just type the name.
[NOTE: The UNIX version doesn't support directory listing yet, you'll need to type the name of the file]. AnaLOG saves the
complete
simulation state along with the circuit, but you may want to do a RESET after loading anyway.

12. The Grid command in the Cursor menu switches between the usual arrow cursor and a cross-hair cursor which is useful for aligning things.

13. The Yardstick command in the Frills menu lets you draw an arrow on the screen which you can use to check the geometric arrangement of objects on the screen. The yardsticks are not "real"---they go away when you refresh the screen. You can also press the "y" key to get a yardstick.

14. If you select Status from the Misc menu, you get a set of screens describing various aspects of the LOG system. Move from screen to screen with the arrow keys. The first screen includes the date
and
time, amount of memory free (approx.), plus various other stuff. Later screens display the status of the LOG editing buffers and keyboard mappings, then there is a screen for the LPLOT program (that handles the PLOT command described below), and for the
analog
simulator.

SURVIVAL TIPS

SURVIVAL TIP #0: Select Exit from Misc to exit LOG.

SURVIVAL TIP #1: If you get lost and confused, press <CNTRL>-C until you know where you are.

```
      SURVIVAL TIP #3:  The single most common mistake in LOG is to
                        stick two gates, e.g. a capacitor and a ground,
                        so close together that they look connected but
                        their red dots pass through each other.  The gates
                        are NOT connected, and the circuit will NOT
simulate
                        correctly.  The solution is to nudge the gates
                        apart a bit, then refresh the screen (space bar)
                        to make sure the red dots are aligned.
```

DIGITAL SIMULATION

Basics of digital simulation
A tour of the digital gates
Hierarchical digital simulation
Digital VLSI simulation

BASICS OF DIGITAL SIMULATION

1. Digital LOG is simulating your circuit constantly, even as you edit. Every wire has a "state," corresponding to the voltage on the wire:

Zero	a wire connected to ground,
One	a wire connected to Vdd,
Weak one	a wire connected to Vdd through a pullup resistor,
None	a wire that is floating (not connected)

2. You send signals into the circuit using switches, and read the state of the circuit using displays such as LED's.

3. If you ever drive a node (wire) with both One and Zero at the same time, the node starts flashing to indicate a "conflict." Since this is not an analog circuit simulator, digital LOG can not represent the correct value of the wire, so it flashes the wire and picks either One or Zero arbitrarily.

4. Digital LOG uses a "unit delay" simulation model. All gates are considered to have the same delay, whether they are NAND gates or complex counters. (The STABILIZE command may change this behavior for short periods of time.)

A TOUR OF THE DIGITAL GATES

1. The standard gates in the Digital LOG library provide all of the logic functions you will need to build a digital circuit. If you need a function that is not provided, you can design it yourself with the LOGED program, or using digital hierarchy.

2. All the standard library gates use consistent conventions for their pins. For example, when a row of pins carries a binary number, the most significant bit is on the left or bottom end

of the row. Signals are active-high unless a "bubble" shows otherwise.

3. Input/output devices and controls:

SWITCH	Looks like a small pointed box. It is a voltage source that drives either One or Zero onto its output. The center lights up black or red to indicate the state of the switch. Touch the switch to change its state.
SWITCH2	A smaller switch suitable for packing into tight rows.
PULSE	Pulse switch. Looks like a double-pointed box. It acts much like SWITCH, except that when you touch it it goes to One only briefly, then back to Zero.
LED	Digital display; small square box. It lights up according to the value being driven on it: red for One (or weak One), black for Zero, and background gray for undriven wires (None). It has connection points all around so you can connect to it from any direction. (Note: This means that placing two LED's right next to each other will short their pins together.)
LED2	Structurally the same as LED, but the input pins are treated a little differently. In LED, all eight pins are connected together. Thus touching LED's will short each other out. In LED2, the eight pins are independent. If any of then is driven high, the LED2 glows red. If any is driven low, it glows black. If some drive high and some low at the same time, the LED2 does not glow. Since the pins are not internally connected, you can let LED2's touch and they won't mind.
LED3	This is a miniature LED useful for tightly packed situations. Connect to it by running the wire right through the center.
EDGE	Edge detector. This is the differently-pointed box. It lights up red whenever it sees any change on its input, and only resets to black when you touch it.
KEYPAD	Numeric keypad. This is a 16-key keypad that generates a binary number from 0000 to 1111 when you press one of the 16 keys on the keypad. The lowest output is most significant. There is also a "strobe" output on the bottom, that generates a pulse each time a key is pressed.
7SEG	Numeric display. This is a 7-segment LED display, which displays the symbols 0 through F for inputs 0000 through 1111. It has two equivalent sets of inputs, for your convenience. The most significant

bit is on the left, and at the bottom. (The two sets of pins are equivalent; you may use either.)

ASCKBD	ASCII keyboard. Analogous to KEYPAD, but with an alphanumeric keyboard instead which generates 8-bit ASCII codes. Tap a key to generate its ASCII code (plus a strobe signal). Tap SH, then a key, to shift that key. CT is the Control key. Configure the gate and select Key-Codes mode to generate positional codes for all the keys instead of ASCII. This lets you handle Shift and Control yourself, if you like.
ASCDISP	ASCII terminal display. Shows 16x64 characters. To print a character, put its ASCII code on the data pins and pulse the Strobe pin. Above Strobe is the Clear pin, which resets the terminal. This terminal understands most standard Chipmunk control codes such as Carriage Return and Line Feed. Form Feed clears the screen. Certain variations on the control codes can be had by Configuring the gate.
CLOCK	This is a two- or four-phase clock generator or oscillator. By default, it is a two-phase square wave generator in which the outputs are CLK and not-CLK; the output changes on every time step (i.e., once per standard gate delay). You can reconfigure it to take several time steps per clock phase, or to clock according to real, physical time, or to be a "synchronous" clock which changes as soon as the rest of the circuit has settled down from the last change. Also, you can change the waveform to four-phase mode, where the outputs are completely non-overlapping CLK1 and CLK2, each with 25% duty cycle.
BREAK	Breakpoint. This is a box with an X in it. This gate is the circuit equivalent of a software "breakpoint." When its inputs see a positive- or negative-going edge (respectively), it turns the simulation off so you can examine what was the state of the system at the instant the transition occurred.
TIE	Pullup resistor. This generates a weak One, which can be overridden without conflict by an open-collector output pulling down.
TIEGND	Pulldown resistor. Generates a weak Zero.

4. Connection gates:

VDD	Power supply. Provides a constant One signal.
GND	Ground. Provides a constant Zero signal.
TO/FROM	Terminals. These are the two arrow-shaped gates

in the Catalog. They are equivalent except for appearance. They are used for putting names on signals: To name a signal FOO, connect a terminal to it, then tap the space next to the arrow and type FOO. If there are several terminals in the circuit with the same names, they are all electrically connected as if by wires. TO/FROM names can be used to pass signals between circuit pages. Signal names are also used with the Scope mode, to be described below. Dropping a TO/FROM into the menu area preserves its programmed name; if the name ends in a number, each fresh gate pulled from the slot gets a new number. Predefined signal names include "Gnd" and "Vdd", equivalent to the GND and VDD gates, and "Reset", which is normally grounded but receives a pulse every time you do a Reset command (in the Misc menu).

5. Standard digital gates:

AND, OR, NAND, NOR, XOR, XNOR
Standard two-input logic gates. Also, AND3, NOR4, etc. are multiple-input gates, and INV is an inverter.

ANDX, ORX etc. These are versions of AND, OR, etc. whose pictures have been transformed according to DeMorgan's theorem of logic, which says that an AND gate is just like an OR gate with its output and all its inputs inverted.

COMPL This is a complementary-output buffer. Its two outputs provide the input and its complement, with identical propagation delays.

DNEG, DPOS, JKNEG, JKPOS, TNEG, TPOS
Flip-flops. Each kind comes in negative- and positive-edge-triggered flavors. There is a choice of D flip-flops (which latch a Data bit on each clock), or J-K flip-flops (which set, reset, toggle, or stay the same on each clock), or Toggle flip-flops (which toggle or stay the same on each clock).

LATCH A level-sensitive latch. When G is One, the output follows the D input. When G is Zero, the output freezes at the last D input value that was followed.

SHIFT A four-bit shift register. On a falling clock transition, it either loads from the parallel inputs (if M is One), or shifts in the direction of the arrow (if M is Zero).

SRAM8K Static RAM or ROM, storing 8K bytes of 8 bits. Address inputs are on the left (MSB lowest). Data input/output pins are at the bottom (MSB on the the left). Control pins are Chip Enable (CE), Read, and Output Enable (OE). If Read is high and CE and OE are low, the addressed byte is driven on the Data pins. If Read and CE are low (regardless of OE), the value on the Data pins is stored in the RAM. Otherwise, the Data pins are undriven and ignored. (To make a ROM, simply leave Read unconnected.)

Configuring the SRAM8K allows you to examine or change any byte in the memory, control whether the memory contents should be recorded in saved circuit files, and save or load the memory to ".hex" data files. SRAM8K has the same unit propagation delay as all Digital LOG gates---bear in mind that real memory chips are much slower!

6. There is also a large number of 7400's series TTL gates to choose from in the gate library. These are all intended to be close models of their true counterparts, but there are no guarantees. In particular, remember all propagation delays are the same.

7. A set of gates for simulating digital VLSI transistors exists. All of these gates begin with "V_" and are described below.

8. The gates whose names begin with "A_" are for use with the ACTEL design system. This is a special box which takes network files produced by LOGNTK from circuits consisting of "A_" gates, and programs gate array chips while-U-wait to produce fast, custom digital chips for prototyping. Use of the ACTEL system is beyond the scope of this document.

9. The gates DIGH, FORCEDRV, and INST1 through INST5 are intended for use with hierarchical definitions, described in the next section.

10. To view the "program" that the simulator uses for a gate, press the shift-D key, then touch the gate in the circuit diagram. A small window pops up containing the program. Press somewhere not on a gate to exit this mode. To erase the definitions, just hit the space bar to refresh the screen. Notations like "#4" refer to pin numbers; letters like "E" refer to internal state variables of the gate. For further documentation, see the LOGED manual.

HIERARCHICAL DIGITAL SIMULATION

1. The Digital LOG simulator is capable of simulating hierarchical designs. That is, you can draw a complicated circuit on one page, then include any number of "instances" of that circuit on other pages in the form of single LOG gates. While this feature is specific to the digital simulator (AnaLOG has no hierarchy, for example), the LOGNTK network file generator also understands hierarchy using the same notation and conventions.

2. The simplest way to build hierarchically is by using the standard digital "instance" gates, INST1 through INST5. These

gates look like blocks of various sizes, with connection pins all along their rims. They also have little arrows so you can tell if they have been rotated or mirrored. (NOTE: There are also "general-purpose" instance gates GINST1 through GINST5 in the library. They are not digital gates and the digital simulator will not understand them! These gates are for use only with the LOGNTK program.)

3. To create a new gate from a circuit diagram, simply put your diagram into one of LOG's nine editing pages. Now, take an INSTn gate of an appropriate size and add it to your diagram. Connect wires to the gate from the diagram in the same way you plan to connect the gate "in real life". For example, you might have three input wires on the left edge, one output on the right edge, and a clock input on the bottom edge.

4. Exact positions don't matter for the instance gates. They simply divide your connections into four classes, "top", "bottom", "left" and "right", and count the wires along each edge. When you use the gate in another circuit, you must make the same number of connections on each edge. The order of the wires you connect matters, but not their exact positions along the edge.

5. Create a label (by pressing the "l" key) which is the name you want to give this circuit, in double-quotes: "Joe". This label can be located anywhere on the page that contains the definition of "Joe". (An alternate, older notation is to make a label of the form: <name: Joe> .)

6. Switch to the CNFG (configuring) gate-tapping mode, then tap the instance gate and enter the same name in the "Instance of" field: Joe. Quotes are optional, and upper/lower case don't matter. Since this gate has the same name as the page it is on, it becomes a "template" or "example" gate. You now have a complete definition for your new gate.

7. To use this gate, create another instance gate on a different page and configure its "Instance of" attribute to have the name of the definition you want to use. You do not need to use the same size or shape of instance gate as you did before. As long as you make the right number of connections, it will work. Instance gates will be dim until they are "happy", that is, correctly connected and configured in order to match their (also correct and complete) definition.

8. If the generic instance gates are too boring, you can create your own instance gate, with any picture you like, using the LOGED program. The easiest way is to copy a gate like INST1 from the "log.gate" file (ask your local maintainer) into your own private gate file, then throw away its picture and draw your own. You can use this gate in just the same way as the standard instance gates, except that typically you will only include as many pins on the
gate
as you actually plan to use.

9. If you don't like template gates, you can instead use labels

containing "port lists." This works well only for instance gates you have drawn yourself. Pins on a gate are numbered from 1 to N, the total number of pins. Include in your definition, instead of the template or example gate, a label of the form: <Port: A B C D> where "A B C D" is a list of TO/FROM signal names. In this example, A will be associated with pin 1 of the instance gate, B with pin 2, etc. The port list must have exactly the right number of names. In general, port lists are more awkward to use than templates. They exist mostly for historical reasons only.

10. If you run out of pages you might want to put more than one definition on a single page. Simply enclose each definition in a dashed box (press the "b" key to draw one of those). When Digital LOG looks for the "Joe" label to find the definition called Joe, and that label is inside a dashed box, then it considers only the things on that page which lie inside the box to be part of that definition. Since each page can be scrolled essentially infinitely far, there is plenty of room for all the definitions you need.

11. Instances may be nested. The definition for "A" can include instance gates for definition "B"---as long as "B" does not include instances of "A"! Aside from circular references, any combination is allowed. (The template gate is a special kind of "circular reference" which is excused.)

12. If you include switches or keypads in your definition, they will be propagated to all instances. For example, if you make a definition with a switch set to "0", then the equivalent node inside every instance will also be "0". If you flip the switch, every instance of that definition will simultaneously switch along with it. LED's and other "display" gates have no effect inside hierarchical definitions.

13. If the definition includes a label of the form: <Inert A B C> then gate types A, B, and C are ignored if they appear in the definition. Every definition effectively includes the command <Inert LED 7SEG>. The word "Inert" and the names of the gates are case-insensitive.

14. Digital hierarchy works by compiling the behavioral descriptions of all the gates in the definition into one big program which simulates the instance gates. Most gates' programs are written purely in this language, which is described in the documentation for the LOGED program. However, some gates, such as the 8K static RAM, are defined by Pascal procedures instead. The digital hierarchy compiler can not compile these gates, and will display an error message if you try to use one in a definition. Nearly every SSI- and MSI-level gate in the standard library is written purely in LOGED-language, though, so this is not often an issue.

15. Every time you make any change to the definition, Digital LOG must "recompile" the definition. If the definition is large, this can take time. While you are editing a definition, you may wish to turn simulation off (press the "o" key). No digital gate definitions will be recompiled until you turn the simulator back on.

16. To gain more control over the digital hierarchy compiler, include a DIGH gate in your definition. The DIGH gate displays the status of the compilation of the definition it is in, and also has controls that affect the kind of compilation that is performed. You can see the results of the compilation by pressing Shift-D (the DEFINE command) and tapping on any relevant instance gate (including the template).

17. Normally, Digital LOG applies many optimization steps when compiling a definition which make the resulting instance program simpler. Depending on what is in the definition, optimization can make the gate substantially faster to simulate, both by making the overall simulator faster, and by reducing the propagation delay of the gate. Optimization may not be safe for definitions that include race conditions or other effects dependent on exact gate delays; also, it may be slow for large definitions. You can configure the DIGH gate (by tapping it while in CNFG mode, on the blue border of the gate) to specify optimization only on request, in which case you must touch the "Optimize" button on the DIGH gate in order to optimize the definition.

18. The DIGH gate's configuration screen also has a switch to disable delay-folding optimizations; with these disabled, gate delay characteristics of the defining circuit are exactly modelled, with only simple local optimizations performed.

19. Another useful optimization for instance programs involves the FORCEDRV gate, which looks like a little diamond. Placing this diamond on any wire in a definition tells the digital instance compiler to assume this node will always be driven with a "1" or "0" during operation, that is, it will never be left floating. You can still leave one of these nodes unconnected if you wish, but the simulator will no longer guarantee that the results will be the same as for original definition circuit. In some cases the presence of FORCEDRV's allows LOG to create a much more compact and efficient instance program. FORCEDRV's are ignored if the optimizer is turned off.

20. Configuring DIGH lets you control other aspects of the hierarchy compiler. In particular, you can set the "Dump mode" to "Write", then specify the name of a file. Then, if you touch "Write" on the DIGH gate at some later time, the definition will be recompiled and written into a file which LOGED can read. This allows you to convert your definition into a permanent gate in the library. To configure the DIGH gate, touch around the blue-colored edges of the gate, away from the buttons in the center of the gate.

DIGITAL VLSI SIMULATION

1. The group of gates whose names begin with "V_" are intended to

emulate digital CMOS transistors and structures in the Digital LOG simulator. Naturally they are not as accurate a simulation as AnaLOG's transistors, but they have the advantages that the digital simulator is much faster, and that they can be used in combination with all the other digital gates in the library.

2. The basic VLSI gates are V_NFET and V_PFET, the CMOS transistors. These gates can model most digital CMOS circuits which do not involve charge sharing or other "analog" effects. The transistors' gate pins have "capacitance" which holds the last charge stored on them, allowing you to build dynamic as well as static registers. The transistors are symmetric: You don't have to worry about which end is the source and which is the drain.

3. V_NFET conducts zeros between its source and drain pins whenever its gate pin is one. It does not conduct logic ones at all. Similarly, V_PFET conducts logic ones when its gate is zero. In other words, V_NFET and V_PFET act just like CMOS transistors, as far as most digital applications are concerned, and that's all you need to know. If you are interested in a detailed understanding of how the CMOS simulation works in LOG, read the following few paragraphs.

4. Because Digital LOG is a gate-level simulator, it needs to know at any given time which pins are "inputs" and which are "outputs". Since transistors are symmetrical devices which can conduct in either direction, the V_NFET and V_PFET gates are implemented as bidirectional buffers which can change their direction of transfer according to their environment. Specifically, any time a V_NFET sees a zero on its output, and an undriven node on its input, it changes direction so that it can conduct the zero onto the undriven node. This change of direction occurs only if the gate is switched on. Likewise, the V_PFET switches direction in order to conduct a one onto an undriven node. In most circuits, this all happens automatically and you can ignore it. But if a particular V_NFET or V_PFET looks like it is getting confused, you might swap it for a V_NFETD or V_PFETD (described below).

5. The "gate capacitance" is implemented by driving a "weak" one or zero onto the gate, according to the last "strong" value seen there. A weak one or zero is the kind of signal produced by a TIE (resistor to Vdd) or TIEGND (resistor to Gnd) gate. If another circuit drives an actual value onto the gate, this strong value overrides the weak "capacitative" value. To avoid undesirable charge-sharing problems, the sources and drains of transistors treat weakly driven ones or

zeros as if they were undriven. This means, for example, that a value stored dynamically on the gate of a transistor will not leak back through any "pass" transistors connected to that gate.

6. There are also "directional" transistors, V_NFETD and V_PFETD. These gates are similar to V_NFET and V_PFET, respectively, except that they conduct in a fixed direction. Information is always propagated *from* the pin with the arrow, *to* the other pin. In terms of current, the directions of the arrows show the direction of current flow.

7. Finally, there are V_NFETX and V_PFETX. They are directional transistors that do not "weakly" drive their gates to show the dynamically stored value there. Instead, they record the value in a hidden internal state variable in the transistor. V_NFETX and V_PFETX are provided as a minimal-frills transistor, in case the fancier transistors are too "smart" to work properly in your circuit. Also, since the programs for V_NFETX and V_PFETX are simple they will simulate faster, especially in hierarchical designs.

8. V_NFETX and V_PFETX can pass weakly driven signals as well as strongly driven ones. As a result, you can use them to simulate NMOS circuits. Use V_NFETX for enhancement devices, and the standard TIE gate as a depletion (pull-up) device. The fancier transistors will not work in this way because the the weak one from the TIE gate will not be able to overpower the weak one or zero being driven by the transistor's capacitance.

9. The following additional gates are provided in the VLSI library:

V_BUF	VLSI buffer. This gate acts like a non-inverting buffer made out of V_NFETX and V_PFETX transistors. That is, it does not affect its input pin in any way, but the output pin gets a copy of the most recently driven input value. You can use V_BUF to get a visible indication of what value is stored on the hidden gate capacitances of V_NFETX's and V_PFETX's on the same wire.
V_INV	VLSI inverter. This gate acts like a usual digital inverter. The difference between V_INV and the standard INV gate is that V_INV's input is dynamic in the same way as the V_NFET and V_PFET transistors' gates. V_INV weakly drives its input according to the capacitively stored value. It is provided simply as a shorthand to drawing the two transistors.
V_AND, V_OR, V_NAND, V_NOR	VLSI AND, OR, NAND, and NOR gates. These gates have capacitive inputs, just as in V_INV.

V_TRANS, V_TRANSN
: CMOS transmission gates (with active-high or active-low control). These gates act much like V_NFET's and V_PFET's connected together to form the traditional CMOS "pass" gate. They require only a single control input, generating the inverted control signal internally. For V_TRANS, the gate conducts ones or zeros in either direction whenever the control is one, and is an open circuit when the control is zero. For V_TRANSN, the gate conducts only when the control is zero. If the gates get confused and you need a directional transmission gate, you can use the 74125 and 74126 "tri-state buffer" gates from the library.

V_CSRL
: Complementary Set-Reset Logic element. This is a one-bit latch designed to emulate a CSRL gate. If the "latch" input is high, then a zero on the upper data input clears the latch and a zero on the lower data input sets the latch. If the latch input is low, or if both data inputs are high or undriven, the latch remembers its previous value. The result is undefined if the latch is high and both data inputs are low. The latched value and its complement are driven on the outputs.

V_CSRL0
: To get a complete two-phase master-slave flip-flop, connect two V_CSRL's in series. The V_CSRL0 gate is electrically equivalent to V_CSRL, but with the control pin in a different place so that parallel "phase 1" and "phase 2" clock busses can be used.

V_CSRL2
: Two-bit parallel CSRL latch.

V_CSRL4
: Four-bit parallel CSRL latch.

V_CSRLN
: CSRL latch with active-low control.

ANALOG SIMULATION

Basics of analog simulation
A tour of the analog gates
Analog input waveforms

BASICS OF ANALOG SIMULATION

1. LOG and AnaLOG are simulating your circuit constantly, even

as you edit. But unlike the digital simulator, AnaLOG requires your circuit to be "complete" before it can simulate. That is, every pin must be connected to another pin, not left dangling. If the circuit is incomplete, i.e., there are still unconnected pins on some of the gates, AnaLOG will display a message to that effect on its scoreboard. As soon as the circuit is complete, the scoreboard will switch to "Simulation in Progress".

2. The gate that looks like a box with an arrow on it is the METER. To monitor your simulation, sprinkle meters liberally around on all of the interesting voltages. There is no limit to the number of meters you can have. Remember that the meter's red dot is at the tip of the arrow, so the tip must be exactly touching the wire or pin of interest. If the meter is unconnected or unhappy, it will display a "~" sign.

3. The "clock icon" is the round thing in the upper-right corner that looks vaguely like a clock. Its importance is in the two numbers directly below it. These are the current time and current timestep, respectively. At the beginning, these are both zero. As soon as the simulation begins, the current time will start advancing. How fast it is advancing is shown as the current time-step, and may vary depending on how fast things are happening in the circuit.

4. A common error is to watch, say, a capacitor charging, then go in and double the capacitance, and complain that it still seems to be charging "at the same rate." If you look at the clock icon, though, you will find that the timestep is twice what it used to be: AnaLOG can simulate the ramp function just as quickly in "real time," but the circuit is (correctly) half as fast in "simulated time." Make a habit of paying attention to the clock so that your observations have the correct frame of reference in time.

5. Although simulation happens continuously, it is sometimes necessary to "reset" the circuit back to its starting conditions. For example, suppose you remove a wire from a running circuit and the voltages that were in the circuit at the time are "impossible" for the new circuit. The simulator will work as hard as it can to figure out what to do, but may eventually give up and print a message. Another example is that you may want to reset various nodes to the initial voltages that you have programmed for them, as described in the next paragraph. In any case, all you have to do is select RESET in the Misc menu, or press the "R" key. This sets the simulation clock back to 0 seconds, and returns the voltages to their initial values.

6. If you open any AnaLOG gate for configuration, one thing you will find is a set of three attributes for every pin on the gate. These attributes are present voltage, reset voltage, and parasitic capacitance, respectively. Every pin has a slight parasitic capacitance to ground, initially 10fF. This capacitance must exist for the simulation to run, but you can set it lower if you find that it's in your way. The "present voltage" is an actual indicator of the voltage currently on that pin, and it changes while you watch during simulation. You can enter new

values here if you wish, but the simulator may not like you if you go around making instantaneous changes in the circuit's voltages. The "reset voltage" is used by the RESET command. If you leave it blank, AnaLOG will solve for an appropriate initial voltage at RESET time. Notice that Present and Reset voltages are for nodes, not gates: if you change a pin's reset voltage, then open up another gate connected to that same node, the reset voltage shows up there, too.

7. When you reset, you will notice that the timestep jumps down, usually to about 1E-18 seconds, then starts increasing. This is partly because AnaLOG is cautiously trying to discover how fast it can safely proceed in the simulation, and partly because it has to solve for unspecified initial voltages by simulating the brief charge-up that would occur when you powered up a real circuit from scratch. If you push RESET and the voltages seem to be frozen, don't depair -- wait for the timestep to get out of the femtosecond range!

8. The AnaLOG scoreboard has some words on its left and right sides. On the left is Memory, with indicators Set and Erase. If you tap the word "Erase," all of your "reset voltages" will be erased. You might do this if editing has caused your reset voltages to be no longer appropriate, and you want AnaLOG to solve the circuit from scratch. If you tap the word "Set," all of the present voltages in the circuit are copied into the reset voltages. For example, you might turn your input waveforms off, let the circuit settle to a stable state, then push Set so that next time you RESET, the circuit will start out at that stable state and you won't have to wait again.

9. AnaLOG is normally very cautious about voltage changes. If it is going along at a particular timestep and some node suddenly starts changing more than about .1V per timestep, AnaLOG will reduce the timestep and simulate in smaller chunks. This is because trying to simulate fast voltage changes all at once results in ugly and inaccurate waveforms.

10. Sometimes all you are looking for is a DC point anyway, and you don't care how ugly your transient waveforms are. In this case, you can touch the word "Relaxed" on the scoreboard. This tells AnaLOG to take timesteps as large as it can regardless of how fast the voltages are changing. Touch "Exact" to return to an accurate simulation. Often what people do is touch Relaxed right after a RESET so that the initial charge-up proceeds quickly, then touch Exact when the timestep gets up to the picosecond-nanosecond range.

A TOUR OF THE ANALOG GATES

1. Clock Icon (TIME). This displays the current time and timestep. It also has a square in the center that changes color to indicate how the simulation is going:

Black	Simulation is turned off
Red	Computing the next timestep

Yellow A timestep has been computed

If you configure the clock icon you will find that it can display a few other time-related values, of interest mostly to AnaLOG implementors only. Note that TIME is not actually part of the analog simulator---you can use it for the digital simulator, too, but it's not very useful there since the digital simulator has such a crude concept of time.

2. VDD and GND. Global power supply. VDD defaults to +5V. You can change the value of Vdd in the NUMBERS gate, but Gnd is always zero, and Vdd is always positive.

3. Terminals (TO, FROM). These are the two arrow-shaped gates in the Catalog. They are equivalent except for appearance. They are used for putting names on signals: to name a signal FOO, connect a terminal to it, then tap the space next to the arrow and type FOO. If there are several terminals in the circuit with the same names, they are all electrically connected as if by wires. Signal names are also used with the Scope mode, to be described below.

4. Scoreboard (NUMBERS). This gate serves many purposes. On the screen it displays whether simulation is proceeding and why; it gives controls for set/erase/exact/relaxed modes; and, when you tap the center bar, it lets you configure AnaLOG's "global" parameters, such as the value of VDD and the maximum allowable timestep. More on these parameters below.

5. Transistors (NFET5, PFET5, PFET6). Typical MOSFETs, with models accurate enough for circuits in Carver Mead's Analog VLSI and Neural Systems book. Compared to SPICE, they are between levels one and two in complexity. AnaLOG transistors are tuned for a 2 micron process, to be most accurate in the subthreshhold regime. This makes them perfect for neural network projects, but only roughly accurate for, say, simulating digital circuits.
For the simulation to be more accurate in strong inversion, the Beta of the NFETs to be 59uS/A and the Beta of of the PFETs to be 25uS/A; these parameters result in a close match to the simple
above-threshold MOS model presented in most textbooks.
The AnaLOG transistors are completely symmetrical between source and drain. PFET6 is a four-terminal transistor, for simulating PFETs with floating wells. Bipolar transistors and diode (NPN1, PNP1, DIODE) are experimental Ebers-Moll models. NSPC1 and PSPC1 are experimental FET models contributed by Bhusan Gupta. In addition to simulating in analog, they are perfect for use with the SPICE netlist tool, LOGSPC (see LOGSPC section)

6. Capacitor (CAPFLOAT). Typical capacitor, defaults to 1pF. Notice that, like all circuit elements, it has a slight

parasitic capacitance to ground as well as the capacitance between the two leads.

7. Resistor (RESFLOAT). A plain linear resistor. For a more VLSI-like resistor, see HRES below.

8. Transconductance amplifiers (OPAMP, WRAMP). These are simulations of the narrow- and wide-range amplifiers in Carver's book. Their behavior is modelled after the circuits at the transistor level (as modelled by AnaLOG), at least if you use them in the normal range of voltages (between 0 and 5V). A little red light turns on if the voltages go outside the range in which the model is a safe approximation for the "real" circuit. Note that the "Iset" pin at the bottom wants a voltage, not a current. If you want to set the current manually, connect a voltage source to the pin and also to a standalone transistor, then measure the current through the transistor.

9. Rectifiers (HWR, FWR). These are half-wave and full-wave rectifiers. An HWR models an OPAMP driving a two-transistor current mirror; an FWR models is like a pair of HWR's.

10. Horizontal resistor (HRES). This big ugly thing is the "horizontal resistor" circuit, which predates the circuit in Carver's book. Obsolete for new designs.

11. Ganglion (GANGLION). This is the basic ganglion circuit, as described in Carver's Book.

12. Voltage/current meter (MMETER). This meter displays the voltage on whatever wire or pin it is connected to. If you configure it you can change it to a current meter, in which case it measures current into the gate from whatever pin it is connected to. (If you connect a current meter to the middle of a wire, it doesn't work because the direction of the current is not well-defined.)

13. Special magic current meter (ISCOPE). This is a widget for use in plotting currents in Scope mode, and will be discussed in the later section on Scope mode.

14. Voltage source (VDIFF). This gate creates a differential voltage between its two pins (the bottom pin is typically connected to ground). It has a finite output resistance (initially 50 ohms), so it is not an "ideal" source in the sense of SPICE. You can use it as a simple DC source with a certain voltage (set by configuring the gate), or you can set to output a pulsed or sinusoidal waveform. This is all described below, under Input Waveforms. Voltage sources display the current-limiting condition by lighting up a

little red light, just like the ones on the lab bench! They also have a switch, in the lower-right, that you can switch to freeze the waveform in time.

15. Current source (IDIFF). Basically a current-sourcing version of the voltage source. It can generate the same kinds of waveforms. It voltage-limits if the voltage across the pins goes below the "crowbar voltage," 0.1V by default.

16. Step generator (STAIRS). A voltage source generating a stairstep function, which moves from one voltage to another in a certain number of steps, then repeats.

17. Voltage switch (VSWITCH). A voltage source that can be manually switched between "on" and "off" voltage levels. This one also has a tiny switch on its face, this time to select which voltage is generated. It can also be set in "monostable" mode, in which a tap on the switch will turn it on for a certain period of time, then back off again automatically. Note that voltages here are implicitly relative to ground.

18. Current switches (ISWITCH1, ISWITCH2). Current-sourcing and current-sinking versions of the VSWITCH. The current source (arrow pointing out) is like an IDIFF connected to VDD, and the current sink (arrow pointing in) is like IDIFF connected to GND.

19. Piece-wise linear two-terminal element (PWL, contributed by Harold Levy). Implements an arbitrary two-terminal element, with an i-v transfer curve. Cd to the log/lib directory, start analog, and load the file pwl-test.lgf to learn more.

20. Resonant tunneling diode (RTD, contributed by Harold Levy). Models a class of quantum well devices. If you do research in this area and want more details, contact har@caltech.edu.

ANALOG INPUT WAVEFORMS

1. DC. This is just a constant amount of voltage or current.

2. Pulse. Depending on how you set it up, this can be a square wave, triangle, or sawtooth. Referring to the attributes for Pulse mode,
the output waveform starts at "initial voltage," then, after "delay time" (plus "reset time" for the first cycle), it begins ramping to "pulsed voltage," at a rate determined by "rise time." It stays at that voltage for "pulse width," then ramps back to the "initial voltage" in "fall time." The cycle repeats the the period
specified. NOTES: If rise and fall time are both relatively small, you get a square wave. If one is large and the other small,

you get a sawtooth. If both are large, and the pulse width is zero,
you get a triangle.

3. Sine. This generates a sine wave, with the frequency, offset, and amplitude desired. The "phase" is specified in degrees. If the "delay" is nonzero, the wave waits that amount of time after RESET before it begins to change.

4. Stairs. Starts at "initial voltage," waits there for "time per step" time, then ramps to next voltage in time "rise/fall time." This occurs "number of steps" times, for a total of "steps"+1 voltages between "initial voltage" and "ending voltage," inclusive. After the cycle, it ramps all the way back to "initial voltage," again in "rise/fall time," and starts over. If "initial delay" is nonzero, the waveform remains constant for that long before beginning.

5. Bistable switch. If the switch is "off," output is "initial voltage." If it is "on," output is "pulsed voltage." When the switch is changed, output switches from initial to pulsed voltage in time "rise time," or from pulsed to initial voltage in time "fall time" (regardless of whether the voltage was actually increasing or decreasing). On RESET, the switch is set to on or off according to "reset state."

6. Monostable switch. Output is normally "initial voltage." When switch is tapped, output ramps to "pulsed voltage" in time "rise time," holds for "pulse width," them ramps back in time "fall time."

OTHER SIMULATION FEATURES

Simulation goodies
"Scope" mode

SIMULATION GOODIES

1. If there are any circuit elements with unconnected pins, AnaLOG will refuse to simulate. If you meant to leave the pin unconnected, you must connect it to something like a "dummy" capacitor to ground. If there are unconnected gates, the NUMBERS icon will display a message of the form "OPAMP is unconnected". Digital LOG doesn't mind unconnected pins on gates.

2. If you give a ":DIM" command, AnaLOG gates will be drawn in dim colors until they are fully connected. Another ":DIM" turns this mode off. This command only works with the analog simulator.

3. Select Probe in the Cursor menu (or press the "." key) to enable Probe mode. The cursor changes to a different style arrow; when you move this arrow over a gate or wire, a message below the AnaLOG NUMBERS icon shows the internal state of the gate and/or

node. In Digital LOG, the lines at the bottom of the screen light up according to the state of the node being probed. Probe mode also displays the name of a gate being touched. Select again to turn the mode off. You can also press "e" or "x" ("Examine") for a temporary Probe mode which turns off automatically when you click the right button.

4. Select Glow in the Cursor menu (or press the "g" key) to enable Glow mode. Wires "glow" in colors according to the voltages they carry. In AnaLOG:

Black	means at (or below) ground.
Dark red	means within one transistor threshhold of ground.
Red	means an intermediate voltage.
Pink	means within one transistor threshhold of VDD.
White	means at (or above) VDD.
Green	means the wire is not driven by an analog gate.

In digital LOG:

Black	means the wire carries a Zero.
Red	means the wire carries a One.
Green	means the wire is not driven.

Glow mode may be left on at all times, though it may make the simulation a little slower.

5. Select Simulation in the Misc menu (or press the "o" key) to turn the simulator off or back on again. When simulation is off, the gates and nodes of the circuit appear to freeze at their last simulated values. Turning the simulator back on resumes the simulation.

6. Press the "t" key to simulate for one time-step, then stop. You can press t as many time as you wish to see the simulation evolve, then press o to turn full-speed simulation back on.

7. LOG tries to balance simulation speed against response time for the mouse and keyboard. If you let go of the mouse and don't press any keys, LOG slowly shifts the balance to prefer simulation. When you again use the mouse or keyboard the balance reverts to favoring editing. You can press the "f" key to switch immediately to "fast" mode (preferring simulation over editing).

"SCOPE" MODE

1. Numbers are great, but real people like to see plots of their output waveforms. In LOG, the mechanism for this is called Scope mode. To use it, first connect terminals (TO or FROM) to each signal to be monitored and give the signals unique names, then select SCOPE in the Misc menu, or press the "s" key. The scope can display both digital and analog signals.

2. The Scope screen consists of a large field of dots representing the "divisions" on a normal scope face, an area to the left of

the dots where signal names can go, and some menu words at the bottom. Use the mouse as usual to select menu words for the functions that you want. In particular, touch QUIT (there's one on each side, for your convenience!) or hit <CNTRL>-C to return to the circuit diagram.

3. To display a certain signal on the scope, just type the name of the signal and press ENTER. The name will appear somewhere at the left edge. You can use the mouse to press and drag the name to any vertical location. To remove a signal name, drag it off the top or bottom edge.

4. Initially, the scope is not "triggered," so nothing is displayed. To begin taking data, touch Trigger. The scope also automatically triggers when you reset the simulation; there is a handy Reset on the menu too. A few moments after you trigger the scope, lines to the right of the signal names will begin to appear.

5. In all probability, the lines will be absurdly wide or ridiculously thin. In other words, the time scale is wrong. To fix this, press the "<" or ">" keys to shrink or expand (respectively) the plot until it fits nicely on the screen. You can also use the arrow keys to scroll around the plots in
time;
by scrolling and expanding you can focus in on any part of the simulation history.

6. Touch Config to change the various Scope mode parameters. You will find here the seconds-per-division and seconds-at-left-edge parameters that the zoom and scroll commands manipulate graphically; you can set them numerically if you prefer. The current time and timestep are displayed for your reference in setting these values.

7. Scope mode has a limit on the number of data points it will take before stopping, initially 2000. You can set this in the Config screen, but beware that if you set it too high AnaLOG will run out of memory and lose ALL of your data!. This screen also displays the number of data points taken so far. What happens when this reaches the maximum depends on the "trigger mode" attribute:

Manual:	Scope starts taking data when you touch "Trigger," and stops when you touch "Trigger" again, or when all the data points are used. Beware that when memory fills, there is no way to resume taking data except by retriggering and losing all of your old data! LOG prints warning messages when you are about to run out of data points.
On Reset:	Like Manual, but also triggers automatically on RESET. This is the default mode.
Triggered:	Like Manual, but also triggers automatically when a triggering pulse is detected on the signal that you specify. Not relevant for

AnaLOG, because triggering on analog signals has not yet been implemented.

Continuous: Triggers manually or on RESET. When data memory is full, old data points are removed from the left and new ones added to the right. Never stops triggering unless you turn it off by hand.

On Reset mode is typically used when you are interested in the first data points of a long simulation; Continuous mode is for when you are interested in the most recent data points.

Note that the word "Trigger" is lit whenever the Scope is taking data, and unlit otherwise.

8. Normally, Scope mode records exactly one data point per timestep returned by the simulator. The "minimum timestep" and "maximum timestep" attributes can modify this. If you would like to conserve memory and take data points only at, say, 100us intervals, you could set "minimum timestep" to 100us. Scope mode will then record a data point only if it's been at least 100us since the last point was taken. The other attribute, "maximum timestep," does not work with the analog simulator.

9. When you touch (instead of dragging) a signal name on the left edge, you Configure the parameters for that signal's particular trace. In AnaLOG, you can set volts-per-division, plus an "origin" voltage for plotting small signals (for example, if you have a signal that varies slightly around 2V, set the origin to 2V and the scale to, say, 50mV per division). You can also change the color of the trace, very helpful for plotting several traces on top of one another. You can also set logarithmic mode, in which case you specify "Vzero" which is the voltage to plot at the origin, with one decimal order of magnitude per division above and below.

10. To plot an AnaLOG current, you must take an ISCOPE gate, which is the funny-looking meter with a hook coming out the back, connect it to the pin to be measured, and connect a terminal (TO or FROM) to the "hook." Now, when you monitor that name in Scope mode, you will measure Amps instead of Volts. Note that the "hook" pin of the ISCOPE is a special "current-mode" data type, and any attempt to connect it to a normal analog signal will fail. If an ISCOPE ever appears to undergo spontaneous combustion, it is because you put it down with the hook touching an analog wire.

11. If you try pressing down the mouse anywhere in the grid of dots, you will get a horizontal and/or vertical line whose meaning depends on two indicators at the bottom of the screen:

Absolute:	Display the value of time, or of the currently-selected signal, that corresponds to the mouse position.
Delta:	Display the difference in time or signal value that corresponds to the distance between the current mouse position, and the position at which it was first pressed down.
Value:	Display the value of the current signal as of the time that corresponds to the mouse position. For example, you move the mouse to the "2.3ns" point, and the Scope displays the value that the current signal had at that time.
Slope:	Display the slope of the line between the starting and current mouse positions, where rise is measured in the units of the current signal, and run is measured in seconds.
Time	Display absolute or delta time, in seconds.
Freq	Display delta time, in Hz (= 1/seconds).
Signal	Display the value of the signal whose name is highlighted, in Volts or Amps. As you touch this word, each signal is highlighted one by one. You can also select a current signal by tapping its trace on the screen.

12. The word ON is a "power switch" for the simulation. If you touch it, it changes to OFF and the simulation halts. When you touch it again, simulation continues. (In the circuit-diagram screen, this function is called Simulation in the Misc menu.)

13. When you touch Dump, the Scope writes out all of the data into a file in a standard format. You may want to do this in order to use a more hefty data analysis tool with data generated by AnaLOG. The CNS 182 "View" program is one such tool, with a special "aload" command for loading AnaLOG dump files.

____

PLOTTING

Plotting a circuit
Plotting your data
Customizing your plots

PLOTTING A CIRCUIT

1. To make a hardcopy plot of your circuit, press "p" or select

Plotting in the Misc menu. You get a screen with an (admittedly ugly) view of what your circuit will look like on paper.

2. To zoom in on an area, just use the mouse to sweep out a rectangle around that area. Press Zoom out to return to viewing the full circuit. The Zoom feature uses "markers," which are little red brackets that show up in the circuit diagram. You can move the markers around in circuit-editing mode, too, and turn them on or off with the Markers command in Frills.

3. Touch Config to configure the plotter. You can change the font that is used for labels (there is a wide selection of fonts available, though most of them are silly). You can select how the plot will be oriented on paper (default is to let LOG choose the orientation to make it come out as large as possible). You can set the size of the paper if your plotter is too primitive to know that for itself. (For the laser printer, the printer always determines the paper size itself.)

4. Touch Options to configure the circuit-diagram plot. The only interesting attribute here is the size of the solder dots (oh, boy!).

5. When you are all ready press File and away it goes, to the system Postscript printer. The PLOT button used to control pen plotters directly, but now a better way of using pen plotters involves plotting into a file (see 6 below) using HPGL. [NOTE: The Unix version is not able to drive pen-plotters directly.]

6. You can also plot into a PostScript file, for inclusion, say, in a TeX document. Set the file name in Config, then touch File.

PLOTTING YOUR DATA

1. Take data on all interesting signals using the Scope screen.

2. To get a high-quality plot, touch the word PLOT on the SCOPE screen.
   This switches to an alternate menu of commands.

3. The word "Y-axis" is lit up, meaning that if you touch any signal name or signal trace, or the word "Time," or type any signal name or arithmetic expression involving signals, then that signal will go on the Y axis of the graph.

4. When "X-axis" is lit, you are selecting what should go on the X axis of the graph. After both axes have been specified, you can change your mind by touching one of the two words, then touching a new signal.

5. Touch Plot to run the plotting program.

6. In the plotting program, touch Config for general plotting options,
   or Options for options specific to data plotting.

7. In the Options screen, you get function name, plus label, units, log/linear, and minimum and maximum values for each axis. If you don't specify a min and max, they are calculated from the data. You can enter any arithmetic expression as the function; variables in the expression may be any signal name from the Scope screen, or Time, or one of A through F. These last are user-defined parameters you can change in the Variables screen.

8. You can also specify line style, and a symbol to plot on data points.

9. Normally the axis labels are drawn at a larger size so that you can read the numbers and exponents. When plotting on paper, the axis labels are regular-size.

10. Zoom out just clears the min and max for both axes to blank.

11. Touch Plot to send the current graph to a local plotter.

12. Touch File to send the current graph to a file or device, as described in the Config screen. Default is the Apple LaserWriter.

13. The simulation is still running while you are in the plotting tool. If the scope is still triggered, the set of data plotted will be those taken as of the time the screen was last refreshed, or the Plot button was pushed. Thus if you just sit and press the space bar, the plot will be slightly different each time. If this is confusing, we suggest turning the simulator off first by changing ON to OFF in the Scope menu.

14. Touch QUIT to return to the Scope.

15. For more advanced data munching, touch instead "Dump" from the SCOPE menu, then give some file name, by pressing the "!" key) and run the "View" program also written at Caltech. This program has a "load" command which can read the Dump file, using signal names as curves. (Note that the Dump command appends to the dump file if it already exists. "Load" will take the most recent curve if the file contains several by the same name.)

CUSTOMIZING YOUR PLOT

1. When LOG starts up, it reads lplot.cnf in the /lib directory (ask your local anaLOG maintainer), which contains certain commands for setting up the plotting system. This file in turn reads pens.cnf, which has mappings of colors onto pen numbers. Each of these commands begins with "lplot:", which signifies that the command is intended for the LOG plotting tool.

2. You can give these commands yourself, either while LOG is running

by pressing ":" and then typing, e.g., "lplot: size B", or by including such a line in your own "log.cnf" file.

3. The most interesting configurable thing for the plotting tool is color. LPLOT uses a two-step method for determining color. Each kind of object has a particular "color-name". These color names (like GATE or WIRE) are looked up in the color-name table to produce a pen-name (like RED or BLACK), which in turn is looked up in the pen-name table to get the actual pen number to use for pen plotters.

4. The following commands may be used after the word "lplot:".

```
LPLOT: COLOR  color-name  pen-name
          This maps a given color-name onto a given pen-name.
          Standard pen names are "black", "red", "green",
          "purple", "blue", "yellow", and "none" (which
          suppresses drawing of that color).  You can also
          specify a second color-name in place of a pen-name,
          in which case the first color-name uses the same
          pen as the one which was recorded for the second.


LPLOT: COLORS
          Display a list of all color names that have been
          recorded in "LPLOT: COLOR" commands.


LPLOT: CONFIG
          Open up the plotter configuration attribute screen.
          This is the same as touching "Config" while in the
          plot mode.


LPLOT: DOTS  gate-name  gate-name  ...
          Undo the effect of previous "NODOTS" commands for
          the specified gates.


LPLOT: FONT  font-file
          Use the specified font file for text.  The default
          file is "lplot.font" in the /lib directory for log.


LPLOT: INVISIBLE  gate-name  gate-name  ...
          Add the specified gates to the list of gates which
          do not show up in plots.  By default, TIME, NUMBERS,
          and DIGH are on this list.


LPLOT: LABELFONT  font-number
          Specify the font number to use for text labels in
          plots.  For the standard font file, this should be
          a number from 1 to 13.  Common numbers are 1 for a
          very simple font, 2 for a standard Roman font, and
          3 for Roman italics.  Default is 2.
```

LPLOT: NODOTS gate-name gate-name ...

Add the specified gates to the list of gates for which extra solder blobs will not be generated. By default, only the digital LED gate is on this list.

LPLOT: PENCOLOR pen-name pen-number

This specifies the mapping of pen names (such as RED or BLACK) onto pen numbers for the plotter. The standard configuration file "pens.cnf", which is included by all the standard "log.cnf" files, contains good default PENCOLOR commands.

LPLOT: PENS

Display a list of all pen names that have been recorded in "LPLOT: PENCOLOR" commands.

LPLOT: PLOTTER hpib-address

When using a pen plotter on the HPIB bus, this command specifies the HPIB address of the plotter. Default is 5.

LPLOT: SIGFONT font-number

Specify the font number to use for signal names in plots. (See LABELFONT command.) Default is 2.

LPLOT: SIZE paper-size

Set the size for paper in pen plotters. Other output devices, such as the laser printer, compute the size on their own. Size A is standard 8.5" by 11" sheets; size B is about 11" by 17". Size AUTO means to compute the size automatically.

LPLOT: VISIBLE gate-name gate-name ...

Remove the specified names from the list of INVISIBLE gates.

5. Here is a list of gate-color names used by LPLOT. Where an alternate name is given, LPLOT looks up that color-name if the main color-name is not found. The ultimate default is pen-name BLACK. Except for SCOPE and BORDER, all of these apply only to circuit-diagram plots.

BORDER	Color of the dotted border on the screen that indicates the limits of the page. Default in "lplot.cnf" is RED.
DASHBOX	(Alternate: LABELTEXT.) Color of "yellow boxes" (as gotten from the Frills menu).

GATE	Color of regular gates.
LABELTEXT	(Alternate: DASHBOX.) Color of labels (as gotten from the Frills menu).
NODE	(Alternate: GATE.) Color of node names, drawn when requested by the Options screen for gate plotting. Node names are invented by LPLOT and do not correspond to node names used by any other program.
OFFLED	(Alternate: GATE.) Color of digital LED's and other readouts which are "off". Default is YELLOW.
ONLED	(Alternate: GATE.) Color of digital LED's and other readouts which are "on". Default is RED.
SCOPE	Color of data plots.
SIGNAL	(Alternate: GATE.) Color of signal names in TO and FROM.
SOLDER	(Alternate: WIRE.) Color of solder blobs.
WIRE	Color of wires.

LOG-TO-NTK CONVERSION

Overview
LOGNTK commands
Identifying things
LOGNTK messages
Notes

OVERVIEW OF LOGNTK

1. LOGNTK is a program to convert diagrams in AnaLOG (.LGF files) into network descriptions of circuits (.NTK files). When you give the ":LOGNTK" command in AnaLOG, it reads the diagram on the screen, treating the AnaLOG or Digital LOG transistors as NTK transistors, other Analog gates as NTK cells, and certain labels as LOGNTK commands. Only labels surrounded by < and > or in quotation marks are significant to LOGNTK.

2. LOGNTK manages hierarchical definitions. The notation used for hierarchy is the same as that used by the Digital LOG simulator. LOGNTK sees the world as cells; each cell turns into one NTK file. To define a cell, draw a circuit diagram, add a label which consists of the cell name enclosed in quotes (as in "Fred"), then define the cell's connectivity either with a template gate or with a port list label. To put several definitions on a page, enclose

each definition in a dashed box (from the Frills menu). All of this
is described in the documentation for the digital simulator.

3. One difference is that while in Digital LOG you normally use INST gates (like INST3), in LOGNTK you normally use "generic instance" GINST gates (like GINST3) instead. LOGNTK recognizes both families of instance gates, but the INST gates can connect only to digital components, whereas the GINST gates can connect to anything.

4. A complete project in LOGNTK may include any number of cells. The top-level cell describes the whole circuit and includes instances of the other cells, which may themselves contain instances, etc. Anything goes, as long as no circular references exist among the instances. At the bottom of the hierarchy are transistors and/or primitive gates. Examples of transistors are the AnaLOG gates NFET5 and PFET5, and the Digital LOG gates V_NFET and V_PFET. Examples of primitive gates are all the A_ gates from the ACTEL gate library.

5. Each NTK file produced by LOGNTK is completely self-contained. When you LOGNTK a cell that has instances of another cell, the program copies the definition of that cell into the output file. An NTK file consists of any number of cell definitions, followed by a "main program" which calls the outermost cells. If your top-level LOGNTK page does not include a cell name, that page defines the "main program" for the project. If you do name your top-level page, LOGNTK puts that page in a cell as usual, and then creates a dummy main program which "calls" that cell once. Most simple programs like NETCMP, the NTK file comparator, do not care about this distinction.

6. When you give the ":LOGNTK" command, LOG converts the definition on the current page into an NTK file. The file name is usually derived from the cell name, with ".ntk" appended. (The name of the LGF file that contains the circuit diagram is irrelevant.) If there are several cell definitions on the page, LOGNTK does each of them in turn. You can also pick a specific definition by typing ":LOGNTK cell-name", where "cell-name" is one of the definitions on the current page. (You can give a list of cell names, and the names may include "*" and "?" wildcards.)

LOGNTK COMMANDS

1. LOGNTK diagrams often contain additional commands in the form of labels. These labels are always surrounded by "<" and ">" brackets.
Any labels of this form, but with commands that LOGNTK cannot recognize, are simply ignored. LOGNTK assumes these commands belong
to some other program, such as Digital LOG's hierarchy compiler.

2. You can also include these commands in your "log.cnf" file, preceded by "logntk:". For example, the label <global clock*> defines global signals just for one cell; the CNF command

```
       "logntk: global clock*" defines global signals permanently.
       Finally, you could also press the ":" and type the
       "logntk: global clock*" command during your LOG session.

   3.  Certain arguments may include wildcard characters.  LOGNTK uses
the
       same kinds of wildcards as Unix: "?" matches any character, "*"
       matches any zero or more characters, and "[ABC]" matches either
       A, B, or C.  Also, most names in LOGNTK are case-insensitive, but
       LOGNTK still strives to duplicate the upper/lower-casing that you
       use into the output file, and always capitalizes all the uses of a
       name consistently.

   4.  Every LOGNTK definition needs at least some commands.  To create a
       named cell, you need a NAME command (or a quoted label; see
below),
       and either a PORT command or a template gate.  For an unnamed
       top-level cell, you need at least a FILE command.

   5.  Here is the list of LOGNTK commands.  The ":" following the
command
       names are optional.


       GLOBAL: signal-names
                   This command lists names which should be "global" to
                   the entire project.  For example, VDD and GND are made
                   global by the standard "logntk.cnf" file.
                   This means that if you create a cell which includes
                   VDD and GND gates, those nodes will automatically be
                   made into "hidden" ports on the cell.  Instances of
                   that cell will gate VDD and GND wired to them
                   automatically.  All you have to do is used VDD and GND
                   naturally, and let the system keep track of it for you.
                   You can declare your own TO/FROM signal names to be
                   global in the same way.  The "signal-names" are case-
                   insensitive, and may include wildcard characters.


       PORT: signal-names
                   This command gives the list of ports in the cell.  It
                   may be used in place of a "template" or "example" gate.
                   Port lists are supported for historical compatibility,
                   but future designs should probably use template gates
                   instead.

                   Each name in the list should correspond to a signal
name
                   in the circuit diagram.  VDD and GND refer to the VDD
and
                   GND circuit symbols.  The position of a name in the
port
                   list, not counting global names, corresponds to the pin
                   number of the associated instance gate.  For example,
if
                   the .GATE picture for an inverter has the input as pin
1
```

and the output as pin 2, then the appropriate command for
the inverter's circuit diagram would be:

```
      <port: in out vdd gnd>     or
      <port: gnd vdd in out>     or whatever.
```

The positioning and ordering of global names in the list
is irrelevant, but any global signal that is used by the
cell or by its sub-cells must be present. (When you use
a template gate instead of a port list, LOGNTK manages
all this for you.) If you give a ":LOGNTK" command and
their is neither a port label nor a template gate in the
definition, LOGNTK will sort the signal names into
alphabetic order and create a PORT label. You will
probably need to edit this command to put the ports in
the proper order.

If several PORT labels appear in the cell definition,
they are appended together in an undefined order. If
you don't mind this uncertainty, LOGNTK can handle an
unlimited number of ports.

ORPHAN: gate-names

This command gives the gate(s) which are used to
represent "orphan" nodes, i.e., nodes that are not
connected to any transistors or subcells. By default,
the built-in gate CIRC1 is considered an orphan gate, so
that a circuit with five CIRC1's will have five orphan
nodes in its NTK file. The gate-names may use wildcards.

IGNORE: gate-names

This command gives a list of gates to be ignored. This
is rarely necessary, because if LOGNTK sees a gate which
has not been defined as a transistor or cell, and for
which no "gatename.NTK" file exists, then it will
automatically ignore that gate with a warning message.
This command suppresses the warning. The gate-names
may include wildcards.

NAME: cell-name

This specifies the name of the cell in the NTK file.
This is an alternate form for the quoted "cell-name"
label. There must be exactly one such label for each
cell.

FILE: file-name

This specifies the name of the output file. The suffix ".NTK" will be added if necessary. If you omit the
FILE
command, the file name will be taken from the cell
name.
(Every LOGNTK page must specify either a cell name or a file name, or both.)

CELL: gate-name cell-name file-name

This command links gates in the circuit diagram to the NTK files they represent. A good idea is to use the same name for the gate, cell, and file; if you give only one name in the command, that name will be used for all three. One time you might need to give the long form is if the NTK file is not in the standard directories, so you need to give a full path for it:

```
<cell: opamp tc_amp /lib/wollib/stdlib/opamp>
```

In this example, OPAMP gates, or generic instance gates programmed as OPAMP's, will convert to calls to the cell TC_AMP, which may be found in the .NTK file specified. CELL commands describing cells that are never instantiated in a file are ignored.

LIB: directories

This command names directories in which .NTK files for sub-cells may be found. LOGNTK looks first in the current directory, then in each of these directories in turn.

SIZE: number

This specifies the sizes of nodes. Currently, there is no way to override this for particular nodes. The default size of 1 should be sufficient for CMOS users. Most NTK-reading programs ignore node sizes, anyway.

STRENGTH: number

This specifies the strengths for transistors. This can be overridden as described for TRANS. The default strength is 2. Most NTK-reading programs ignore transistor strengths.

TRANS: gate-name type strength

This command declares transistor symbols. If you need to make your own transistor gates, you will need to use this command. The "type" is one of N, P, or D. The strength is optional; if it is a positive number N, then the strength of the transistor is N; if it is missing, the strength is the default value; if it is -N, the strength

```
                is found in attribute number N of the transistor
                gate, which must be an integer-valued attribute.
                The file logntk.cnf contains TRANS commands
                for all the standard LOG transistors.


     PRIMITIVE: gate-name  cell-name
                This command declares primitive gates, gates which
                should be converted into instances of cells which
                have no definition.  If the gate has N pins, then
                the output file will call the cell with N ports,
                in order of pin numbers on the gate.  For example,

                      <primitive: a_and2 and2>

                Causes the ACTEL gate A_AND2 to convert to a call
                to cell AND2.  Both names may contain wildcards;
                the actual commands in logntk.cnf are:

                      logntk:  primitive  a_*  *
                      logntk:  primitive  a_tribuf  tribuff

                which maps A_AND2 to AND2, A_DFC to DFC, etc.,
                and A_TRIBUF to TRIBUFF (a special case).  Note
                that more specific names come after more general
                names.  The command PRIMITIVE may be abbreviated
                as PRIM.


     TOP        Force this to be a top-level NTK file, that is,
                after the main cell has been defined, one "call"
                to the cell is also written into the file.  The
                NETCMP program can only read top-level NTK files.
                This mode is the default; you only need the TOP
                command to override an earlier LEAF command.


     LEAF       Force this to be a leaf-level NTK file, that is,
                only the cell's definition is included (plus
                definitions for any subcells used).  There is
                usually no reason to use LEAF, since most programs
                can read top-level NTK files and ignore the top part.


     NOTOP      Set a compatibility mode in which a LOGNTK command
                on a page with no NAME command (or quoted name label)
                will produce an error instead of creating a top-
                level NTK file.  If you are a stick-in-the-mud like
                John Lazzaro, you can put this in your log.cnf file.
```

IDENTIFYING THINGS

```
   1.  Cells and nodes in the NTK file are assigned names automatically
by
       LOGNTK.  The one exception is that if a node has a signal name
       associated with it, that name is used.  The :IDENTIFY command in
```

AnaLOG lets you find out what names correspond to what circuit objects. Type:

```
:IDENTIFY
```

and move the cursor over any node or gate. The associated name appears at the bottom of the screen. The IDENTIFY command is finished when you press the mouse left button down, or hit the <CNTRL>-C or SELECT key.

2. NOTE: If you change anything in the circuit between :LOGNTK and :IDENTIFY, the names chosen may not be the same for the two commands. This is true even if you changed it and then put it back the way it was.

3. You can also type:

   ```
   :IDENTIFY name name name ...
   ```

   to "light up" the gates or nodes corresponding to those names. You can press the space bar to refresh the screen afterwards.

LOGNTK MESSAGES

1. LOGNTK displays the following messages at the top of the screen during operation. If there is a "fatal" message, the output NTK file will not be complete.

```
Can't understand command: FOO
```

This command appeared in a LOGNTK: line in the LOG.CNF file, and it is not one of the command words listed above. Unrecognized commands in < > labels are ignored, since it is assumed that they belong to some other LOGNTK-like program.

```
Syntax error in TRANS command
```

The transistor-type argument is missing or wrong.

```
Cell file FOO.NTK not found     (fatal, unless next message
                                 is shown too)
```

The file named in a CELL command does not exist.

```
Using file FOO.NTK instead of file BAR.NTK
```

The CELL command listed a file which does not exist, but the "gatename.NTK" algorithm found the file. You should edit the label accordingly. (The above example corresponds to a command of <cell: foo xxx bar>.)

```
<cell: FOO>
```

This is a label added if the "gatename.NTK" algorithm worked and there was no CELL command for this gate.

```
           This just serves as a reminder that LOGNTK made this
           assumption for you.


No cell or file name!                             (fatal)
<file: (place file name here)>
           There were no <name: xxx>, <file: xxx> or "xxx" labels
           on the page.  LOGNTK assumes you are making a top-level
           file, and creates a <file> label which you can edit.
           If you are making a cell instead, throw away the <file>
           label and make a cell-name label.


No cell name!                                     (fatal)
<name: (place cell name here)>
           No name was specified for this cell.  You are given a
           label which you can edit to specify the cell name.
           This error occurs only in NOTOP mode.


More than one <name: xxx> or "xxx" label on the page!     (fatal)
           Multiple cell-name labels appeared, not enclosed in
           dashed boxes.


Cell FOO is recursively defined!                  (fatal)
           LOGNTK detects some, but not all, such errors.


Warning: Port FOO does not appear in the circuit
           A name in the PORT command does not correspond to any
           signal name in the circuit.  This may be because your
           subcells use GND and VDD but you don't explicitly use
           them here (not an error), or it may be because you
           forgot to label part of your circuit.


Signal FOO is missing from the port list          (fatal)
Global node FOO is not a port
           You used a cell that requires a certain global signal,
           such as VDD or GND, which you omitted from your own
           PORT list.  All global signals used by a cell must be
           passed down through all higher-level cells in the
           hierarchy.


Warning: FOO appears more than once on port list
           You included the same name more than once in a PORT
           command.  The NTK file format has no way to express
           that two ports of a cell are shorted together, so
           LOGNTK can not write an NTK file for such a cell.
           The later occurrence of the name on the port list
           is ignored.


Template has pins shorted together                (fatal)
           Analogous problem as "FOO appears more than once
           on port list", but for template-defined gates.
```

```
Warning: Template gate has no connections
           This is not strictly an error, but in NTK a cell
           with no ports is usually a sign of trouble.


Using port list: A B C D
           If the cell was defined with a template gate, this
           message shows the order in which the signals were
           assigned as ports, just for your information.


Port list ignored for top-level NTK file
           This is a top-level file (not a cell), but there
           was a PORT command present.  The PORT command is
           ignored.


More ports than pins in cell FOO                  (fatal)
Too few connections in cell FOO                   (fatal)
           The NTK file for subcell FOO listed more non-global
           ports than there are pins (on an instance gate
           created in LOGED) or connections (on a generic
           instance gate).  This could be because the .GATE
           and .LGF files for that subcell are out of sync, or
           because you forgot to list some of the ports as
           GLOBAL.


More pins than ports in cell FOO                  (fatal)
Too many connections in cell FOO
           The NTK file for subcell FOO listed too few non-global
           ports.


Warning: Gate FOO was ignored
           A gate of type FOO was ignored because no definition
           for it could be found, but that gate was not listed
           in an IGNORE command.


No nodes or gates called FOO
           An :IDENTIFY command gave a name which does not exist.
```

2. There are also a few "informational" messages which can be turned off by giving a ":VERBOSE OFF" command in LOG.

NOTES

1. LOG nodes which are not connected to any transistors or cells are ignored. Thus, you can not get orphan nodes just by drawing stray wires. You must explicitly use "orphan" gates as described above under ORPHAN.

2. Cells which pass "global" nodes to their subcells, but which do not actually use those nodes themselves, must include those global names in their own port lists. For example, if a higher-level cell does not use GND and VDD but its subcells do, you must include GND and VDD in this cell's PORT list. (Note that if you are using the recommended "template gate" method rather than writing a PORT list, this is all handled for you automatically.)

3. LOGNTK is case-insensitive, but case-preserving. That is, if you call something FOO in some places, and Foo in others, the names will be considered the same, and one of the names will be chosen for use everywhere in the output file. Which name is chosen is arbitrary, except that preference is given to usage in the PORT command. Note that LOG itself is case-sensitive in some ways, for example, TO/FROM signal names "Vdd" and "VDD" are considered to be different by LOG but this difference is not acknowledged by LOGNTK. This sort of disagreement can cause LOGNTK to produce strange or erroneous output files.

4. The "VDD" and "GND" symbols are exactly equivalent to TO's or FROM's with the names Vdd and Gnd, respectively.

5. In LOGED, you can use the CONNECT command to cause two pins to be connected internally. (For example, the crossing-wires gate CROSS2 and the toggle switch SW2 work this way.) If you use such a gate as an instance in LOGNTK, only the lowest-numbered of the connected pins corresponds to a port in the PORT list; the other pins in the connection group are skipped when being matched to ports, in the same way as global ports are skipped when being matched to pins. Once again, using a template gate to describe your ports handles this automatically.

LOG-TO-SPICE CONVERSION

Description
Logspc Additions

DESCRIPTION

Harold Levy of Caltech modified the LOGNTK program to create a new program, LOGSPC, that creates SPICE decks instead of .ntk files. The tool is used identically to LOGNTK, so the manual section above describes LOGSPC also. The command to invoke the tool is :logspc, and configuration information is in the file lib/logspc.cnf. The FET models NSPC1 and PSPC1, contributed by Bhusan Gupta at Caltech, are perfect for use with LOGSPC. These gates also simulate in analog; however, these models are considered "experimental" as opposed to "standard" models in analog, and are not supported by the chipmunk maintainers.

LOGSPC ADDITIONS

A few additional commands have been built into LOGSPC to better support SPICE capabilities. Note that the SIZE and STRENGTH commands of LOGNTK are ignored in LOGSPC, and that the following commands are used instead:

```
          NBULK: signal-name
          PBULK: signal-name
                    These commands assign explicit names to the implicit
                    bulk terminals for 3-terminal FET (e.g. p/n-FETs 4
and 5)
                    and bipolar (e.g. NPN1, PNP1) models.  The signal-
name
                    is output in capitals unless quoted.

          CHLEN: number
                    This command assigns an explicit channel length (in
                    microns) to FET models using w/l ratios (e.g. p/n-
FETs
                    4, 5, and 6).
```

Also note that as of LOGSPC version 0.5b the ability to extract VDIFF gates is provided. This feature may be shut off with the command

```
          LOGSPC: IGNORE VDIFF
```

THE LOGED PROGRAM

Introduction
List of commands
DRAW command
LABEL command
DEF command
Digital gate definition language
Text file format
.GATE file format

INTRODUCTION

1. The LOGED program is responsible for maintaining the libraries of gates stored in .GATE files. When LOG starts, the GATES commands in its configuration file specify which gate libraries to use.

2. The "loged" command may be given by itself or with a the name of a file to load initially. Additional arguments are interpreted as they would be for a LOAD command, as a list of gates to load from the file. Note that in LOGED terms, a "gate" is a kind of gate, such as NAND or 74138, not an instance of the gate-kind.

3. To exit LOGED, type QUIT. You will be given a chance to save the gates in memory if they have been modified. If the gates came from one file originally, LOGED uses that name. Otherwise, if no file was loaded or several files were mixed together, LOGED prompts for the file name to use.

4. At any time in LOGED there is a "current gate." Many commands operate on the current gate. The GATE, NEXT or PREV commands can be used to select the current gate. LOGED stores the following information for each gate:

   a. The name of the gate, up to eight characters (uppercase).
   b. The group number (library-screen page number), from 0 to 8.
   c. The gate's picture, including pins and grab box.
   d. The "simtype" number, telling which simulator "owns" the gate.
   e. The gate's definition or program, simulator-dependent.
   f. Connectivity information in case pins are hard-wired together.
   g. Miscellaneous flags affecting LOG's treatment of the gate.
   h. The gate's labels screen, shown when configuring the gate.

   Various commands described below manipulate all these parameters for the current gate.

5. Many commands accept gate-name arguments which may include wildcard characters. The '.' and '*' characters stand for exactly one, or zero or more, characters respectively. Use '%' or '?' (respectively) instead for wildcards which prompt yes-or-no for each potential match. A terminating ';' followed by digits matches only gates in the groups specified. For example, the pattern 'a*bc;23' matches only gates whose names begin with 'a' and end with 'bc', and which are in groups 2 or 3. If a list of several names is given separated by spaces, gates which match any of those names are accepted.

LIST OF COMMANDS

1. A complete list of LOGED top-level commands follows. Some commands are discussed in detail in later sections.

2. Command and gate names are case-insensitive. Commands can usually be abbreviated to one or two letters.

3. In addition to these commands, you can execute any Shell command by preceding it with a '!' character, or enter an interactive sub-shell by typing '!' alone. Type QUIT to the sub-shell to return to LOGED.

```
CONNECT  num num num ...
```

Connect the specified pins together in the current gate. When an instance of the gate is made in LOG, these pins will be forced to the same electrical node. For example, the CROSS gates in the standard library emulate crossing wires by having pairs of connected pins. The pins of a gate are divided into connectivity classes; initially every pin is in its own class. The CONNECT command with two or more numbers joins the classes of the specified pins. CONNECT with one pin number unconnects the pin, splitting it back into its own class. CONNECT with no arguments displays the current connectivity of the gate.

COPY new-name
Make a copy of the current gate under the specified name. All aspects of that gate are identical to the current gate, except the name. The new copy becomes the new current gate.

DEF
Edit the simulator's definition for this gate. Exact effect depends upon the current simtype of the gate. This command only works of the relevant simulator is available. The type 0, 1, and 7 DEF commands always work; type 16 and 32 commands work only if DIGLOG or ANALOG have been permed, respectively. (Note that ANALOG actually supports both simulators.)

DELETE
Delete the current gate. Given a name, deletes all gates which match the name.

DO name command-line
For each gate that matches the specified name, make it the current gate and execute the LOGED command shown. This command must not delete or rename the gate; to delete multiple gates, use DELETE directly.

DRAW
Edit the current gate's picture. Defined below.

DUMP
Print a "data sheet" for the current gate on the laser printer. If a file name is given, the PostScript for the data sheet is written to the file.

EDIT
Edit the current gate's information using the text editor, CAGED. Given a name, edits all gates which match the name. The information is written into a temporary text file as if by the WRITE command; CAGED is run on the file; when the editor returns, the file is reloaded as if by a READ command.

FLAG flag-name
Turn the specified flag on or off for the current gate. Flag names are case-insensitive and may be abbreviated.

NOFLIP   This disables flipping or rotation of

the gate. Tapping on the gate always
acts as if CNFG mode were selected.

TOGGLE The gate is a toggle switch. When an
instance is tapped, it always mirrors
about its (original) y-axis. The gate
can still be rotated in the menu area.

VISIBLE The gate is visible even in Invisible
gates mode.

NAMED The gate has an editable signal name,
like TO or FROM. Should be used only
for simtype 1 gates.

NRIGHT The gate's name is on the right, as in
the TO gate. Used with NAMED, above.

NOSOLDER When plotted, the gate should not have
solder blobs added for T-connections
involving the pins.

USER1 A user-definable flag. A simulator,
tool,
or individual gate program can interpret
this flag in any way desired.
Currently,
this flag is unused.

USER2 Another user-definable flag.

With no argument, this command just displays the
current
flag settings.

GATE Switch to a new current gate. If the specified gate
does
not exist, a new "blank" gate is created. (Often it is
easier to create a new gate by copying an existing,
similar one.) The NEXT command can be used to switch
to
a new gate with no danger of accidentally creating one.

GROUP n Set the current gate's group number to "n", an integer
from 0 to 8. With no argument, displays the current
group number. The group number determines which
library page the gate appears in. Group numbers for
the standard libraries are:

0	General-purpose gates
1	Generic digital gates
2	7400's series digital gates
3	(unassigned)
4	Analog gates
5	Digital VLSI gates
6	Digital ACTEL gates

```
                         7        (unassigned)
                         8        (obsolete gates)


       HELP       Run the Chipmunk Help System on this help file.


       LABEL      Edit the current gate's labels and attributes.  See
                  below.


       LIB        Display a list of all gates, or the gates which match
                  the specified names.


       LOAD  file-name  gate-name gate-name ...
                  Load the gates in a file into memory.  If gate-names
are
                  specified, only those gates which match the names are
                  loaded.  By default a ".gate" extension is added to the
                  name; to load from a textual file, specify ".text"
                  explicitly.


       MEMORY     Display the amount of memory free.


       NEXT       Switch to the next gate in alphabetical order, or to
the
                  next gate which matches the specified name.


       PREV       Switch to the previous gate in alphabetical order, or
to
                  the previous gate that matches the specified name.


       QUIT       (or EXIT.)  Leave the LOGED program.  If any gates have
                  been changed, you are given the option to save them
first.


       READ       Much the same as LOAD, except the default file name
                  extension is ".def", typically a file written by the
                  "Write" option of the digital hierarchy compiler.


       RENAME  new-name
                  Change the name of the current gate to the new name.
                  If a gate by that name already exists in memory, you
                  are given the option to replace it.


       SAVE  file-name  gate-name gate-name ...
                  Save all gates in memory, or only the specified gates,
                  into a file.  With no arguments, saves all gates back
                  into the original file.  A ".gate" extension is
appended
```

by default. If the extension is not ".gate", the file is written in a textual format readable by humans.

SIMTYPE n   Set the simulator number of the gate to "n". This is an integer from 0 to 255 which identifies the simulator responsible for the gate. Currently assigned simtypes include:

0	General-purpose, "inert" gates. Used labelling (e.g., ARROW) and connection (e.g., CRUNCH) gates.
1	Signal-name gates. Pin 1 of the gate is connected to a certain signal name, which is either hard-wired (as in VDD and GND) or configurable (as in TO and FROM).
7	Self-sufficient Pascal gates. The code which defines this gate is given complete access to LOG's simulator interface. For example, the TIME gate is a general-purpose simulation timing and performance monitor.
16	Digital gates. Simulation definition is written in a simple language specific to the digital simulator. This language includes a CALL statement for calling more general Pascal procedures to simulate the gate.
32	Analog gates. Definition consists of the name of a Pascal procedure for simulating the gate.
33	Analog current-mode simtype. This type is used internally by the analog simulator to support current meters.

The gate's simtype defines which procedure is called when you give a DEF command. This command is defined in greater detail below.

WRITE   Similar to SAVE, except the default file extension is ".text", and if no arguments are given, the current gate is saved in a file named after the gate.

THE "DRAW" COMMAND

1. The "DRAW" command puts you in a graphical editor mode for drawing the visible picture of the gate. It also allows you to place the gate's pins and "grab box".

2. To exit the DRAW command, press "q" or the <CNTRL>-C key. (On the Bobcat keyboard, this key is marked "Select".)

3. Gate pictures are made up of many kinds of objects. Initially, you are set to draw the usual cyan vectors (line segments). Tap the mouse at two locations to draw a new vector between those points (analogous to drawing a wire in LOG). Grab the body of the vector to move it; grab an endpoint to change its direction and size. Move the vector off the grid to delete it.

4. Tap the mouse on the color or shape name to change. Circles and ellipses are defined by a vector; an ellipse's diameters equal the sides of the bounding box of the vector, and a circle's diameter equals the larger of the two sides. Filled ellipses are also available. Filled polygons are defined by four points; to make a triangle, put two points in the same place; to make a larger polygon, combine several triangles or quadrilaterals. Rounded boxes are rectangles with circular arcs for corners. "Curves" are Bezier curves defined by four points; the two inner points govern the curve's slope at its endpoints. Text objects contain a string of characters of a particular size; LOG will draw the characters at that size if possible, but will not draw them at all if there is no font small enough. Point-markers and box-markers are invisible placeholders which a gate's program may use to specify the location of some dynamic display such as an LED.

5. Some aspects of a picture can not be edited graphically. These include the radius of the corners of a rounded box, and the size, orientation and contents of a text object. To edit the picture textually, you can press the "e" key. This is analogous to the EDIT command of LOGED; it runs the text editor CAGED on a temporary file containing the gate's definition. See below for a description of this format.

6. The gate's "grab box" is a dotted yellow box that starts out at the edges of the drawing grid. This box defines what portion of the picture is considered the gate's "body", for purposes of tapping and dragging the gate in LOG. It should generally be as large as possible, but must not enclose any pins. (If a pin were inside the box, it would not be possible to draw a wire to touch the pin.) To move the grab box, press on any edge or corner of the box.

7. The pins of a gate are numbered consecutively from 1. To create a new pin, grab one of the numbers on the left edge of the screen and drag it into place. Only one pin of a given number may exist at a time, and when the gate is saved, the pin numbers must be contiguous (if there are "n" pins, they must be numbered 1 through "n"). Pins are forced to multiple-of-five pixel grid lines, as
are
the user's editing actions while in LOG; this makes it easier for the user to connect exactly to the pins.

8. To move all objects in a given rectangular area, just press the mouse at one corner of the area (with no objects under it) and drag out a rectangle, then move the objects into place and press

the mouse again. This is analogous to LOG's MOVE command. To cancel the move, click the right button.

9. To copy another gate's picture, press "c", then give the name of the gate to be copied. The current gate's picture is replaced by the picture, pins, and grab box of that gate. Other aspects of the current gate (such as the name, labels, and definition) are not affected.

10. If you are drawing a gate which must look similar to another gate, you can press "s" to "shadow" that gate. Its picture will appear dimly in the background of the editing grid. To remove the shadow, press "s" and enter a blank name.

11. By default the editing grid is the exact size of a cell in the Catalog screen or menu area. To draw a larger gate, press "<" and ">" to zoom up or down. If the gate is larger than the default grid, it will be shrunk to fit in a Catalog or menu area space, but will be shown full-size in an actual drawing.

12. If you need more pins than there are numbers for on the screen, press "[" and "]" to change the range of pin numbers shown. Pin numbers may currently range up to 128.

13. To switch to the next or previous gate in alphabetical order, press the "n" or "p" key. On a Bobcat, you can press the Next and Prev keys. This is exactly like leaving DRAW mode, typing NEXT or PREV, and giving another DRAW command.

14. Other keys you can press are space bar to refresh, "!" to start a sub-shell, and "?" for help.

THE "LABEL" COMMAND

1. To edit the text shown when the user taps on a gate in CNFG mode, use the LABEL command. This command runs a simple text editor. To exit the editor, press <CNTRL>-C (or Select on Bobcats).

2. The labels editor lets you move the cursor with the arrow keys, delete or insert a line using the <CNTRL-F> and RETURN keys, insert characters, or erase characters with <CNTRL-G>. (Currently, there is no way to recover a deleted line, or to move or copy lines except by retyping them.)

3. Any line without a colon (":") is displayed as-is on the screen when the gate is configured. To include a colon in a label, use a double colon ("::").

4. If a line does contain a colon, it describes an attribute. Everything to the left of the colon is a coded string showing the name, type, and default value of the attribute. Everything

after the colon is shown as-is as the label for the attribute. Blanks are ignored between parts of the attribute information.

```
<name> <variant> <precision> <type> <default> ':' <label>
```

5. The name is optional, and must be enclosed in square brackets. Actually, several bracketed names may be used if for some reason the attribute must have several names. The code for a gate may refer to an attribute by its name, or by an integer index ranging from 1 to the number of attributes.

6. The type is a single capital letter. The following types are defined:

I	Integer. If a precision is given, the number is blank-padded to at least that width when displayed. If a default is given, it becomes the initial value of the attribute for newly created gates. If there is no default, the attribute is initially blank.
H	Hexadecimal integer. The number is zero-padded to at least the specified width; if no precision is given but there is a default, the precision is equal to the number of digits in the default.
R	Real number. The precision gives the number of decimal places shown on the number; if no precision is given the number will be shown in floating format. Large and small numbers are shown in scientific notation.
U	Real number with units. The default is of the form, `<units-string> ',' <default>` where the comma and default may be omitted if the attribute should be blank by default. The number is shown followed by the units-string; large and small numbers are shown using standard engineering notation abbreviations. The attribute: `[clock-freq] 2FHz,10000:Clock frequency::` will display its default as "10KHz".
F	Real number with units, scientific notation. This is just like "U" format except large and small numbers are shown in scientific notation with the units-string appended.
C	String attribute. The precision specifies the number of characters of length the string may have; with no precision the string length is unlimited. The default may be any string of characters, unmodified except that leading and trailing blanks are removed.
A	String attribute. To the user, this is just like a "C" type attribute with no precision specified. Internally, this is stored as a variable-length

string, whereas a "C" type attribute is stored as a fixed length string. Precision is ignored.

B Boolean value. The default may be "Y" or "N" (or "T" or "F") to specify true or false. If the default is blank (or "X"), the attribute is initially blank. If the default is "T" or "F" or "X", the attribute is displayed as "True" or "False". If the default is "Y" or "N" or blank, the attribute is displayed as "Yes" or "No".

V Variant. This attribute represents a choice among several named options. The default string is actually a sequence of comma-separated phrases. These phrases are numbered from 0. The precision actually specifies the default variant shown; with no precision, variant 0 is the default. If the attribute should be able to be blank, simply leave one of the phrases empty.

```
      1 V , Yes, No, Maybe : What do you think?
```

This attribute takes on one of the four values blank, Yes, No, or Maybe; it is initially Yes.

7. A variant name may be included in any attribute, followed by a semicolon. The name must be one of the options for a variant attribute that precedes this attribute in the list. The attribute will be invisible unless the proper variant was selected.

```
     V Square,Circle: Kind of shape::
     Square; R:Width::
     Square; R:Height::
     Circle; R:Radius::
```

8. If the type letter is preceded by the letter 'O', the attribute is optional and can be made blank by typing a blank line. An attribute with no default is always optional. If the attribute has a default and is not optional, then it must always have a nonblank value. (This is true even of string attributes.) Entering a blank line for a nonoptional attribute restores the default value.

THE "DEF" COMMAND

1. The DEF command edits the simulator's definition for the gate. What happens when you type DEF depends entirely on what simtype you have given to the gate. Also, for simtypes 16 and above, the relevant simulator must have been loaded into the computer already via a "perm" or "use" command.

2. For simtype 0, the gate has no simulation behavior and its DEF command does nothing.

3. For simtype 1, the DEF command displays the signal name current

programmed into the gate, then gives you the option to change this name if you wish. If you leave the name blank, the gate will not connect to any signal by default; presumably, you have set the NAMED flag so that a signal name can be entered at run time.

4. For simtype 7, the DEF command displays and allows you to change the name of the Pascal procedure responsible for the gate. This must be of the form, "modulename_procedurename". The writing of Pascal coded gates is beyond the scope of this manual; refer to Dave Gillespie's Master thesis, or examine the LOG sources for examples.

5. For simtype 16, you enter a simple text editor. Use the up arrow and down arrow keys to move, press <cntrl-K> and <cntrl-I> to delete or insert lines. Editing functions currently not implemented
implemented are RECALL, to restore a deleted line, and left arrow and right arrow to edit a line. Indentation and formatting is handled automatically. If LOGED detects a syntax error in your line, it converts it into a comment. However, note that some syntax errors are not caught but will produce code that may cause the simulator to crash or behave strangely. For a description of the LOG digital gate language, see below. Press <CNTRL>-C or <CNTRL-D> to exit.

6. For simtype 32, you are prompted for the name of a Pascal procedure
which takes responsibility for the gate, in much the same way as for simtype 7.

DIGITAL GATE DEFINITION LANGUAGE

1. Simple digital gates in LOG can be described entirely in the modest interpreted language described here. For more complicated functionality, for displays and user interaction, or for access to the gate's attributes, you must write Pascal code using a CALL statement. The interpreted language is usually referred to as "LOGED language," though it is really part of the digital simulator, not of LOGED proper. The language is also called Gate Description Language, or GDL.

2. A GDL program consists of a sequence of statements, displayed one per line. These statements are typically either assignments to pins or variables, or IF/ELSE/END constructions. GDL contains no loops, arrays, or arithmetic. The program is executed once per LOG time-step, from top to bottom, performing boolean operations on values in the gate.

3. Resources available to a GDL program are pins, notated as in "#3"; internal nodes, notated as in "##6"; and state variables, such as "A", "P", or "V26". A state variable is much like a Pascal variable; an assignment to it replaces the previously assigned value, and lasts until the next assignment. It is a 1-bit quantity
whose value is interpreted as a boolean logic 1 or 0. A pin or internal node is a true LOG node; it uses a five-valued logic

(One, Zero, undriven or "None", and weak One and Zero) and must be assigned to once per time-step. The assigned value does not show
up
on the node until the next time-step. If the gate uses only nodes (no state variables), the order of execution of statements within the gate is essentially arbitrary.

4. Pin numbers range from 1 to the number of pins on the gate. The 16 state variables A through P are always available. For internal nodes and additional state variables, you can use the INST statement described below. All state variables are initially Zero when the gate is created.

4. Each GDL statement must be one of the following:

```
<variable> = <expression>
            Assign the value of the expression to the variable.
            The previous value of the variable is replaced.  This
            is exactly like a variable assignment in a conventional
            language.  If the value of the expression's value is
            None, the variable is set to One.


<node> = <expression>
            Output the value of the expression to the pin or
            internal node.  If two statements (in the same or
            different gates) output conflicting values to the
            node in the same time-step, a conflict is registered
            on the node.  Outputting None to a node has no effect.


<node> = PULLUP   or   PULLDN
            Output a weak One or Zero (respectively) to the
            node.  This essentially registers a default value
            for the node; if no other GDL statement assigns a
            strong value to the node in this time-step, the
            weak default will be used.  A strong value overrides
            a weak value without conflict.  A conflict will be
            registered if the node is weakly pulled to One and
            Zero simultaneously.

<node> '<' <expression>
            Output an open-collector value to the node.  If
            the expression's value is Zero, it is driven to the
            node.  If the expression's value is One or None,
            the node is left alone.


IF <condition> <expression>
   <statements>
ELSE
   <statements>
END
            If the condition is satisfied, the first set of
            statements are executed.  Otherwise, the second set
            (if any) are executed.  The conditions available
            cover all possibilities of One, Zero, and None
```

values for the expression:

IF x	True if x=One or None.
IFONE x	True if x=One.
IFZN x	True if x=Zero or None.
IFZERO x	True if x=Zero.
IFCONN x	True if x=One or Zero.
IFNONE x	True if x=None.

CALL <procedure-name>
Call the Pascal procedure specified. The name has the usual "modulename_procedurename" form. If there is are CALL statements in a gate's procedure, those procedures are called in order when the gate is simulated, drawn, tapped, created, destroyed, copied, configured, etc.

INST <num-nodes> , <num-vars>
This statement, if used, must be the first statement of the program (before even comments and blank lines). The statement "INST 17,6" reserves 17 internal nodes, ##0 through ##16, and 6 additional state variables, V0 through V5, for the gate. If the number of
variables
is zero, it and the comma can be omitted.

5. GDL statements may also be blank lines or comments. A comment line begins with a "#" not followed by a digit. Blank lines and comments do have a small impact on the simulator's performance.

6. A GDL expression is a series of "factors" joined by binary operators. All operators have the same precedence and are evaluated left-to-right. In the following list of operators, X is any expression and Y is any factor.

x AND y x OR y x XOR y	Boolean AND, OR, or exclusive OR of X and Y. If either X or Y is None, the result is the other input. If both are None, the result is None.
x NAND y	Equivalent to "NOT (x AND y)".
x NOR y	Equivalent to "NOT (x OR y)".
x SAME y	Both X and Y must be pins. Equal to One if both pins are connected to the same electrical node, or Zero if the pins are not on the same node.

7. A GDL factor is either an expression in parentheses, or one of the following (X is any factor; P is any pin or internal node):

<var>	A state variable. The result is One or Zero depending on the current value of the variable.
<node>	A pin number or internal node. The result is either One, Zero, or None, depending on the value driven on the node in the previous time-step.
ONE	The constant value, One. This can also be written "TRUE" or "1".
ZERO	The constant value, Zero. Also "FALSE" or "0".
NOT x	The result is One if X is Zero, Zero if X is One, or None if X is None.
RISE p	The result is One if the specified node is receiving a rising transition, that is, the previous time-step drove the pin to a One, but the time-step before that drove it to a Zero. Otherwise, the result is Zero.
FALL p	The result is One if the specified node is receiving a falling (One to Zero) transition.
FIX x	The value of X, with None converted to Zero.
STRONG p	The result is One or Zero if the specified node was driven by a strong (i.e., normal) value on the previous time-step, or None if the node was undriven or was only weakly driven.

8. For examples of GDL programs, consult the many gates in the files "log.cnf" and "actel.cnf".

TEXT FILE FORMAT

(This section is not yet completed.)

".GATE" FILE FORMAT

(This section is not yet completed.)

REFERENCE MATERIAL

List of commands
Configuration files
Digital simulator commands
Analog simulator commands
Single-key commands
Color names
Command-line options
X Display Preferences

LIST OF COMMANDS

1. This is a partial list of commands that may be entered in their full-length form (i.e., press ":", then type the command). Many of these commands may also appear in the file "log.cnf" which is read when LOG starts up.

2. In this list, "ON/OFF" means either the word "ON", to turn an option on, or "OFF", to turn an option off, or no word at all, to "toggle" the option between on and off.

3. Many commands have single-key equivalents. The bindings shown here are the defaults, which may be changed by the MACRO command.

4. The following commands, when used in the form of single-key commands, may be used even in the middle of drawing a wire, selecting a rectangle, etc.:

```
ALTPOSN       GLOW          OFF           SIM/ONOFF/POWER
ARROW         GLOWSOLDER    ON            SNAP
AVOID         GRID          PAGE n        TRACE
CLOSEFILES    HELP          PROBE         TRACEFILE
CONFLICT      HOME          QUIET         VERBOSE
DOTS          INVISIBLE     RESET         ZOOMDN
DUMP          INVLABEL      RESPONSE      ZOOMUP
DUMPFILE
```

   Other commands will cause any other operation in progress to abort before the new command begins.

5. Here is the complete list of LOG commands:

ABORT	(<CNTRL>-C) Abort the current command, mode, or screen.
ALTPOSN	("," key.) Scroll to an alternate location on the page. LOG essentially maintains two current locations on the page, like the "mark" in the Emacs text editor; this command exchanges them.
ARROW	Revert to the standard arrow-shaped cursor. This undoes any previous PROBE or GRID command.

AUTOWINDOW Toggle a new window management mode. In this mode whenever text input into the newcrt window is necessary the newcrt window automatically rises in front of any windows that occlude it. Once the window is no longer needed, it returns to its previous condition.

BOX ("b" key.) Draw a dashed box.

CAT (Shift-"C" key.) Switch to the Catalog screen.

CENTER Shift the entire circuit so that it is centered in the LOG circuit universe, i.e., so that a Home command will home to the center of the circuit. Note that this actually moves the circuit elements, whereas scrolling with the knob just changes your view on the elements.

CLEAR Delete the entire circuit page. Use with caution!

CLOSEFILES
Close the "dump" and "trace" files, if they are open. These files will automatically be re-opened (for appending) if they are again used. The dump file is used by the "Dump" command in the Scope. The trace file is used by various LOG debugging commands.

CLOSEH Enter "close-horizontal-space" mode, similar to OPENH except that it deletes the contents of the box you draw and shifts other things to the left. DANGEROUS!

CLOSEV Enter "close-vertical-space" mode, similar to OPENV. DANGEROUS!

CNFG ("c" key.) Switch gate-tapping mode to open a gate's configuration screen when it is tapped.

COLOR screen-color red-value green-value blue-value
COLOR color-name screen-color
Customize the LOG color scheme. LOG's colors are determined in two steps. A "palette" of up to 16 available colors is specified by the first form of the command. "Screen-color" is either a number from 0 to 15, or one of the names listed below. Each of the red, green, and blue values run from 0 (black) to 255 (full intensity). The second form of the COLOR command determines which of these 16 colors is used for various kinds of objects. The color names defined by LOG are listed in a later section.

Screen-colors:

```
   0  GRAY     4  MRED     8  BLACK    12 DCYAN
```

1	RED	5	ORANGE	9	PINK	13	DRED
2	GREEN	6	CYAN	10	DYELLOW	14	LGRAY
3	YELLOW	7	WHITE	11	XGREEN	15	CRED

On eight-color screens, all sixteen names or numbers may still be used; LOG combines similar colors internally. The effect will be that changing the red/green/blue content of one of the above colors may affect other colors as well on an 8-color screen.

CONFLICT n
CONFLICT ON/OFF
Turn conflict reporting on or off. In Digital LOG, node conflicts are reported if one gate tries to drive a node to "1" while another is driving it to "0". AnaLOG does not report conflicts. Conflicts are shown by a bright pulsating red. If conflicts are turned off, they are not visible on the screen. If an integer "n" is given, conflicts are turned on and the number of time steps to wait before reporting a given conflict is set to "n". For example, if n=0, conflicts are reported immediately. For n=1, each conflict must remain for at least two simulation timesteps before it is considered interesting. The default is conflict reporting on, with n=1.

COPY ("/" key.) Enter copy-area mode.

CSTOP ON/OFF
Turn conflict-stopping mode on or off. When this mode is on, the simulation stops (as if you had given an OFF command) the instant any conflict is reported. The CONFLICT command can be used to select how long a conflict must persist before it is reported. This mode is off by default.

DEFINE (Shift-"D" key.) Display the simulator's definition
for

a gate-kind, specified either on the command line or by touching a representative gate. Pressing shift-"D" in the Catalog screen also displays the definition for the gate the cursor is over.

DEL ("d" key.) Enter delete mode.

DOTS ON/OFF
Control dots-visible mode. When this is on (default), red connection dots on gates are always visible. When this is off, red dots are usually visible only on pins that are not connected to wires.

DUMP   Write out a "debugging dump" to the printer. For LOG
            maintainers only.

DUMPFILE filename
            Close any current dump file, then select
"filename.text"
            as the new dump file. If this file already exists, new
            text will be appended to it.

EXAMINE   ("e" or "x" key.) Enter Probe mode temporarily.
            The cursor returns to normal as soon as you click the
            right button. This is in contrast to the PROBE
command,
            which turns the Probe cursor on until you explicitly
            turn it off again.

EXIT   Exit from LOG. Also in the Misc menu, or on the Shift-
Q,
            Shift-Z, and Control-D keys.

EXTRACT   Delete and re-paste the entire contents of the page.
If
            LOG's electrical connectivity information becomes
corrupt,
            this will usually fix it.

FAST   ("f" key.) Turn on fast mode. This happens
            automatically after a certain amount of time passes
with
            no mouse or keyboard inputs.

GET gatename gatename gatename ...
            Load the specified gates into the Catalog screen from
            the gate library. The gate names may include wildcard
            characters '*' and '?'.

GLOW ON/OFF
            ("g" key.) Turn glow mode on or off.

GLOWSOLDER ON/OFF
            Turn glowing-solder mode on or off. This controls
            whether solder dots are visible while in GLOW mode.
            It is on by default; turning glowing-solder off may
            improve simulation performance in GLOW mode.

GRID   (Shift-"G" key.) Change the cursor to a cross-hair
            shape. If the cursor is already a cross-hair, change
            it back to the regular arrow cursor.

GROUP n "Name"
Set the name that appears on the top of page number "n" of the Library screen. The number "n" must be from 0 to 8. If the "Name" argument is omitted, this command instead selects which page number the Library screen will display next time it is entered.

HELP ("?" key.) Display this file using the Chipmunk Help System.

HOME ("h" key.) Reset the scroll and zoom amounts to the default position on the page.

IDENTIFY Used by the LOG-to-NTK conversion tool.

INVISIBLE ON/OFF
("i" key.) Turn "Invisible" mode on or off. In this mode, wires and most gates are invisible. However, labels, boxes, and switch/indicator gates are visible. Also, operations that edit the circuit are disallowed. Thus, a user can switch into Invisible mode and then operate a simulated "front panel" without confusion or fear of changing the circuit.

INVLABEL ON/OFF
(Shift-"I" key.) Turn "Invisible Labels" mode on or off. In this mode, labels and boxes are invisible.

LABEL message
("l" key.) Create a new label on the page. If a "message" argument is supplied, a label with the specified text is created on the lower-left corner of the screen. If no argument is supplied, LOG interactively enters the text for the label.

LIBRARY ("l" key or LIBR item in the Catalog screen.) Switch directly to the Gate Library screen.

LOAD filename
(Shift-"L" key.) Load the specified circuit file onto the current page, replacing anything that was on that page before. If no file name is specified, LOG displays a list of files in the current directory and asks the user to select one, or type the name. If the file name is '*', LOG simply asks the user to type the name. The

```
                shift-L key is bound to "LOAD *".


     LOGNTK     Start the LOG-to-NTK conversion tool.


     LOGNTK: parameter value
                Set up a configuration parameter for the LOGNTK tool.
                Commands of this form usually appear in the file
                "/lib/log/logntk.cnf".


     LPLOT: parameter value
                Set up a configuration parameter for the PLOT tool.
                Commands of this form usually appear in the file
                "/lib/log/lplot.cnf".


     MACRO  key-name  command
                Associate the specified command with a key.  The key-
name
                may be a single letter or other character (lower-case
                letters represent unshifted keys; upper-case letters
                represent shifted keys); or one of the words "bs",
"tab",
                "sp" (space bar), or "cr" (Return or Enter); or a
control
                key of the form "^A" or "^["; or a three-digit decimal
                ASCII code.  The "command" may be any of the commands
                listed here, and may include arguments.  If you give
only
                a key-name, the computer displays the command that is
                currently bound to the key.  MACRO with no arguments
                displays a list of all current key bindings.


     MAKE gate-name
                Bring a gate in from the catalog by name, and put it in
                the drawing area.  A shorthand for using the mouse in
the
                the CAT and LIBR screens.  If the gate-name contains
                wildcard characters, multiple gates may be loaded into
                the Catalog but only one of them will be created on the
                circuit page.


     MARKER ON/OFF
                ("m" key.)  Turn printing markers on or off.  The
                printing markers are used mainly by the "PLOT" command;
                they indicate the area of the circuit page which is to
                be fitted into the printed page.  The printing markers
                appear on the corners of the visible screen, and you
                can move them by regular editing operations to bracket
                any rectangular area of the page.


     MESSAGE message
                Display the specified message in the "message area" of
```

the screen. The message area runs down the left edge of the screen; messages are drawn in yellow overlaying the drawing area.

MIRX — Select "mirror-in-X" gate-tapping mode. When a gate is tapped in this mode, it is flipped across its vertical axis. This works both in the menu area and in the drawing area, although some gates (such as switches) have special behaviors when tapped in the drawing area.

MIRY — Select "mirror-in-Y" gate-tapping mode.

MOVE — ("m" key.) Move an area of the diagram. This is similar to deleting the area with DEL, then pasting it back in a new location with PASTE.

NAME file-name
See and/or change the "current file name" for the circuit page. This is the file name that will be used by default in later "SAVE" commands.

OFF — Turn simulation OFF.

ON — Turn simulation ON.

OPENH — Enter open-horizontal-area mode. Sweep out a rectangle;

everything to the right of the lefthand edge of the rectangle is moved to the right by the amount of the width of the rectangle. Wires straddling the left edge are stretched appropriately. BEWARE: This command may corrupt the LOG data structures if used in such a way as to break or make electrical connections in the circuit. Unless you really know what you are doing, use the cut-and-paste method of moving things.

OPENV — Enter open-vertical-area mode, similar to OPENH. DANGEROUS!

PAGE n — Switch the display to circuit page "n", from 1 to 9. "n" may also be "+" or "-", to switch to the next or previous numbered page.

PASTE — Enter Paste mode.

PLOT — Run the circuit plotting tool.

```
       POPUP  menu  position  kind  command  message
                    Change one of the locations in the pop-up menus.
"Menu"
                    is a number that selects which menu (1=Frills,
2=Editing,
                    3=Cursor, 4=Misc); "position" is an integer from 1 to 8
                    that identifies which spot in the menu:   [ 1  3  5  7
]
                                                              [ 2  4  6  8
]
                    "Kind" is normally zero; "command" is the command (from
                    this list) that is to be executed if the menu is
selected,
                    enclosed in "quotes" if it contains spaces; and
"message"
                    is the wording for that menu element.  If "message"
begins
                    with '*' or '#', it will be colored red or light blue,
                    otherwise it will be green.


       PROBE        ("." key.)  Change to the Probe cursor.  If already
                    using the Probe cursor, change back to an arrow.  In
                    Probe mode, holding the cursor near a wire or gate pin
                    displays the value on that electrical node, in a
                    simulator-dependent way.  Holding the cursor near a
                    gate in the drawing or menu areas displays the name
                    of the gate on the screen.  Tapping a gate or node
                    opens it for configuring, if it has anything to
                    configure.  (Neither AnaLOG nor Digital LOG nodes
                    are configurable, but many gates are.)  Tapping and
                    configuring a gate in the menu area sets the defaults
                    for newly created gates pulled out of that menu slot.
                    Finally, tapping out in the drawing area draws a
                    yardstick.  See also the EXAMINE command (on the
                    "e" and "x" keys) which selects a temporary
                    Probe mode.


       QUIET        ("q" key.)  Turn sound prompting on or off.


       READ  filename
                    Read in a circuit from a file.  This is similar to the
                    LOAD command, but it rebuilds the circuit's
connectivity
                    information as it loads, rather than simply reading it
                    from the file.  As a result, READ is much slower than
                    LOAD, but it is far more robust in case the file is
                    corrupt or out of date.  For example, if you run LOGED
                    and add more pins to a gate, then try to LOAD a circuit
                    diagram containing that gate, LOG may crash since the
                    connectivity recorded in the file is out of date.  But
                    the READ command will read the file with no problems.
                    In general, use the fast LOAD command unless you have
                    a good reason not to.
         REFRESH    (Space bar.)  Refresh the screen.


         RESET      (Shift-"R" key.)  Send a Reset signal to the
  simulator(s).
                    This resets simulation time back to zero, and has other
                    simulator-dependent effects.  For AnaLOG, it resets all
                    voltages to their initial values.  For Digital LOG,
                    nothing happens except that the TO/FROM signal "Reset",
                    if used, gets a brief "1" pulse.


         RESPONSE  min  max  rate
                    Set response-time parameters.  With no arguments,
                    displays current parameters.  Initial values are
                    2, 50, and 35, respectively.  During editing, LOG
                    tries to check the mouse and keyboard at least every
                    "min" centi-seconds.  If the user stops editing,
                    this rate slowly increases until the maximum of
                    "max" centi-seconds.  The rate of increase is
                    one unit per "rate" centi-seconds of idle time.


         ROT        Select "Rotate by 90 degrees" gate-tapping mode.  When
                    a gate is tapped in this mode, it rotates 90 degrees
                    counter-clockwise.  This works both in the menu area
  and
                    in the drawing area, although some gates (such as
                    switches) have special behaviors when tapped in the
                    drawing area.


         SAVE filename
                    Save circuit page(s).  If a name is given, saves the
                    current page in "name.LGF".  If the name is "*",
  prompts
                    the user to enter the name to save.  If no name is
                    given, re-saves all circuit pages which have been
                    since last saved or loaded.  Shift-"S" is bound
                    to "SAVE *".


         SCOPE      ("s" key.)  Switches to the Scope screen.


         SHELL shell-command    ( or:  !shell-command )
                    (Also, "!" as a single-key command.)  Start a sub-
  Shell.
                    If a shell-command is provided the shell executes this
                    one command and returns.  If no shell-command is
  provided,
                    you get an interactive sub-shell.  Type quit or press
                    control-D to return from the sub-shell.


         SIM (or ONOFF or POWER)
                    Turn simulation ON or OFF (toggling).
         SNAP       ("$" key.)  Turn "snap-to-grid" mode on or off.
  Default
                    is off.  In this mode, the cursor arrow always jumps
                    to the nearest spot on the LOG editing grid rather
                    than moving smoothly.


         STATUS     Enter the "Status" display.  Use the knob or arrow
                    keys to switch among various pages.  A page-name may
                    also be given with the STATUS command:

                          LOG      General system status
                          MEMORY   Memory usage status
                          MACRO    Keyboard mappings
                          16       Digital simulator
                          32       Analog simulator
                          LPLOT    Plotting tool
                          LOGNTK   LOG-to-NTK tool


         STEP n     Turn simulation OFF, if necessary, then step the
                    simulation by one time-step.  If an integer "n" is
                    provided, executes "n" time-steps before stopping.


         TAPMODE    (menu selection, or shift-"M" key.)  Select the next
                    gate-tapping mode, in order:  ROT, MIRX, MIRY, CNFG.


         TOOL name  Send a "select" command to the specified tool.  Various
                    tools respond to this in various ways; the default is
                    to ignore the signal.  (The Digital LOG simulator, tool
                    name "16", responds by putting up a configuration
                    screen of digital-LOG parameters.)  If a TOOL command
                    is given with no "name" argument, it puts up a screen
  of
                    available tool names and "selects" whatever tool the
                    user chooses with the mouse.  The shift-"T" key calls
                    up an interactive TOOL command.


         TRACE      Turn the debugging "wallpaper" mode on or off.  Please
                    don't do this unless a LOG maintainer is there.  Trace
                    output goes to the "Trace" file, which by default is an
                    appropriate printer.


         TRACEFILE filename
                    Close any current trace file, then select
  "filename.text"
                    as the new trace file.  If this file already exists,
  new
                    text will be appended to it.
```

VERBOSE	("v" key.) Turn informational messages on or off. These messages are mostly of the "Glow mode is now OFF" variety.
VMESSAGE	Similar to MESSAGE, but works only if in Verbose mode.
YARDSTICK	Draw a "yardstick", useful for measuring the spacing of objects in your diagram.
ZOOMDN	Zoom down, to view more of the circuit page.
ZOOMUP	Zoom up, to view part of the circuit in greater detail.

CONFIGURATION FILES

1. LOG reads the file "log.cnf" every time it starts up. If you do not have one in the current directory, it uses "/lib/log/log.cnf". You can give the "-c" command-line switch to specify your own configuration file. The "use log" script actually loads the program "diglog" into the computer and aliases the "log" command to "diglog -cdlog", causing "dlog.cnf" (in the current directory or in /lib/log) to be used. Similarly, "use analog" loads the program "analog" and aliases "log" to "analog -calog".

2. To create your own configuration file, the best approach is to make a file named "log.cnf", "alog.cnf", or "dlog.cnf", whichever is appropriate, in your current directory. In that file, put a line something like "include /lib/log/alog.cnf", followed by any additional commands to override the system defaults. Do not copy the system CNF files if you can avoid it; these files are subject to change at any time.

3. For example, your CNF file might look like this:

```
        { A comment saying what this is used for }
        include /lib/log/dlog.cnf     { Inherit from Digital LOG }
        gates + ~me/log/mygates       { Add on my own gates file }
        menu a b c d e vdd gnd        { Choose my own menu gates }
        glow on                       { Turn on "glow" mode }
```

   In this example, the "+" in the GATES command means to add the name(s) to the existing list of gates files. Since the MENU command does not have a "+", it replaces the list from the other file. This notation works with the GATES, GET, GETGROUP, and MENU commands in configuration files.

4. The following ":" commands may also be used in a "log.cnf" configuration file:

```
AVOID         GLOWSOLDER    OFF           QUIET
COLOR         GROUP         ON            RESPONSE
CONFLICT      INVISIBLE     SIM           SNAP
DOTS          INVLABEL      POPUP         VERBOSE
GLOW          MACRO         PROBE
```

5. The following additional commands are allowed in configuration files:

BOBCAT command-line
CHIPMUNK command-line

Process the command only if LOG is being run on the specified kind of computer. The "command-line" here is any of the commands listed in this section; in other words, the words BOBCAT and CHIPMUNK are prefixes that can be added to any command in a CNF file.

COMMAND toolname command command command ...

Register that the listed commands are handled by the indicated tool. If a command is given and its owning tool has not been initialized, that tool will be initialized.

DO command-line

Add the command-line to a list of ":" commands to be done as soon as LOG is started. These commands are executed as if typed from the keyboard, i.e., they may be commands which are ordinarily forbidden in CNF files. The "DO" commands are executed in the order they originally appeared in the CNF files.

GATES filename filename filename ...

Register the specified ".gate" files into the LOG gate library. Files are searched for a given gate starting at the first filename of the first GATES command in the CNF file. Ordinarily, the presence of the first GATES command in a given CNF file overrides all GATES commands in previously read CNF files (i.e., INCLUDE files). If the first word after GATES is "+", the listed filenames will instead be added to the existing list of gates files. If no explicit directory is given, LOG looks first in the current directory, then in the "home" directory, then in /lib/log. If the gates file is not found, it is ignored.

GET gate gate gate ...

Load the specified gates into the gate catalog. The gate names may include wildcard characters "*" and "?", and may also specify one or more group numbers from 0 to 8: "GET A*;34" gets all gates whose names begin with A, which are found on pages 3 or 4 of the Library. If this is the first GET command in its CNF file, it overrides any previous GET commands from other files

```
               unless GET is followed by the word "+".  (The obsolete
               word LOAD is accepted as a synonym for GET.)


    GETGROUP gate gate gate ...
               Load the specified group of gates into the gate
catalog.
               This like a normal GET command, except the gates are
               guaranteed to be placed together (provided there's
               room), and a box is drawn around them on the catalog
               screen.


    HELP filename
               Name of a ".help" file to be used when the user
               asks for help.


    HOME directory-name
               Specify a directory to be used as the "home"
               directory.  If a CNF file, gates file, help file,
               or news file is not found in the current directory,
               LOG looks next in this "home" directory, then
               finally in "/lib/log".  The default "home" directory
               is "~/log", that is, the "log" subdirectory of the
               user's home directory.


    INCLUDE cnf-file
               Read commands from the specified ".cnf" file, then
               resume reading from this file.  Include files may
               be nested.


    MENU gate gate gate ...
               Load the specified gates into the menu area.  If
               there are more gate names than menu slots, only the
               first few names are used.  The first word may be
               "+" to add the specified gates to a list from a
               previous CNF file.


    NEWS file-name
               Name of a ".text" file whose contents are displayed
               when LOG begins.


    SIGNALS number
               (or NODES.)  Specifies the maximum number of TO/FROM
               signal names that can exist at one time during the
               LOG session.  Default is 200.


    TABLET number
               Set the HP-IB address of the graphics tablet to be
               used.  Default is 0, which means look for the tablet
               on the HP-HIL or at the standard address.
```

```
TOOL tool-name "full-name"
           Register a tool name so that the run-time TOOL command
           will display its name.  (Ordinarily, tools are
           automatically registered as soon as they are needed.)


UNDO       Delete all "DO" commands registered up to this point.


tool-name: tool-command
           This directs a command to a particular LOG "tool".
           The standard tool names are:

                 16       Digital simulator
                 32       Analog simulator
                 LPLOT    Plotting tool
                 LOGNTK   LOG-to-NTK tool
```

DIGITAL SIMULATOR COMMANDS

1. The following additional ":" commands are available with the Digital LOG simulator. Except for STABILIZE they may be used in "log.cnf" files by preceding them with the word "16:", as in "16: stabdelay 2.5".

```
DIGON      Turn digital simulation ON.  This is the normal case.


DIGOFF     Turn digital simulation OFF.  This command is similar
           the regular OFF command, except only the digital
           simulator is affected.  If several simulators are
           being run at once (say, digital and analog), the
           other simulators will proceed unaffected.


DIGONOFF ON/OFF
           Turn digital simulation ON or OFF.


STABDELAY time
           Set amount of time to wait for circuit stabilization
           (see STABILIZE) to "time" seconds.  Default is 5.0.
           With no argument, the command shows the current value.


STABILIZE  Attempt to stabilize an oscillating circuit.  Because
           all digital gates have identical, exact propagation
           delays, oscillations can develop in feedback circuits
           that would never arise in a real circuit, where the
           propagation delays are not so exactly balanced.  The
           STABILIZE command randomizes the propagation delays of
           gates in order to kill such oscillations.  Of course,
           a circuit that really is oscillating will continue
           to do so.  The "stabilization" effect wears off after
```

about five seconds, usually plenty of time.

DIGSTEP time-step
Set the amount of simulated time per digital timestep. Since every digital time step is the same, this value normally does matter unless you are using the digital simulator along with another simulator which has a more concrete idea of time (such as AnaLOG). The default value is "10ns", i.e., 1E-8 seconds. The value may be specified in engineering or real-number notation. With no argument, the command shows the current value.

ANALOG SIMULATOR COMMANDS

1. The following additional ":" commands are available with the AnaLOG simulator.

DIM ON/OFF	Turn dim-unconnected-gates mode on or off.
ERASE	Same as touching Erase on the "Scoreboard" gate. If you are selecting Set and Erase often, you might want to use the MACRO command to put SET and ERASE on a pair of keys, for convenience.
EXACT	Same as touching Exact on the "Scoreboard" gate.
NEWMODEL	Set non-saving device-model mode, in which modelling parameters for transistors, and simulation constants such as "Vdd", are reset to their default values rather than being loaded from each circuit file.
OLDMODEL	Set the default device-model mode, in which modelling parameters for transistors, and simulation constants such as "Vdd", are saved in circuit files along with the transistor and "Scoreboard" (NUMBERS) gates.
PROBEPREC digits	Set the number of decimal digits for voltages displayed by the "probe" cursor. Default is 3.
RELAXED	Same as touching Relaxed on the "Scoreboard" gate.
SET	Same as touching Erase on the "Scoreboard" gate.

2. Also, a great many AnaLOG parameters may be set in ".CNF" files. These commands all begin with the prefix "32:". Defaults appear in the file "models.cnf" in the analog /lib directory -- ask your local maintainer for details

SINGLE-KEY COMMANDS

Many commands also appear in single-key forms. In this list, lower-case letters represent unshifted keys, and capital letters are shifted keys.

:	Enter a full-line command.
space	Refresh the screen: REFRESH.
knob	(or arrow keys.) Scroll across the page.
<CNTRL>-C	Abort the current command or mode.
1-9	(Digits.) Switch to a new page. Same as PAGE 1 to PAGE 9.
!	Shell escape: SHELL.
*	Draw the contents of the Paste buffer: PASTE.
,	(Comma.) Switch to alternate viewing position: ALTPOSN.
.	(Period.) Enter or leave PROBE mode.
/	Copy an object or area into the Paste buffer: COPY.
<	Zoom down, to see more of the diagram at once: ZOOMDN.
>	Zoom up, to focus on a smaller area of the diagram: ZOOMUP.
?	Display this help file.
+	(also NEXT on Bobcats.) Switch to next higher page number.
-	(also PREV on Bobcats.) Switch to next lower page number.
b	Draw a dashed box: BOX.
c	Switch to "CNFG" gate-tapping mode: CNFG.
C	Enter the Catalog screen: CAT.
d	Delete objects: DEL.
D	Show simulator's definition for a gate: DEFINE.
e	Enter temporary PROBE mode: EXAMINE.
f	Turn on FAST mode.
g	Turn GLOW mode on or off.
G	Turn GRID (cross-hair cursor) mode on or off.
h	Return to home (unscrolled) position: HOME.
i	Turn INVISIBLE mode on or off.
I	Turn INVLABEL (invisible labels) mode on or off.
l	Draw a label: LABEL.
L	Load a new circuit page: LOAD.
m	Move an object or area: MOVE.
M	Select the next gate-tapping mode: TAPMODE.
$	Turn SNAP-to-grid mode on or off.
o	Turn simulation on or off: ONOFF.
p	Enter the circuit-plotting tool: PLOT.
r	Switch to "ROT" gate-tapping mode: ROT.
R or 0	Reset the simulators: RESET.
q	Turn QUIET mode on or off.
Q or Z	(or Control-D.) Exit from the LOG system: EXIT.
s	Enter the Scope screen: SCOPE.
S	Save the current page to a file: SAVE.
t	Run simulation for one time-step, then stop: STEP.
T	Enter the Tool screen: TOOL.

v	Turn VERBOSE mode on or off.
x	Enter temporary PROBE mode: EXAMINE.
y	Draw a YARDSTICK.

COLOR NAMES

1. The following list is the complete set of color-names defined so far in the LOG system. See the COLOR command for details on how to reconfigure these color names.

2. After each color-name is its default color, then a description of what screen objects that color-name governs.

BACKGR	Gray	Screen background color.
BASELINE	Cyan	Color of line above menu area.
BLUEWORD	Cyan	Light blue text in pop-up menus.
CATALOGBOX	Dim yellow	Gate grouping box in Catalog screen.
CATGATE	Cyan	Color of gates in Catalog screen.
CHART	Yellow	Default color of traces in Scope.
CONFLICT	C-Red	Color of "signal-conflict" in a wire.
CURSOR	White	Regular cursor color.
DASHBOX	Dim yellow	Color of dashed boxes in circuits.
DEFINEBACK	Black	DEFINE command's box background.
DEFINEBOX	Cyan	DEFINE command's box outline.
DEFINETEXT	Green	DEFINE command's text color.
DIMGATE	Dim cyan	Color of "dimmed" gates in circuits.
DIVISION	White	Color of grid dots in Scope.
GATE	Cyan	Normal color of gates in circuits.
GATEBLACK	Black	Black parts of gate pictures.
GATEGREEN	Green	Green parts of gate pictures.
GATEORANGE	Orange	Orange parts of gate pictures.
GATEPIN	Red	Color of gate connection dots.
GATERED	Red	Red parts of gate pictures.
GATEWHITE	White	White parts of gate pictures.
GATEYELLOW	Yellow	Yellow parts of gate pictures.
GLOW1	Black	AnaLOG glow color for <= 0 volts.
GLOW2	Dim red	AnaLOG glow color for 0-15
GLOW3	Medium red	AnaLOG glow color for 15-50
GLOW4	Red	AnaLOG glow color for 50-85
GLOW5	Pink	AnaLOG glow color for 85-100
GLOW6	White	AnaLOG glow color for >= Vdd.
IDENTIFY	Red	Color of identified things in LOGNTK.
IMETER	Orange	AnaLOG current meter readout.
INSTANCE	Dim yellow	Name shown in Digital instance gates.
KINDGATE	Green	Color of gates in menu area.
LABELTEXT	Dim yellow	Color of circuit labels.
LIMITON	Red	AnaLOG current-limit indicator.
MARKER	Pink	Color of printing markers.
MENUWORD	Green	Color of text in menu area.
MESSAGE	Yellow	Informative messages in drawing area.
OFFLED	Black	Digital LOG logic "0" signal.
ONLED	Red	Digital LOG logic "1" signal.
PAGE1	Green	Color of the word "PAGE".
PAGE2	Yellow	Color of current page number.
PAGE3	Green	Color of the word "OF".
PAGE4	Yellow	Color of maximum page number.

PEN1	White	Color for plotter pens #1-6.
PEN2	Red	. These colors are used by the PLOT
PEN3	Green	. command for circuit and data
PEN4	Orange	. plotting.
PEN5	Cyan	.
PEN6	Yellow	.
POPUPBOX	White	Outline of pop-up menus.
POPUPWORD	Green	Text in pop-up menus.
POPUPSEL	Pink	Highlighted text in pop-up menus.
PROBE	Dim yellow	AnaLOG probe-mode display color.
REDWORD	Red	Red text in pop-up menus.
SCROLL	White	Color of scroll-indication lines.
SELECT	White	Color of area-selection rectangle.
SELWORD	Yellow	Highlighted words in menu area.
SIGNAL	Pink	Color of TO/FROM signal names.
SOLDER	Green	Color of solder dots on wires.
SWITCHOFF	Light gray	AnaLOG switch "off" color.
SWITCHON	Light gray	AnaLOG switch "on" color.
TEMPLATE	Pink	Name shown in Digital template gates.
TIME	Dim yellow	Color of TIME gate display
VMETER	Dim yellow	AnaLOG voltage meter readout.
WIRE	Green	Color of non-glowing wires.
XWIRE	X-Green	Color of wires being drawn.

COMMAND-LINE OPTIONS

These are all the command-line options log currently understands:

-x displayname : This option specifies the screen where the newcrt and mylib windows will appear. For example, the command analog -x foo.school.edu:0.0 results in the newcrt and mylib windows appearing on the console of the machine foo.school.edu. An alternative to setting the DISPLAY environment variable in the launch shell.

-c filename : Specifies a .cnf configuration file.

X DISPLAY PREFERENCES

Log looks in your .Xresources file for geometry information. If set
properly, this allows windows of a certain size to be automatically placed at a particular place at every invocation of log; manual placement of windows is no longer necessary. This feature was added by
Jim Clark at Harvard. Adding these lines to your .Xresources file:

```
mylib.geometry: =1150x550+0+0
newcrt.geometry: +510+586
```

will result in good placement and sizing for the color monitors shipped with SPARC IPC's; for other screens, charge the numbers after
"+" signs for new positions, and change the numbers between the "x" for modify the size of the mylib window. The newcrt window cannot be

```
      modified; log expects it to be a fixed size.

      Psys now senses if a display screen is black and white, and tries
to
      use stipple instead of colors. However, there are times where you
want
      to force the tools to display in black and white, even if the
display
      screen is color. These times include using an OS/2 based color PC
as an
      X terminal for the analog tools, because of present
incompatabilities
      between PC graphics cards and analog. To force a black and white
      display, add this line to your .Xresources file:

      mylib.color:  black_and_white

      and then restart your X server. Black and white support was
provided
      by Tor Sverre Lande (Oslo), and the .Xresources flag was added by
Mike
      Godfrey (Stanford).

(End of LOG documentation.)
```